

Mourad Fakhfakh  
Esteban Tlelo-Cuautle  
Patrick Siarry *Editors*

# Computational Intelligence in Analog and Mixed-Signal (AMS) and Radio-Frequency (RF) Circuit Design

 Springer

Computational Intelligence in Analog  
and Mixed-Signal (AMS)  
and Radio-Frequency (RF) Circuit Design

Mourad Fakhfakh · Esteban Tlelo-Cuautle  
Patrick Siarry  
Editors

# Computational Intelligence in Analog and Mixed-Signal (AMS) and Radio-Frequency (RF) Circuit Design

 Springer

*Editors*

Mourad Fakhfakh  
Department of Electronics  
ENET'Com, University of Sfax  
Sfax  
Tunisia

Patrick Siarry  
Laboratory LiSSi (EA 3956)  
Université Paris-Est Créteil  
Vitry-sur-Seine  
France

Esteban Tlelo-Cuautle  
Department of Electronics  
INAOE  
Tonantzintla, Puebla  
Mexico

ISBN 978-3-319-19871-2      ISBN 978-3-319-19872-9 (eBook)  
DOI 10.1007/978-3-319-19872-9

Library of Congress Control Number: 2015942631

Springer Cham Heidelberg New York Dordrecht London  
© Springer International Publishing Switzerland 2015

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media  
([www.springer.com](http://www.springer.com))

# Preface

Computational intelligence has been an astounding success in the engineering domain, particularly in electronic design. Over the last two decades, improved techniques have raised the productivity of designers to a remarkable degree. Indeed, in the areas of digital, analog, radio-frequency, and mixed-signal engineering, there is a focused effort on trying to automate all levels of the design flow of electronic circuits, a field where it was long assumed that progress demanded a skilled designer's expertise. Thus, new computational-based modeling, synthesis and design methodologies, and applications of optimization algorithms have been proposed for assisting the designer's task.

This book offers the reader a collection of recent advances in computational intelligence—algorithms, design methodologies, and synthesis techniques—applied to the design of integrated circuits and systems. It highlights new biasing and sizing approaches and optimization techniques and their application to the design of high-performance digital, VLSI, radio-frequency, and mixed-signal circuits and systems.

As editors, we invited experts from related design disciplines to contribute overviews of their particular fields, and we grouped these into the following:

- Volume 1, “Computational Intelligence in Analog and Mixed-Signal (AMS) and Radio-Frequency (RF) Circuit Design,” contains 17 chapters, divided into two parts: “Analog and Mixed-Signal Applications” (Chaps. 1–8) and “Radio-Frequency Design” (Chaps. 9–17).
- Volume 2, “Computational Intelligence in Digital and Network Designs and Applications,” contains 12 chapters, divided into three parts: “Digital Circuit Design” (Chaps. 1–6), “Network Optimization” (Chaps. 7–8), and “Applications” (Chaps. 9–12).

Here, we present detailed descriptions of the chapters in both volumes.

## **Volume 1—Computational Intelligence in Analog and Mixed-Signal (AMS) and Radio-Frequency (RF) Circuit Design**

### **Part I—Analog and Mixed-Signal Applications**

Chapter 1, “I-Flows: A Novel Approach to Computational Intelligence for Analog Circuit Design Automation Through Symbolic Data Mining and Knowledge-Intensive Reasoning,” was written by Fanshu Jiao, Sergio Montano, Cristian Ferent, and Alex Doboli. It presents an overview of the authors’ ongoing work toward devising a new approach to analog circuit synthesis. The approach computationally implements some of the facets of knowledge-intensive reasoning that humans perform when tackling new design problems. This is achieved through a synthesis flow that mimics reasoning using a domain-specific knowledge structure with two components: an associative part and a causal reasoning part. The associative part groups known circuit schematics into abstractions based on the similarities and differences of their structural features. The causal reasoning component includes the starting ideas as well as the design sequences that create the existing circuits.

Chapter 2, “Automatic Synthesis of Analog Integrated Circuits Including Efficient Yield Optimization,” was written by Lucas C. Severo, Fabio N. Kepler, and Alessandro G. Girardi. Here, the authors show the main aspects and implications of automatic sizing, including yield. Different strategies for accelerating performance estimation and design space search are addressed. The analog sizing problem is converted into a nonlinear optimization problem, and the design space is explored using metaheuristics based on genetic algorithms. Circuit performance is estimated by electrical simulations, and the generated optimal solution includes yield prediction as a design constraint. The method was applied for the automatic design of a 12-free-variables two-stage amplifier. The resulting sized circuit presented 100 % yield within a 99 % confidence interval, while achieving all the performance specifications in a reasonable processing time. The authors implemented an efficient yield-oriented sizing tool which generates robust solutions, thus increasing the number of first-time-right analog integrated circuit designs.

Chapter 3, “Application of Computational Intelligence Techniques to Maximize Unpredictability in Multiscroll Chaotic Oscillators,” was written by Victor Hugo Carbajal-Gómez, Esteban Tlelo-Cuautle, and Francisco V. Fernández. It applies and compares three computational intelligence algorithms—the genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO)—to maximize the positive Lyapunov exponent in a multiscroll chaotic oscillator based on a saturated nonlinear function series based on the modification of the standard settings of the coefficient values of the mathematical description, and taking into account the correct distribution of the scrolls drawing the phase-space diagram. The experimental results show that the DE and PSO algorithms help to maximize the positive Lyapunov exponent of truncated coefficients over the continuous spaces.

Chapter 4, “Optimization and Cosimulation of an Implantable Telemetric System by Linking System Models to Nonlinear Circuits,” was written by Yao Li, Hao Zou, Yasser Moursy, Ramy Iskander, Robert Sobot, and Marie-Minerve Louërat. It presents a platform for modeling, design, optimization, and cosimulation

of mixed-signal systems using the SystemC-AMS standard. The platform is based on a bottom-up design and top-down simulation methodologies. In the bottom-up design methodology, an optimizer is inserted to perform a knowledge-aware optimization loop. During the process, a Peano trajectory is applied for global exploration and the Nelder–Mead Simplex optimization method is applied for local refinement. The authors introduce an interface between system-level models and their circuit-level realizations in the proposed platform. Moreover, a transient simulation scheme is proposed to simulate nonlinear dynamic behavior of complete mixed-signal systems. The platform is used to design and verify a low-power CMOS voltage regulator for an implantable telemetry system.

Chapter 5, “Framework for Formally Verifying Analog and Mixed-Signal Designs,” was written by Mohamed H. Zaki, Osman Hasan, Sofiène Tahar, and Ghiath Al-Sammame. It proposes a complementary formal-based solution to the verification of analog and mixed-signal (AMS) designs. The authors use symbolic computation to model and verify AMS designs through the application of induction-based model checking. They also propose the use of higher order logic theorem proving to formally verify continuous models of analog circuits. To test and validate the proposed approaches, they developed prototype implementations in Mathematica and HOL and target analog and mixed-signal systems such as delta sigma modulators.

Chapter 6, “Automatic Layout Optimizations for Integrated MOSFET Power Stages,” was written by David Guilherme, Jorge Guilherme, and Nuno Horta. It presents a design automation approach that generates automatically error-free area and parasitic optimized layout views of output power stages consisting of multiple power MOSFETs. The tool combines a multitude of constraints associated with DRC, DFM, ESD rules, current density limits, heat distribution, and placement. It uses several optimization steps based on evolutionary computation techniques that precede a bottom-up layout construction of each power MOSFET, its optimization for area and parasitic minimization, and its optimal placement within the output stage power topology network.

Chapter 7, “Optimizing Model Precision in High Temperatures for Efficient Analog and Mixed-Signal Circuit Design Using Modern Behavioral Modeling Techniques: an Industrial Case Study,” was written by Sahbi Baccar, Timothée Levi, Dominique Dallet, and François Barbara. It deals with the description of a modeling methodology dedicated to simulation of AMS circuits in high temperatures (HT). A behavioral model of an op-amp is developed using VHDL-AMS in order to remedy the inaccuracy of the SPICE model. The precision of the model simulation in HT was improved thanks to the VHDL-AMS model. Almost all known op-amp parameters were inserted into the model which was developed manually. Future work can automate the generation of such a behavioral model to describe the interdependency between different parameters. This is possible by using modern computational intelligence techniques, such as genetic algorithms, or other techniques such as Petri nets or model order reduction.

Chapter 8, “Nonlinearities Behavioral Modeling and Analysis of Pipelined ADC Building Blocks,” was written by Carlos Silva, Philippe Ayzac, Nuno Horta, and Jorge Guilherme. It presents a high-speed simulation tool for the design and analysis of pipelined analog-to-digital converters implemented using the Python programming language. The development of an ADC simulator requires the behavior modeling of the basic building blocks and their possible interconnections to form the final converter. This chapter presents a Pipeline ADC simulator tool which allows topology selection and digital calibration of the frontend blocks. Several block nonlinearities are included in the simulation, such as thermal noise, capacitor mismatch, gain and offset errors, parasitic capacitances, settling errors, and other error sources.

## **Part II—Radio-Frequency Design**

Chapter 9, “SMAS: A Generalized and Efficient Framework for Computationally Expensive Electronic Design Optimization Problems,” was written by Bo Liu, Francisco V. Fernández, Georges Gielen, Ammar Karkar, Alex Yakovlev, and Vic Grout. Many electronic design automation (EDA) problems encounter computationally expensive simulations, making simulation-based optimization impractical for many popular synthesis methods. Not only are they computationally expensive, but some EDA problems also have dozens of design variables, tight constraints, and discrete landscapes. Few available computational intelligence methods can solve them effectively and efficiently. This chapter introduces a surrogate model-aware evolutionary search (SMAS) framework, which is able to use much fewer expensive exact evaluations with comparable or better solution quality. SMAS-based methods for mm-wave integrated circuit synthesis and network-on-chip parameter design optimization are proposed and are tested on several practical problems. Experimental results show that the developed EDA methods can obtain highly optimized designs within practical time limitations.

Chapter 10, “Computational Intelligence Techniques for Determining Optimal Performance Trade-offs for RF Inductors,” was written by Elisenda Roca, Rafael Castro-López, Francisco V. Fernández, Reinier González-Echevarría, Javier Sieiro, Neus Vidal, and José M. López-Villegas. The automatic synthesis of integrated inductors for radio-frequency (RF) integrated circuits is one of the most challenging problems that RF designers have to face. In this chapter, computational intelligence techniques are applied to automatically obtain the optimal performance trade-offs of integrated inductors. A methodology is presented that combines a multiobjective evolutionary algorithm with electromagnetic simulation to get highly accurate results. A set of sized inductors is obtained showing the best performance trade-offs for a given technology. The methodology is illustrated with a complete set of examples where different inductor trade-offs are obtained.

Chapter 11, “RF IC Performance Optimization by Synthesizing Optimum Inductors,” was written by Mladen Božanić and Saurabh Sinha. It reviews inductor theory and describes various integrated inductor options. It also explains why integrated planar spiral inductors are so useful when it comes to integrated RF circuits. Furthermore, the chapter discusses the theory of spiral inductor design,



inductor modeling, and how this theory can be used in inductor synthesis. In the central part of the chapter, the authors present a methodology for synthesis of planar spiral inductors, where numerous geometries are searched through in order to fit various initial conditions.

Chapter 12, “Optimization of RF On-Chip Inductors Using Genetic Algorithms,” was written by Eman Omar Farhat, Kristian Zarb Adami, Owen Casha, and John Abela. It discusses the optimization of the geometry of RF on-chip inductors by means of a genetic algorithm in order to achieve adequate performance. Necessary background theory together with the modeling of these inductors is included in order to aid the discussion. A set of guidelines for the design of such inductors with a good quality factor in a standard CMOS process is also provided. The optimization process is initialized by using a set of empirical formulae in order to estimate the physical parameters of the required structure as constrained by the technology. Then automated design optimization is executed to further improve its performance by means of dedicated software packages. The authors explain how to use state-of-the-art computer-aided design tools in the optimization process and how to efficiently simulate the inductor performance using electromagnetic simulators.

Chapter 13, “Automated System-Level Design for Reliability: RF Front-End Application,” was written by Pietro Maris Ferreira, Jack Ou, Christophe Gaquière, and Philippe Benabes. Reliability is an important issue for circuits in critical applications such as military, aerospace, energy, and biomedical engineering. With the rise in the failure rate in nanometer CMOS, reliability has become critical in recent years. Existing design methodologies consider classical criteria such as area, speed, and power consumption. They are often implemented using post-synthesis reliability analysis and simulation tools. This chapter proposes an automated system design for reliability methodology. While accounting for a circuit’s reliability in the early design stages, the proposed methodology is capable of identifying an RF front-end optimal design considering reliability as a criterion.

Chapter 14, “The Backtracking Search for the Optimal Design of Low-Noise Amplifiers,” was written by Amel Garbaya, Mouna Kotti, Mourad Fakhfakh, and Patrick Siarry. The backtracking search algorithm (BSA) was recently developed. It is an evolutionary algorithm for real-valued optimization problems. The main feature of BSA vis-à-vis other known evolutionary algorithms is that it has a single control parameter. It has also been shown that it has a better convergence behavior. In this chapter, the authors deal with the application of BSA to the optimal design of RF circuits, namely low-noise amplifiers. BSA performances, viz. robustness and speed, are checked against the widely used particle swarm optimization technique, and other published approaches. ADS simulation results are given to show the viability of the obtained results.

Chapter 15, “Design of Telecommunications Receivers Using Computational Intelligence Techniques,” was written by Laura-Nicoleta Ivanciu and Gabriel Oltean. It proposes system-, block- and circuit-level design procedures that use computational intelligence techniques, taking into consideration the specifications for telecommunications receivers. The design process starts with selecting the proper architecture (topology) of the system, using a fuzzy expert solution. Next, at

the block level, the issue of distributing the parameters across the blocks is solved using a hybrid fuzzy–genetic algorithms approach. Finally, multiobjective optimization using genetic algorithms is employed in the circuit-level design. The proposed methods were tested under specific conditions and have proved to be robust and trustworthy.

Chapter 16, “Enhancing Automation in RF Design Using Hardware Abstraction,” was written by Sabeur Lafi, Ammar Kouki, and Jean Belzile. It presents advances in automating RF design through the adoption of a framework that tackles primarily the issues of automation, complexity reduction, and design collaboration. The proposed framework consists of a design cycle along with a comprehensive RF hardware abstraction strategy. Being a model-centric framework, it captures each RF system using an appropriate model that corresponds to a given abstraction level and expresses a given design perspective. It also defines a set of mechanisms for the transition between the models defined at different abstraction levels, which contributes to the automation of design stages. The combination of an intensive modeling activity and a clear hardware abstraction strategy through a flexible design cycle introduces intelligence, enabling higher design automation, and agility.

Chapter 17, “Optimization Methodology Based on IC Parameter for the Design of Radio-Frequency Circuits in CMOS Technology,” was written by Abdellah Idrissi Ouali, Ahmed El Oualkadi, Mohamed Moussaoui, and Yassin Laaziz. It presents a computational methodology for the design optimization of ultra-low-power CMOS radio-frequency front-end blocks. The methodology allows us to explore MOS transistors in all regions of inversion. The power level is set as an input parameter before we begin the computational process involving other aspects of the design performance. The approach consists of trade-offs between power consumption and other radio-frequency performance parameters. This can help designers to seek quickly and accurately the initial sizing of the radio-frequency building blocks while maintaining low levels of power consumption. A design example shows that the best trade-offs between the most important low-power radio-frequency performances occur in the moderate inversion region.

## **Volume 2—Computational Intelligence in Digital and Network Designs and Applications**

### **Part I—Digital Design**

Chapter 1, “Sizing Digital Circuits Using Convex Optimization Techniques,” was written by Logan Rakai and Amin Farshidi. It collects recent advances in using convex optimization techniques to perform sizing of digital circuits. Convex optimization techniques provide an undeniably attractive promise: The attained solution is the best available. In order to use convex optimization techniques, the target optimization problem must be modeled using convex functions. The gate sizing problem has been modeled in different ways to enable the use of convex optimization techniques, such as linear programming and geometric programming. Statistical and robust sizing methods are included to reflect the importance of

optimization techniques that are aware of variations. Applications of multiobjective optimization techniques that aid designers in evaluating the trade-offs are described.

Chapter 2, “A Fabric Component Based Approach to the Architecture and Design Automation of High-Performance Integer Arithmetic Circuits on FPGA,” was written by Ayan Palchaudhuri and Rajat Subhra Chakraborty. FPGA-specific primitive instantiation is an efficient approach for design optimization to effectively utilize the native hardware primitives as building blocks. Placement steps also need to be constrained and controlled to improve the circuit critical path delay. Here, the authors present optimized implementations of certain arithmetic circuits and pseudorandom sequence generator circuits to indicate the superior performance scalability achieved using the proposed design methodology in comparison with the circuits of identical functionality realized using other existing FPGA CAD tools or design methodologies. The hardware description language specifications as well as the placement constraints can be automatically generated. A GUI-based CAD tool has been developed, that is integrated with the Xilinx Integrated Software Environment for design automation of circuits from user specifications.

Chapter 3, “Design Intelligence for Interconnection Realization in Power-Managed SoCs,” was written by Houman Zarrabi, A.J. Al-Khalili, and Yvon Savaria. Here, various intelligent techniques for modeling, design, automation, and management of on-chip interconnections in power-managed SoCs are described, including techniques that take into account various technological parameters such as cross talk. Such intelligent techniques guarantee that the integrated interconnections, used in power-managed SoCs, are well-designed, energy-optimal and meet the performance objectives in all the SoCs operating states.

Chapter 4, “Introduction to Optimization Under Uncertainty Techniques for High-Performance Multicore Embedded Systems Compilation,” was written by Oana Stan and Renaud Sirdey. The compilation process design for massively parallel multicore embedded architectures requires solving a number of difficult optimization problems, nowadays solved mainly using deterministic approaches. However, one of the main characteristics of these systems is the presence of uncertain data, such as the execution times of the tasks. The authors consider that embedded systems design is one of the major domains for which applying optimization under uncertainty is legitimate and highly beneficial. This chapter introduces the most suitable techniques from the field of optimization under uncertainty for the design of compilation chains and for the resolution of associated optimization problems.

Chapter 5, “Digital IIR Filter Design with Fix-Point Representation Using Effective Evolutionary Local Search Enhanced Differential Evolution,” was written by Yu Wang, Weishan Dong, Junchi Yan, Li Li, Chunhua Tian, Chao Zhang, Zhihu Wang, and Chunyang Ma. Previously, the parameters of digital IIR filters were encoded with floating-point representations. It is known that a fixed-point representation can effectively save computational resources and is more convenient for direct realization on hardware. Inherently, compared with floating-point repre-

sentation, fixed-point representation may make the search space miss much useful gradient information and, therefore, raises new challenges. In this chapter, the universality of DE-based MA is improved by implementing more efficient evolutionary algorithms (EAs) as the local search techniques. The performance of the newly designed algorithm is experimentally verified in both function optimization tasks and digital IIR filter design problems.

Chapter 6, “Applying Operations Research to Design for Test Insertion Problems,” was written by Yann Kieffer and Lilia Zaourar. Enhancing electronic circuits with ad hoc testing circuitry—so-called design for test (DFT)—is a technique that enables one to thoroughly test circuits after production. But this insertion of new elements itself may sometimes be a challenge, for bad choices could lead to unacceptable degradations of features of the circuit, while good choices may help reduce testing costs and circuit production costs. This chapter demonstrates how methods from operations research—a scientific discipline rooted in both mathematics and computer science, leaning strongly on the formal modeling of optimization issues—help us to address such challenges and build efficient solutions leading to real-world solutions that may be integrated into electronic design software tools.

## **Part II—Network Design**

Chapter 7, “Low-Power NoC Using Optimum Adaptation,” was written by Sayed T. Muhammad, Rabab Ezz-Eldin, Magdy A. El-Moursy, Ali A. El-Moursy, and Amr M. Refaat. Two power-reduction techniques are exploited to design a low-leakage-power NoC switch. First, the adaptive virtual channel (AVC) technique is presented as an efficient way to reduce the active area using a hierarchical multiplexing tree of VC groups. Second, power gating reduces the average leakage power consumption of the switch by controlling the supply power of the VC groups. The traffic-based virtual channel activation (TVA) algorithm is presented to determine traffic load status at the NoC switch ports. The TVA algorithm optimally utilizes virtual channels by deactivating idle VC groups to guarantee high-leakage-power saving without affecting the NoC throughput.

Chapter 8, “Decoupling Network Optimization by Swarm Intelligence,” was written by Jai Narayan Tripathi and Jayanta Mukherjee. Here, the problem of decoupling network optimization is discussed in detail. Swarm intelligence is used for maintaining power integrity in high-speed systems. The optimum number of capacitors and their values are selected to meet the target impedance of the system.

## **Part III—Applications**

Chapter 9, “The Impact of Sensitive Inputs on the Reliability of Nanoscale Circuits,” was written by Usman Khalid, Jahanzeb Anwer, Nor H. Hamid, and Vijanth S. Asirvadam. As CMOS technology scales to nanometer dimensions, its performance and behavior become less predictable. Reliability studies for nanocircuits and systems become important when the circuit’s outputs are affected by its sensitive noisy inputs. In conventional circuits, the impact of the inputs on

reliability can be observed by the deterministic input patterns. However, in nano-scale circuits, the inputs behave probabilistically. The Bayesian networks technique is used to compute the reliability of a circuit in conjunction with the Monte Carlo simulations approach which is applied to model the probabilistic inputs and ultimately to determine sensitive inputs and worst-case input combinations.

Chapter 10, “Pin Count and Wire Length Optimization for Electrowetting-on-Dielectric Chips: A Metaheuristics-Based Routing Algorithm,” was written by Mohamed Ibrahim, Cherif Salama, M. Watheq El-Kharashi, and Ayman Wahba. Electrowetting-on-dielectric chips are gaining momentum as efficient alternatives to conventional biochemical laboratories due to their flexibility and low power consumption. In this chapter, we present a novel two-stage metaheuristic algorithm to optimize electrode interconnect routing for pin-constrained chips. The first stage models channel routing as a traveling salesman problem and solves it using the ant colony optimization algorithm. The second stage provides detailed wire routes over a grid model. The algorithm is benchmarked over a set of real-life chip specifications. On average, comparing our results to previous work, we obtain reductions of approximately 39 and 35 % on pin count and total wire length, respectively.

Chapter 11, “Quantum Dot Cellular Automata: A Promising Paradigm Beyond Moore,” was written by Kunal Das, Arijit Dey, Dipannita Podder, Mallika De, and Debashis De. The quantum dot cellular automata (QCA) is a promising paradigm to overcome the ever-growing needs in size, power, and speed. In this chapter, we explore charge-confined low-power optimum logic circuit design to enhance the computing performance of a novel nanotechnology architecture, the quantum dot cellular automata. We investigate robust and reliable diverse logic circuit design, such as hybrid adders and other binary adder schemes, among them bi-quinary and Johnson–Möbius, in QCA. We also examine zero-garbage lossless online-testable adder design in QCA. Multivalued logic circuit design, with potential advantages such as greater data storage, fast arithmetic operation, and the ability to solve nonbinary problems, will be important in multivalued computing, especially in the ternary computing paradigm.

Chapter 12, “Smart Videocapsule for Early Diagnosis of Colorectal Cancer: Toward Embedded Image Analysis,” was written by Quentin Angermann, Aymeric Histace, Olivier Romain, Xavier Dray, Andrea Pinna, and Bertrand Granado. Wireless capsule endoscopy (WCE) enables screening of the gastrointestinal tract by a swallowable imaging system. However, contemporary WCE systems have several limitations—battery, low processing capabilities, among others—which often result in low diagnostic yield. In this chapter, after a technical presentation of the components of a standard WCE, the authors discuss the related limitations and introduce a new concept of smart capsule with embedded image processing capabilities based on a boosting approach using textural features. We discuss the feasibility of the hardware integration of the detection–recognition method, also with respect to the most recent FPGA technologies.

Finally, the editors wish to use this opportunity to thank all the authors for their valuable contributions, and the reviewers for their help for improving the quality of the contributions.

The editors are also thankful to Ronan Nugent, Springer Senior Editor, for his support, and for his continuous help.

Enjoy reading the book.

Sfax  
Puebla  
Paris  
December 2014

Mourad Fakhfakh  
Esteban Tlelo-Cuautle  
Patrick Siarry

# Contents

## Part I Analog and Mixed-Signal Applications

<b>1</b>	<b>I-Flows: A Novel Approach to Computational Intelligence for Analog Circuit Design Automation Through Symbolic Data Mining and Knowledge-Intensive Reasoning . . . . .</b>	<b>3</b>
	Fanshu Jiao, Sergio Montano, Cristian Ferent and Alex Doboli	
<b>2</b>	<b>Automatic Synthesis of Analog Integrated Circuits Including Efficient Yield Optimization . . . . .</b>	<b>29</b>
	Lucas C. Severo, Fabio N. Kepler and Alessandro G. Girardi	
<b>3</b>	<b>Application of Computational Intelligence Techniques to Maximize Unpredictability in Multiscroll Chaotic Oscillators . . . . .</b>	<b>59</b>
	Victor Hugo Carbajal-Gómez, Esteban Tlelo-Cuautle and Francisco V. Fernández	
<b>4</b>	<b>Optimization and Co-simulation of an Implantable Telemetry System by Linking System Models to Nonlinear Circuits . . . . .</b>	<b>83</b>
	Yao Li, Hao Zou, Yasser Moursy, Ramy Iskander, Robert Sobot and Marie-Minerve Louërat	
<b>5</b>	<b>Framework for Formally Verifying Analog and Mixed-Signal Designs . . . . .</b>	<b>115</b>
	Mohamed H. Zaki, Osman Hasan, Sofiène Tahar and Ghiath Al-Sammame	

<b>6</b>	<b>Automatic Layout Optimizations for Integrated MOSFET Power Stages</b> . . . . .	147
	David Guilherme, Jorge Guilherme and Nuno Horta	
<b>7</b>	<b>Optimizing Model Precision in High Temperatures for Efficient Analog and Mixed-Signal Circuit Design Using Modern Behavioral Modeling Technique: An Industrial Case Study</b> . . . . .	177
	Sahbi Baccar, Timothée Levi, Dominique Dallet and François Barbara	
<b>8</b>	<b>Nonlinearities Behavioral Modeling and Analysis of Pipelined ADC Building Blocks</b> . . . . .	217
	Carlos Silva, Philippe Ayzac, Nuno Horta and Jorge Guilherme	
 <b>Part II Radio-Frequency Design</b>		
<b>9</b>	<b>SMAS: A Generalized and Efficient Framework for Computationally Expensive Electronic Design Optimization Problems</b> . . . . .	251
	Bo Liu, Francisco V. Fernández, Georges Gielen, Ammar Karkar, Alex Yakovlev and Vic Grout	
<b>10</b>	<b>Computational Intelligence Techniques for Determining Optimal Performance Trade-Offs for RF Inductors</b> . . . . .	277
	Elisenda Roca, Rafael Castro-López, Francisco V. Fernández, Reinier González-Echevarría, Javier Sieiro, Neus Vidal and José M. López-Villegas	
<b>11</b>	<b>RF IC Performance Optimization by Synthesizing Optimum Inductors</b> . . . . .	297
	Mladen Božanić and Saurabh Sinha	
<b>12</b>	<b>Optimization of RF On-Chip Inductors Using Genetic Algorithms</b> . . . . .	331
	Eman Omar Farhat, Kristian Zarb Adami, Owen Casha and John Abela	
<b>13</b>	<b>Automated System-Level Design for Reliability: RF Front-End Application</b> . . . . .	363
	Pietro Maris Ferreira, Jack Ou, Christophe Gaquière and Philippe Benabes	



**14 The Backtracking Search for the Optimal Design of Low-Noise Amplifiers . . . . . 391**  
Amel Garbaya, Mouna Kotti, Mourad Fakhfakh  
and Patrick Siarry

**15 Design of Telecommunication Receivers Using Computational Intelligence Techniques . . . . . 413**  
Laura-Nicoleta Ivanciu and Gabriel Oltean

**16 Enhancing Automation in RF Design Using Hardware Abstraction . . . . . 439**  
Sabeur Lafi, Ammar Kouki and Jean Belzile

**17 Optimization Methodology Based on IC Parameter for the Design of Radio-Frequency Circuits in CMOS Technology . . . . . 471**  
Abdellah Idrissi Ouali, Ahmed El Oualkadi, Mohamed Moussaoui  
and Yassin Laaziz

# Contributors

**John Abela** University of Malta, Msida, Malta

**Kristian Zarb Adami** University of Oxford, Oxford, UK

**Ghiath Al-Sammam** Concordia University, Montreal, Québec, Canada

**Philippe Ayzac** Thales Alenia Space, Toulouse, France

**Sahbi Baccar** IRSEEM Laboratory-ESIGELEC, Rouen, France

**François Barbara** Schlumberger Etudes et Productions, Schlumberger, France

**Jean Belzile** MEIE, Québec, Canada

**Philippe Benabes** Department of Electronic Systems, GeePs, UMR CNRS 8507, CentraleSupélec - Campus Gif, Gif-sur-Yvette, France

**Mladen Božanić** Department of Electrical and Electronic Engineering Science, Faculty of Engineering and the Built Environment, University of Johannesburg, Johannesburg, South Africa

**Victor Hugo Carbajal-Gómez** INAOE, Puebla, Mexico

**Owen Casha** University of Malta, Msida, Malta

**Rafael Castro-López** Instituto de Microelectrónica de Sevilla, IMSE-CNM, CSIC and Universidad de Sevilla, Seville, Spain

**Dominique Dallet** IMS-Laboratory, Bordeaux INP, Bordeaux, France

**Alex Doboli** Department of ECE, Stony Brook University, Stony Brook, NY, USA

**Mourad Fakhfakh** National School of Electronics and Telecommunications of Sfax, University of Sfax, Sfax, Tunisia

**Eman Omar Farhat** University of Malta, Msida, Malta

**Cristian Ferent** Department of ECE, Stony Brook University, Stony Brook, NY, USA

**Francisco V. Fernández** Instituto de Microelectrónica de Sevilla, IMSE-CNM, CSIC and Universidad de Sevilla, Seville, Spain

**Pietro Maris Ferreira** Department of Electronic Systems, GeePs, UMR CNRS 8507, CentraleSupélec - Campus Gif, Gif-sur-Yvette, France

**Christophe Gaquière** IEMN, UMR CNRS 8520, Department of DHS, Lille-1 University, Lille, France

**Amel Garbaya** National School of Electronics and Telecommunications of Sfax, University of Sfax, Sfax, Tunisia

**Georges Gielen** ESAT-MICAS, Katholieke Universiteit Leuven, Leuven, Belgium

**Alessandro G. Girardi** Alegrete Technology Campus, Federal University of Pampa, Alegrete-RS, Brazil

**Reinier González-Echevarría** Instituto de Microelectrónica de Sevilla, IMSE-CNM, CSIC and Universidad de Sevilla, Seville, Spain

**Vic Grout** Department of Computing, Glyndwr University, Wrexham, UK

**David Guilherme** Instituto de Telecomunicações, Lisbon, Portugal

**Jorge Guilherme** Instituto Politécnico de Tomar, Tomar, Portugal

**Osman Hasan** National University of Sciences and Technology, Islamabad, Pakistan

**Nuno Horta** Instituto de Telecomunicações, Lisbon, Portugal

**Ramy Iskander** Université Pierre et Marie Curie, Paris, France

**Laura-Nicoleta Ivanciu** Technical University of Cluj-Napoca, Cluj-Napoca, Romania

**Fanshu Jiao** Department of ECE, Stony Brook University, Stony Brook, NY, USA

**Ammar Karkar** School of Electrical Electronic and Computer Engineering, Newcastle University, Newcastle, UK

**Fabio N. Kepler** Alegrete Technology Campus, Federal University of Pampa, Alegrete, RS, Brazil

**Mouna Kotti** High School of Sciences and Technologies of Hammam Sousse, University of Sousse, Sousse, Tunisia

**Ammar Kouki** LACIME, ÉTS, Montréal, Canada

**Yassin Laaziz** LabTIC, National School of Applied Sciences of Tangier, Abdelmalek Essaadi University, ENSA Tanger, Tangier, Morocco

**Sabeur Lafi** LACIME, ÉTS, Montréal, Canada

**Timothée Levi** IMS-Laboratory, University of Bordeaux, Bordeaux, France

**Yao Li** Université Pierre et Marie Curie, Paris, France

**Bo Liu** Department of Computing, Glyndwr University, Wrexham, UK

**Marie-Minerve Louérat** Université Pierre et Marie Curie, Paris, France

**José M. López-Villegas** Departament d'Electrònica, Universitat de Barcelona, Barcelona, Spain

**Sergio Montano** Department of ECE, Stony Brook University, Stony Brook, NY, USA

**Yasser Moursy** Université Pierre et Marie Curie, Paris, France

**Mohamed Moussaoui** LabTIC, National School of Applied Sciences of Tangier, Abdelmalek Essaadi University, ENSA Tanger, Tangier, Morocco

**Gabriel Oltean** Technical University of Cluj-Napoca, Cluj-Napoca, Romania

**Jack Ou** Department of Electrical and Computer Engineering, California State University Northridge, Northridge, CA, USA

**Abdellah Idrissi Ouali** LabTIC, National School of Applied Sciences of Tangier, Abdelmalek Essaadi University, ENSA Tanger, Tangier, Morocco

**Ahmed El Oualkadi** LabTIC, National School of Applied Sciences of Tangier, Abdelmalek Essaadi University, ENSA Tanger, Tangier, Morocco

**Elisenda Roca** Instituto de Microelectrónica de Sevilla, IMSE-CNM, CSIC and Universidad de Sevilla, Seville, Spain

**Lucas C. Severo** Alegrete Technology Campus, Federal University of Pampa, Alegrete-RS, Brazil

**Patrick Siarry** Laboratoire LiSSi (EA 3956), Université Paris-Est Créteil, Créteil, France

**Javier Sieiro** Departament d'Electrònica, Universitat de Barcelona, Barcelona, Spain

**Carlos Silva** Portugal Telecom, Lisbon, Portugal

**Saurabh Sinha** Faculty of Engineering and the Built Environment, University of Johannesburg, Johannesburg, South Africa

**Robert Sobot** The University of Western Ontario, London, ON, Canada; ENSA/ETIS, University of Cergy-Pontoise, Cergy-Pontoise, France

**Sofiène Tahar** Concordia University, Montreal, QC, Canada

**Esteban Tlelo-Cuautle** INAOE, Puebla, Mexico

**Neus Vidal** Departament d'Electrònica, Universitat de Barcelona, Barcelona, Spain

**Alex Yakovlev** School of Electrical Electronic and Computer Engineering, Newcastle University, Newcastle, UK

**Mohamed H. Zaki** Concordia University, Montreal, QC, Canada

**Hao Zou** Université Pierre et Marie Curie, Paris, France

**Part I**  
**Analog and Mixed-Signal Applications**

# Chapter 1

## I-Flows: A Novel Approach to Computational Intelligence for Analog Circuit Design Automation Through Symbolic Data Mining and Knowledge-Intensive Reasoning

Fanshu Jiao, Sergio Montano, Cristian Ferent and Alex Doboli

**Abstract** This chapter presents an overview of the authors' ongoing work toward devising a new approach to analog circuit synthesis. The approach computationally implements some of the facets of knowledge-intensive reasoning that humans perform when tackling new design problems. This is achieved through a synthesis flow that mimics reasoning using a domain-specific knowledge structure with two components: an associative part and a causal reasoning part. The associative part groups known circuit schematics into abstractions based on the similarities and differences of their structural features. The causal reasoning component includes the starting ideas as well as the design sequences that create the existing circuits.

### 1.1 Introduction

Research in cognitive psychology suggests that human reasoning relies on organized knowledge structures to perform activities such as concept comparison and identification, concept learning, and problem solving through deduction and induction [1–4]. In particular, analog circuit design mainly depends on the designers' expertise and ability to create new designs by combining basic devices,

---

F. Jiao (✉) · S. Montano · C. Ferent · A. Doboli  
Department of ECE, Stony Brook University, Stony Brook, NY 11794-2350, USA  
e-mail: fanshu.jiao@stonybrook.edu

S. Montano  
e-mail: sergio.montano@stonybrook.edu

C. Ferent  
e-mail: cristian.ferent@stonybrook.edu

A. Doboli  
e-mail: alex.doboli@stonybrook.edu

sub-circuits, and ideas from similar solutions as the source for innovation. These activities are arguably hard to replicate through optimization-based methods, which have been traditionally popular in analog circuit EDA [5–9]. This limits the effectiveness of current EDA methods for activities such as circuit topology selection, topology refinement through feature reuse, design retargeting, topology design, and design verification.

Analog circuit design is knowledge intensive [10, 11]. Reasoning steps such as abstraction, instantiation, analogies (similarities), induction, concept combination, and constraint relaxation are utilized in complex design strategies (design plans) to tackle a new application. While this process might involve optimization and equation solving, it also includes pattern identification with respect to both the utilized design features and reasoning sequences. In fact, it has been suggested that the human brain is analogous to a sophisticated and effective pattern recognition machine [12]. This observation suggests that the circuit design process realizes an effective traversal of the design space by identifying, using, and reusing various design patterns (design steps). These patterns (or design features) include various topological circuit structures (e.g., differential input structures, cascode structures, buffers, and current sources) as well as specific constraints among circuit parameters (i.e., matched devices). Using certain features is justified by the functional and performance requirements of an application, and the performance trade-offs and bottlenecks of current solutions. Hence, the circuit design process can be seen as a collection of starting ideas (like ideas previously used in similar designs and new insight) followed by a sequence of design steps, in which every step adds a design feature that is causally justified by the need to address a given constraint of the design. We think that extending current analog circuit EDA methods by incorporating activities inspired by human reasoning can leverage the effectiveness and capabilities of automated tools by narrowing the gap between their solutions and human-devised circuits.

The analog circuit design space is complex, nonlinear, tightly coupled, and highly discontinuous. The traversal of the space to find new design features is challenging. As explained in the next section, diversification is a major activity in circuit design every time new structural (topological) features must be identified or invented as the current features cannot tackle well the existing performance bottlenecks. For example, addressing new performance challenges, such as ultrahigh frequencies, low power consumption, and high robustness to process parameter variations, required the creation of new structural features for more effective compensation and adaptation [13–16]. Novel features are a diversification from the current set of circuit features present in the knowledge domain. Effectively identifying the directions for diversification is difficult as traditional diversification approaches, such as focusing on the unexplored regions or introducing random changes into a solution, do not guarantee that an actual bottleneck is addressed. Moreover, such strategies might produce very complex circuit structures even though, often, there is a simpler solution.

This chapter presents an overview of our recent work toward devising a new approach to analog circuit EDA by computationally implementing some of the

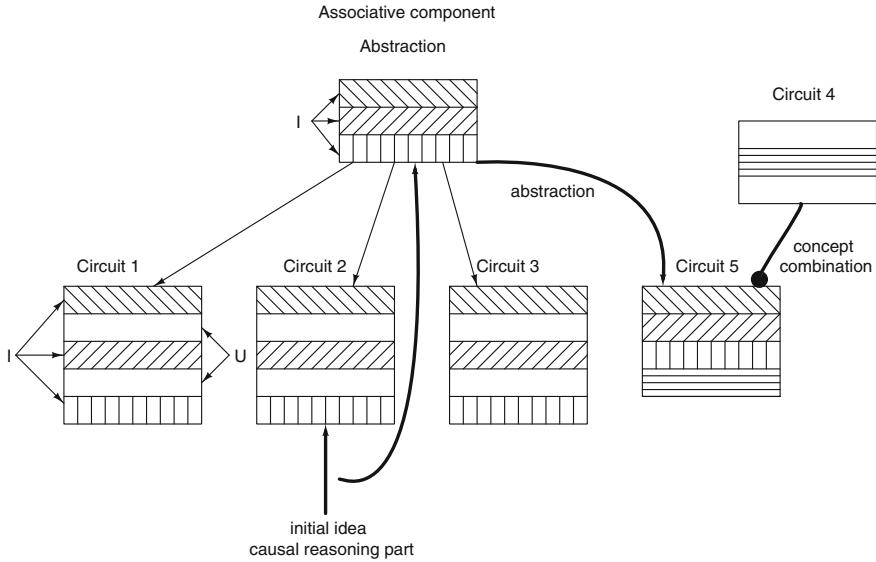


facets of the knowledge-intensive reasoning tasks that humans perform when tackling new problems [17–21]. This is achieved through a synthesis mechanism that mimics reasoning on a domain-specific knowledge structure. The knowledge structure has two components: an associative part and a causal reasoning component. The associative part [17, 19] groups the known circuit schematics into abstractions based on the similarities and differences of their structural features. The associative part also characterizes the performance trade-offs (e.g., gain–bandwidth–noise) and bottlenecks of every circuit instance and abstraction [18] in the associative part. Each abstraction is a summary of the common symbolic expressions that describe the behavior of the related instances. Abstractions support a quick traversal of the solution space during synthesis, whenever a bottleneck must be addressed (e.g., if the abstraction does not include the symbolic expression of the bottleneck, then it includes an alternative topology that does not have the bottleneck).

The reasoning procedure for synthesis is based on operators such as abstraction–instantiation, induction, and concept combination [22, 23]. In addition, the synthesis flow reuses design sequences that have been previously used to solve similar design applications. Such sequences together with the starting design ideas are stored by the causal reasoning component of the knowledge structure. The starting ideas constitute the spark that initiated a design (or the aha moment [3]). A detailed presentation of the algorithms that mine the starting ideas and design sequence of an existing circuit is offered in [21]. This chapter offers a comprehensive yet intuitive presentation of the synthesis routines and the related knowledge structure. In addition, the experimental section gives detailed insight about the automated mining of the starting ideas and design sequences for four high-performance analog circuits.

Figure 1.1 offers an intuitive description of the synthesis flow. It starts by identifying the initial features of the circuit, e.g., a circuit developed for a similar application. Circuit<sub>2</sub> is such a circuit in the figure. As the circuit includes a performance bottleneck (i.e., gain–noise) that prevents it from meeting the requirements, the synthesis approach uses the abstraction operator to move to node Abstraction, which collectively describes Circuit<sub>1</sub>, Circuit<sub>2</sub>, and Circuit<sub>3</sub>. The common features of an abstraction are denoted as set  $I$  (invariant) and the distinguishing features as set  $U$  (uniqueness). Then, the features of the abstraction are combined with some features of Circuit<sub>4</sub> (e.g., its adaptation scheme) to produce Circuit<sub>5</sub>, the final solution. The associative part of the knowledge structure includes the circuit instances and the abstractions. The causal reasoning part presents the reasoning steps (like abstraction followed by concept combination) that produced a new solution.

We think that the presented approach represents a new perspective on synthesis of analog circuits to extend the capabilities of current optimization- and solving-based techniques. Also, the approach is not similar to the older expert system-based synthesis techniques as it is not limited to a static set of built-in if-then rules. It automatically mines new design knowledge, such as in our ongoing work to mine knowledge in real time from electronic documents in databases, such as IEEE Xplore. While the overall synthesis flow is currently under development, encouraging progress has been made on devising the associative component and the



**Fig. 1.1** Synthesis flow based on knowledge-intensive reasoning

causal reasoning part of the knowledge structure, including the examples discussed in this chapter. Finally, we think that this approach fits more naturally the way analog designers reason, hence creating a still very limited yet intriguing path toward computationally mimicking some facets of human creativity, a well-known challenge in computing.

The chapter has the following structure. Section 1.2 discusses the importance of diversification in analog circuit design and the challenges posed by two different types of diversification. Section 1.3 presents the reasoning-based synthesis method and the associative and causal reasoning components of the knowledge structure. References to the detailed presentation of the algorithms are also indicated. Section 1.4 presents new experimental results on mining the initial ideas and design sequences for four state-of-the-art circuits. Finally, conclusions end the chapter.

## 1.2 Motivation: The Importance of Effective Diversification

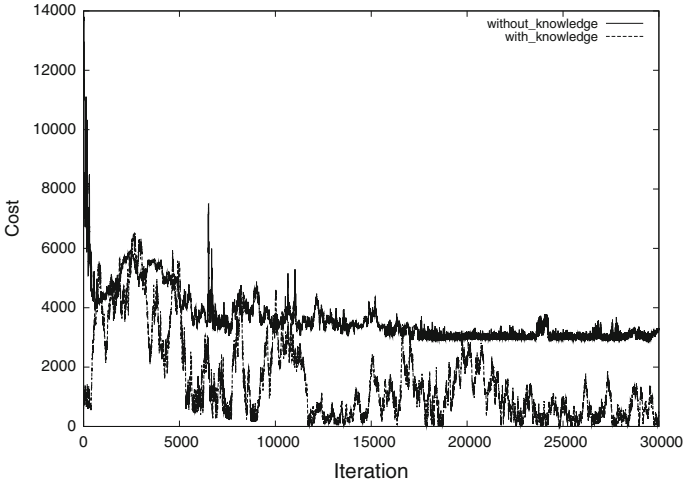
Analog circuit design is knowledge intensive. It is well accepted that the expertise and experience of a designer are critical in deciding the quality of the final solution (e.g., its performance) [10, 11]. The solution space of a circuit design problem is complex, nonlinear, tightly coupled, and highly discontinuous. For example, a circuit design includes a large number of variables, such as the transconductances and capacitances of its devices. The variables are linked through complicated,

nonlinear equations that express the DC, AC, and transient behavior of the circuit [8, 18, 24]. Moreover, the variables are tightly coupled with each other through symbolic (mathematical) expressions that describe a circuit's behavior and performance. Finally, strong discontinuities characterize the solution space as the transitions from a given circuit schematics (topology) to an incremental extension of it can introduce significant changes to the symbolic expressions of the circuit behavior and performance. This section explains that traditional optimization techniques, such as optimization-based methods, often experience difficulties in tackling the complex analog circuit design solution space. Instead, circuit designers use knowledge-intensive reasoning to find effective design plans and strategies to address the above challenges.

We classify the nature of divergence specific to analog circuit design into two types:

- *Closed-ended (or enumerable) diversification*: This diversification type corresponds to situations in which the enumeration of all diversification cases is possible, even though the number of resulting cases can be very large. Hence, the set of diversification situations is enumerable. For example, the solution space for optimizing the parameter values of a circuit includes many local optima. Each parameter sub-range that corresponds to a local optimum defines an area that is divergent (distinct) from another local optimum. The set of possible parameter sub-ranges is enumerable starting from the possible device sizes for a given fabrication process.
- *Open-ended (or non-enumerable) diversification*: This diversification type presents situations in which the diversification cases can be enumerated only with respect to a given set of building blocks and block connection rules. However, the resulting cases do not necessarily express all possible diversification situations as adding new building blocks and/or rules produces new diversification cases. For example, the solution space for the possible circuit topologies can be described using building blocks such as MOSFETs, resistors, and capacitors, as well as rules to connect the building blocks, i.e., series, parallel, and star. Note, however, that new diversifying cases can result by adding new building blocks, e.g., new devices. The resulting diversifications are open-ended as an unbounded (non-enumerable) number of new cases can be created through continuously connecting the building blocks into structures of growing complexity.

In our previous work, we studied optimization algorithms tackling closed-ended diversification [9]. The studied problem was to search for the circuit parameters that optimize the performance of a given analog system, e.g., high-frequency continuous-time filters or high-resolution  $\Delta\Sigma$  analog-to-digital converters (ADCs). As shown in Fig. 1.2, traditional optimization algorithms that exploit some form of gradient descendant methods, such as simulated annealing, have difficulties in finding good-quality parameter values mainly due to the hardness to systematically visit the divergent regions of a solution space. Diversification strategies, such as randomly accepting worse solutions at higher temperatures (in simulated annealing) or forcing the algorithm toward previously unexplored regions (in tabu



**Fig. 1.2** Synthesis convergence with and without knowledge about variable domains [9]

search), did not significantly improve solution quality. This is mainly because interesting diversifying regions have characteristics that are hard to predict from those of the current regions; therefore, it is difficult to identify (compute) the search direction along which such diversifying regions are placed. The history of previously found good-quality regions might offer little insight about future good-quality regions unless insight similar to designer reasoning is mined from the former. However, knowledge about the parameter ranges that are likely to produce high-quality solutions results in a significant improvement of the cost function values as shown in Fig. 1.2.

This conceptual difficulty has been addressed by the proposed algorithm [9] by using three different diversification-related steps: (i) variable domain pruning, (ii) wave front expansion to cover all search directions, and (iii) identifying the parameter correlation patterns specific to the promising search directions. Variable domain pruning eliminates the parameter sub-domains for which the resulting values of the system behavior and performance vary in large ranges. Even though such parameter sub-domains can still produce constraint-satisfying solutions, finding the performance-optimizing parameter values is harder than for the parameter sub-domains for which the resulting behavior and performance presents a lesser variation. Interval arithmetic was used for parameter domain pruning [9]. The second step, wave front expansion, is called every time a convex subspace has been fully explored (using a descending gradient-based search) and diversification is needed. Wave front expansion is initially along all directions for changing the parameter values, but as the number of possible directions grows quickly, it reduces the search complexity by sampling the alternatives using orthogonal arrays. This way of sampling encourages that each parameter sub-range combination has equal chance of being searched. Otherwise, there is no guarantee that diversification

systematically explores all alternatives. Finally, the third step, identifying correlation patterns, further reduces the number of diversifying alternatives based on the observation that high-quality design solutions implement certain required constraints among the parameter values, e.g.,  $p_1 \gg p_2$  or  $p_3 \approx p_4$ . Even though such parameter correlations are probably unavailable at the start of an optimization algorithm (unless designer expertise is employed to find them), correlations can be automatically detected (learned) during wave front expansion. A search direction that embeds a useful parameter correlation produces a rapid drop of the cost function value, while less useful search directions are characterized by either oscillating or plateaulike cost function values [9]. The latter directions are quickly found and dropped during wave front expansion. Experiments for filter and  $\Delta\Sigma$  ADC synthesis showed that using the three steps for closed-ended diversification finds more constraint-satisfying designs as well as designs of better performance, i.e., the method found between 1.8 and 2.6 times more constraint-satisfying solutions than other state-of-the-art tools [9].

Open-ended (or non-enumerable) diversification represents situations in which the diversification cases are enumerated with respect to a given set of building blocks and connection rules, even though new building blocks and/or rules can be added to the set. Traditionally, such diversification situations have been tackled through two methods: template-based search and genetic algorithms. In template-based search [25–27], a structural template describes the characterizing features of the possible circuit topologies, e.g., the main signal flows through the contemplated circuit topologies. The diversification routine uses the set of building blocks and connection rules to generate different topologies but which all realize the same signal flow as the given template. Genetic algorithms (GA) [28–30] achieve diversification through the well-known operators' selection, mutation, and combination (and their extensions), which are sometimes extended with analog design-related steps or constraints. GAs can construct, potentially, an unlimited set of diversifying solutions. However, the arguably unsystematic traversal of the solution space does not guarantee that optimal solutions are found for a problem. Also, the search time might become very large for complex search problems. Moreover, the repeated applying of the three operators can result in very complex structures, which are less common to the topologies that a designer devises through knowledge-intensive reasoning [31]. As explained in [31], traditional topology synthesis, i.e., using GAs, might experience difficulties in producing circuits with novel (hence diversifying) yet useful structural features. The synthesized circuit topologies include unique features, but such features are rarely used by designers.

This chapter proposes a reasoning-based approach for tackling diversification during circuit design. The approach attempts to mimic at a very basic level the knowledge-intensive reasoning process conducted by a designer. The process starts with a set of initial design features which are the starting points (e.g., the initial ideas) for solving the problem. The performance bottlenecks of the initial features are found, i.e., the design features that prevent the design from meeting the needed performance. Then, a sequence of design steps attempts to remove the bottlenecks, such that every step is justified either by addressing an existing performance

limitation or by relaxing constraints, so that future performance-enhancing design steps are possible (steps which are impossible without constraint relaxation). The reasoning-based approach includes two main components, which are discussed in the next section: (i) an *associative component* that presents the domain knowledge specific to analog circuit design and (ii) a *reasoning component* that expresses the various reasoning sequences that were used in creating a circuit solution. Reasoning sequences are reused during the devising of new circuit solutions.

### 1.3 Knowledge Structures

The proposed knowledge-intensive reasoning method for circuit synthesis is shown in Fig. 1.3. More details about the method can be also found in [32, 17]. It incorporates design knowledge-intensive diversification using the associative and reasoning components. As detailed next, the associative part includes *instances* (e.g., actual circuit designs) and *abstract concepts* that summarize the features of a set of instances. The first step of the methodology selects the initial features of the design solution by relating every node  $C_x$  (instance or abstract concept) of the associative part  $St$  to the desired design requirements and then finding the bottlenecks that prevent it from achieving the needed performance. Another way of

```

for each node Cx in structure St {
  find bottlenecks and relate them to
  the topological elements of the circuits
  (i.e. variables, expressions, constraints);
}
C = select node in St;
while d(C, <f, p>) > acceptable {
  if (bottleneck is due to unique features of C) {
    find bottom-up in St first node Cn
    but without expressions that cause the bottleneck;
  }
  if (not successful)
    C = concept-combination (C, St, bottleneck);
  if (not successful) {
    add to bottleneck the features that condition
    the expressions of the bottleneck;
    find bottom-up in St first node Cn
    but without expressions that cause the bottleneck;
  }
  else if (successful){
    find new bottlenecks and relate them to
    the topological elements of the circuits
    (i.e. variables, expressions, constraints);
  }
}

```

**Fig. 1.3** Conceptual description of knowledge-intensive reasoning method [32]

selecting the initial solution features is by selecting a previously solved problem that is most similar to the current requirements and then using its initial features as the starting point of the new design. As explained in the second part of this section, this requires reusing information stored by the reasoning component. Bottlenecks are related to the actual structural elements of a circuit, i.e., its design variables (like device sizes), the symbolic expressions describing the circuit behavior, and the constraints imposed for correct operation. A technique for finding circuit bottlenecks is presented in [18].

Next, the method selects node C, which is the most attractive candidate in implementing the new circuit design. The node describes the initial features used to develop the solution. The iterative process continues as long as the derived solution C does not meet the desired functional and performance requirements expressed as pair  $\langle f, p \rangle$ . Hence, distance  $d$  between the behavior of solution C and pair  $\langle f, p \rangle$  exceeds an acceptable value. Each step of the method attempts three subsequent reasoning steps:

- First, *abstraction* searches bottom up the associative part  $St$  to find an abstract concept corresponding to node C, so that the abstract concept does not include the performance-limiting bottleneck of node C. If the search is successful, it indicates that the bottleneck is specific only to node C but not the abstract concept to which node C corresponds to and for which node C is an implementation. Hence, the abstract concept can include alternative instances to node C (e.g., another circuit topology) without the bottleneck. This instance becomes the current design that is further extended into the final solution.
- Second, if abstraction was unsuccessful, *concept combination* attempts to remove the bottleneck by combining the features of node C with the features of a concept that does not include the bottleneck. For example, the combined features might include various adaptation or compensation mechanisms that are used to improve performances such as bandwidth and linearity.
- Third, if the previous two steps could not find a solution, then *constraint relaxation* tries to lighten up the constraints that guarantee the correct operation of the circuit, i.e., the constraints that set the right device operation regions or the constraints on the device parameter values. This step might then enable concept combinations that otherwise would not be feasible.

The detailed analysis used by the three steps utilizes an automated mechanism for systematically producing comparison data between two analog circuits [18]. The similar and distinguishing performance characteristics of circuits with respect to gain, bandwidth, common-mode gain, noise, and sensitivity are captured. The technique utilizes matching of both topologies and symbolic expressions of the compared circuits to find the nodes with similar behavior. The impact on performance of the unmatched nodes is used to express the differentiating characteristics of the circuits. The produced comparison data are important for getting insight into unique benefits and limitations of a circuit, selecting fitting circuit topologies for system design, and optimizing circuit topologies.

The next two subsections present the two parts used to support knowledge-intensive reasoning by the proposed method, the associative component and the reasoning component.

### 1.3.1 Associative Component

The associative component clusters the existing circuit instances into higher-level descriptions, called abstractions [17, 19]. The clustering is based on the common features (called attributes) of the instances forming a concept as well as the features that offer unique expression of the specific attributes of each instance. These features form a set that is called set  $I$  (invariants) [17]. In addition, a second set, set  $U$  (uniqueness), presents the features that are unique to the abstraction as compared to other abstractions. Finally, set  $E$  (enabling) introduces all conditions and constraints that must be met during the operation of the design. All features are symbolic expressions over variables.

This description is important because the symbolic expressions in set  $I$  indicate the trade-offs and bottlenecks specific to an abstraction. Any instance of the abstraction will include them. Hence, the abstraction step of the reasoning method in Fig. 1.3 checks the existence or not of a bottleneck by analyzing set  $I$  of the abstraction. Moreover, set  $U$  indicates whether a bottleneck is unique for an abstraction, and therefore, diversifying toward other abstractions might result in removing the bottleneck. Finally, set  $E$  is used by the constraint relaxation step of the method to indicate the constraints that are candidates for removal.

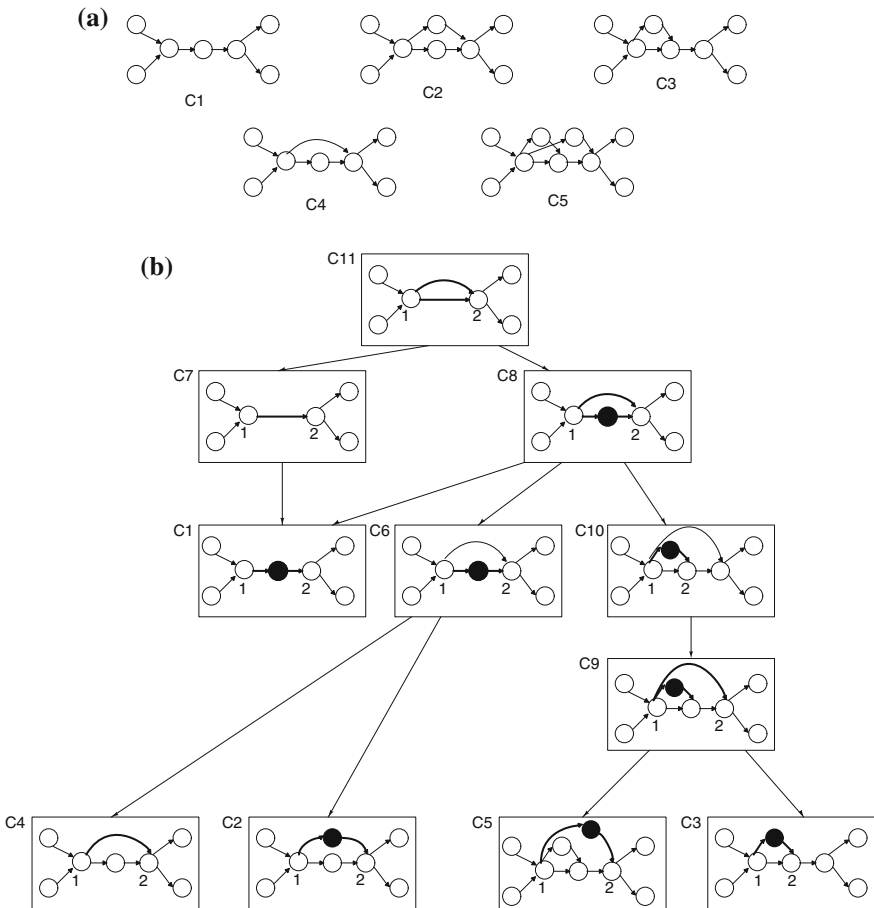
*Example* Let us assume three design instances characterized by the following set of features:  $D_1 = \{x_o = x_1 * x_2; x_1 = 3 * x_i; x_2 = x_1 + x_i\}$ ,  $D_2 = \{x_o = x_1 * x_2; x_1 = 8 * x_i; x_2 = x_1 + x_i\}$ , and  $D_3 = \{x_o = x_1 * x_2; x_1 = 6 * x_i; x_2 = x_1 + x_i\}$ . (For brevity, we did not use the symbolic expressions describing concrete circuit topologies, but the analogy holds.) Each expression is a feature of the design, e.g., it describes the behavior of the circuit. Variables  $x_i$  are the design variables, where  $x_i$  is the input and  $x_o$  is the output. The abstraction of the three instances is expressed by the following features  $D_a = \{x_o = x_1 * x_2; x_1 = A * x_i; x_2 = x_1 + x_i\}$ . Set  $I = \{x_o = x_1 * x_2; x_1 = A * x_i; x_2 = x_1 + x_i\}$ . Note that feature  $x_1 = A * x_i$ , with  $A$  being a symbolic constant, represents the specific features  $x_1$  of each design  $D_i$ . Finally, compared to another abstraction,  $D_b = \{x_o = x_1 * x_2; x_1 = B * x_i; x_2 = x_1 - x_i\}$ , the set  $U$  of  $D_a$  is feature  $x_2 = x_1 + x_i$  as this feature does not occur for  $D_b$ . Set  $E$  might include constraints on the variables, e.g.,  $x_i \in (-1.0, 1.0)$ .

The description of similar instances as abstractions, of similar abstraction as higher-level abstractions, and so on, produces a hierarchical, graphlike associative component, as shown in Fig. 1.4. Arrows indicate instances of an abstraction, and arrows with bubbled heads represent combined features. Note that the representation is a graph (not a tree) as instances can belong to multiple abstractions.



The number of levels in the hierarchy depends on the sampling of the features of the available designs, e.g., the alternative symbolic expressions of similar features, the way in which one feature is combined with other features, and the function and purpose (goal) of a design.

**Example** Figure 1.4b presents the associative component built for the five circuits shown in Fig. 1.4a. Each circuit is described by its signal flow graph, in which the nodes are the circuit nodes, arcs indicate the coupling among nodes, and poles are attached to every node. Every arc is labeled by the symbolic expression of the coupling among nodes. The symbolic expressions of the poles and couplings are computed using the algorithm presented in [24, 33]. For brevity, the symbolic



**Fig. 1.4** Associative component of the knowledge structure [19]

expressions were not shown in the figure. Nodes  $C_6$ – $C_{11}$  in Fig. 1.4b are the abstractions computed for the five circuits. The invariant set (set  $I$ ) of a circuit includes all the symbolic expressions of the circuit poles and couplings through the arcs. The unique features of the instances and abstractions (set  $U$ ) correspond to the symbolic expressions of the node poles and arcs shown in bold. They represent the distinguishing elements of the instances or abstractions as compared to the other alternatives that have the same parent in the graph.

In addition to circuit comparison, the proposed concept structure includes three other symbolic operators: circuit concept instantiation–abstraction, circuit concept combination, and design feature induction [22, 23]. The operators algorithmically implement the steps of the method in Fig. 1.3. The first operator constructs the associative component for a set of known solutions, while the later two provide the mechanisms to extend the knowledge representation to find novel solutions. Instantiation–abstraction organizes the features at various levels by replacing signals or blocks in a design through clusters of signals or blocks with the same behavior [22]. The concept combination operator produces a new circuit concept for an application by mixing the features of two existing circuits, so that resulting performance is improved [20]. The generic concept induction operator uses the existing information on design feature variety from all concepts in the structure to create novel concepts that have not yet been explored, such as different connection patterns among signal nodes that can relax trade-offs [22, 23].

Our ongoing work attempts to update the associative component, so that it gets updated in real time every time a new circuit is published in the literature. A crawler continuously scans databases for scientific literature on analog circuit design, e.g., journals and conference proceedings published by IEEE Xplore, and downloads the papers presenting new circuits. The schematics of the circuit is automatically identified in the pdf file and then converted into its Spice description. After constructing its macromodel (based on the Spice netlist) as presented in [24], the circuit is then added to the associative component by using the clustering method proposed in [19] as well as the symbolic circuit comparison algorithm in [18].

### 1.3.2 Causal Reasoning Component

Each abstraction of the associative component is a *branching point* (BP) in a reasoning flow as the designer could have possibly adopted a different alternative to realize a solution. Obviously, a different alternative might produce different outcomes and performance. Each BP is characterized by a number of variables (e.g., the feature variables in sets  $I$  and  $U$  of the corresponding abstraction), the coupling between variables (e.g., the number of shared variables between the features), and performance values that can be achieved. The causal component of the knowledge structure indicates the reasons that justified the designer’s selection of a particular alternative [21]. Reasons include improvement of the performance capabilities of a

design (e.g., by changing its bottlenecks), modifying the functionality (i.e., the performed processing of the input signals), and relaxing the design constraints of set  $E$  (enabling conditions).

As already explained, the knowledge-intensive reasoning method in Fig. 1.3 builds new design solutions starting from initial design features (e.g., initial ideas) and then employing a design sequence, which are a set of reasoning steps that ultimately produce a circuit design. Each step in a design sequence must be *justified*, meaning that it implements a structural feature of an abstraction or it enhances the performance of another circuit design. Design sequences are design plans (or reasoning strategies) that are reused by the method in Fig. 1.3 during circuit synthesis.

The causal component includes the initial design features and design sequences specific to the circuit instances that are stored by the associative component. A detailed presentation of the algorithms that mine initial ideas and design sequences for a given circuit is offered in [21]. We offer next a summary of the two algorithms. The experimental section illustrates the two mining algorithms for several modern, high-performance analog circuits.

The *algorithm for mining the initial design features* of an analog circuit attempts to identify what were reasonably the starting ideas (of the authors) when devising the circuit. While the complete, iterative process for devising a new circuit might be impossible to infer, we think that at least the ideas (of the last iteration) that lead to final solution can be automatically mined. The algorithm first identifies all the structural features of a given circuit by matching its device connections to the device connections present in the associative component. This set is denoted as set  $\Sigma$  (also called complete set). Next, the algorithm implements the observation that all features in set  $\Sigma$  should be justified by a design sequence starting from the set of initial features. Hence, the set of initial features are those that allow the finding of a design sequence of justified steps (hence, steps either which instantiate more abstract features or which improve the performance of the circuit). Moreover, the set of initial ideas includes (i) features that were previously used by the same authors in devising new circuits (called set  $A$ ), (ii) features that appear in the cited papers (called set  $I$ ), and (iii) features that indicate new insight by the authors. The features of the first two categories can be found by analyzing previously published designs as well as any cited circuits. They represent the tentative set of initial features. Then, the algorithm attempts to find whether all the features of the complete set  $\Sigma$  can be justified starting from the current set of initial features. If this is successful, then the current set is also the final set of initial features. However, if there are any unjustified features, then the algorithm finds the minimum subset among the unjustified features which then justify all the remaining structural features. The minimum set is added to sets  $A$  and  $I$  to represent the complete set of initial features.

The *algorithm for constructing a design sequence* takes as input the current set of initial features and then explores the maximal set of features in the complete set  $\Sigma$  that can be justified based on the initial features. It iteratively finds the features that

can be immediately justified from the initial set, and then, it finds the features that can be justified based on the features found by the previous step and so on until no more features in set  $\Sigma$  can be identified.

### 1.4 Experiments

This section presents four case studies to illustrate the mining of the initial features (ideas) and design sequences that were used in creating a circuit. The procedures use causal reasoning, e.g., understanding the cause—effect relations that produce the performance bottlenecks of a circuit topology and how the design steps of the sequence relax the performance trade-offs.

#### 1.4.1 Circuit 1

The circuit in Fig. 1.5 is a highly linear, fully differential OTA for high-frequency, continuous-time low-pass filters [15]. The complete set of features (set  $\Sigma$ ) includes all structural features of the circuit: three differential pair input stage (a cross-coupled quad cell-based input stage together with an additional linearizing symmetrical

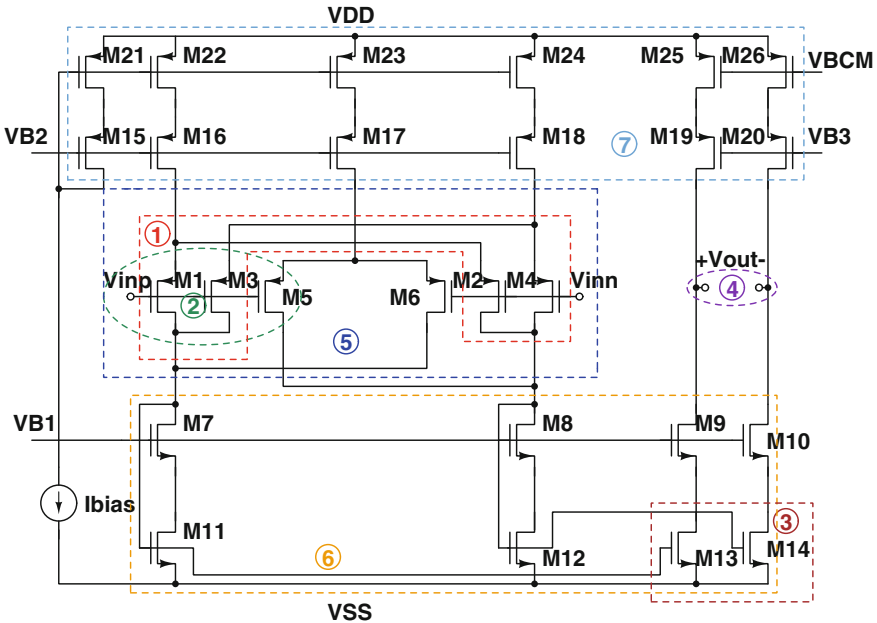
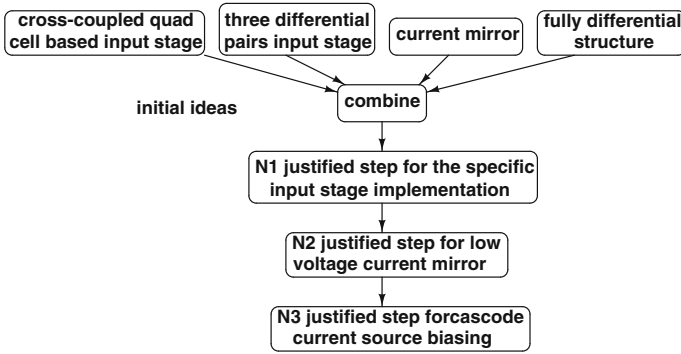


Fig. 1.5 Circuit 1



**Fig. 1.6** Initial ideas and design sequences for circuit 1

differential pair), low voltage current mirror, fully differential structure, and cascode current source biasing.

Figure 1.6 illustrates the mined starting ideas and the corresponding design sequence. The starting features of the design (set  $S$ ) include combining cross-coupled quad cell-based input stage (set  $A$ ), three differential pair input stage (set  $A$ ), current mirror at second stage (set  $A$ ), and fully differential structure (resulting from designer insight). The starting features correspond to structures labeled as 1, 2, 3, and 4 in Fig. 1.5. Beginning with the starting features in set  $S$ , the uncovered features of the circuit are computed by the difference: set  $\Sigma$ —set  $S$ . These features are labeled as 5, 6, and 7 in the figure. They are used to mine the design sequence that added these features to the circuit. For each feature, the analysis for justifying the corresponding step starts first with the more abstract features as well as the more likely features. The design steps of the sequence are as follows.

Design step N1 combines a three differential pair input stage, a cross-coupled quad cell-based input stage together with an additional linearizing symmetrical differential pair. The three differential input pair is justified by a previous and similar design (cited by the paper) that uses a cross-coupled quad cell-based input stage. The additional symmetrical differential pair realizes linear CMOS transconductance elements. To justify the need of having this structure as part of step N1, the transient response of this circuit was compared with a straightforward reference, the circuit with single differential pair input (OTA2). Table 1.1 summarizes the total harmonic distortion results. Transistor sizing followed the design constraints presented in the paper. Both circuits are configured with the same biasing current and input voltage. A 10-MHz sine wave simulation shows that circuit 1 (OTA) has 0.995 % total harmonic distortion. OTA2 with single differential input has 8.185 % total harmonic distortion, which is more than 8 times worse.

Design step N2 adds a low voltage current mirror to the circuit. Using a low voltage current mirror instead of a single current mirror is due to its high output resistance and reduced drain–source voltage (only 0.4-V margin is left across devices M9 and M13 for keeping both of them in saturation). In order to illustrate

**Table 1.1** THD comparison between OTA and OTA2

Circuit	Ibias ( $\mu\text{A}$ )	Input voltage ( $V_{pp}$ )	THD (%)
OTA	200	1	0.995
OTA2	200	1	8.185

**Table 1.2** Performance trade-offs of OTA

Variables	CM gain	DC gain	Noise	Dominant pole
$C_{gs1}$	–	–	↑	–
$C_{gs2}$	–	–	↑	–
$C_{gs3}$	–	–	↑	–
$C_{gs4}$	–	–	↑	–
$C_{gs5}$	–	–	↑	–
$C_{gs6}$	–	–	↑	–
$g_{ms19}$	–	–	↑	–

**Table 1.3** Performance trade-offs of OTA3

Variables	CM gain	DC gain	Noise	Dominant pole
$g_{md7}$	↓	↓	↑	↑
$C_{gd7} + C_{db7}$	–	–	–	↓
$g_{mg9}$	–	–	↑	–
$g_{md9}$	↓	↓	↑	–
$g_{ms9}$	↑	↑	↓	–

its advantage with respect to high output resistance, Tables 1.2 and 1.3 compare the trade-offs of the low voltage current mirror (M7, M9, M11, and M13) (OTA) and basic current mirror (M11 and M13) (OTA3). The tables only include the variables that cause different effects on gain, noise, and bandwidth. Table 1.3 indicates a higher flexibility of gain and pole position because of the cascode devices M7 and M9. The additional term  $g_{ms9}/g_{md9}$  introduces an enhanced DC gain for the low voltage current mirror structure.

Design step N3 adds a cascode current source biasing to the circuit. Using cascode biasing instead of single current source biasing is justified by an improved power supply rejection ratio (PSRR). For unity gain configuration, PSRR simulation shows that the cascode current source biasing achieves 5.44-dB rejection. A circuit with single current source biasing results in  $-7.06$  dB rejection, which is more than 12 dB less.

In summary, Fig. 1.6 presents the causal reasoning of the design sequence for circuit 1. The design sequence starts from combining features such as cross-coupled quad cell-based input stage, three differential pair input stage, current mirror, and fully differential structure. The design sequence includes the justified design steps that add to the circuit the specific three pairs input stage implementation, low voltage current mirror, and cascode current source biasing.

### 1.4.2 Circuit 2

Circuit 2 in Fig. 1.7 is a multi-path operational transconductance amplifier implemented in a sixth-order 10.7-MHz band-pass switched-capacitor filter [13]. The complete set of features (set  $\Sigma$ ) includes all circuit features: three-path OTA (a folded-cascode OTA, a current mirror cascode OTA, a current mirror folded-cascode OTA), double differential pair cross-coupled input, CMFB circuit, and current source biasing.

The starting features (set  $S$ ) include combining the following features: a folded-cascode OTA (set  $\Gamma$ ), a current mirror cascode OTA (set  $\Gamma$ ), a complementary folded-cascode OTA (set  $A$ ), a multi-path OTA (set  $A$ ), and a fully differential structure. The starting features correspond to the structures labeled as 1 and 2 in Fig. 1.7. The multi-path OTA is an abstract idea originally discussed in a cited paper and which implements a two-path OTA.

Beginning with the starting features in set  $S$ , the uncovered features of the design are computed by the difference set  $\Sigma$ —set  $S$  and labeled as 3, 4, and 5. The uncovered features are utilized to mine the design sequence that added them as justified steps to the set of initial features. Similar to circuit 1, for each feature, justification first starts with the more abstract features as well as the more likely feature.

Figure 1.8 presents the causal reasoning information for circuit 2. The design sequence starts from combining the features folded-cascode OTA, current mirror cascode OTA, complementary folded-cascode OTA, multi-path OTA, and fully differential structure. The design steps add the specific three-path OTA implementation, CMFB circuit, and current source biasing to the initial ideas.

Design step N1 adds the three-path OTA including a folded-cascode OTA, a current mirror cascode OTA, and a current mirror folded-cascode OTA. The double differential pair cross-coupled input is the unique causal structure, which is required in consistent design sequence. The three-path OTA structure is justified by a previous design that used a two-path OTA. In order to illustrate its advantage over a two-path OTA, Tables 1.4 and 1.5 present the trade-offs of circuit 2 (OTA) and the

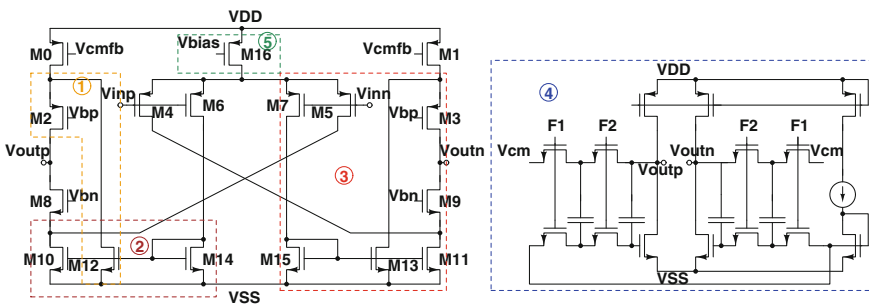


Fig. 1.7 Circuit 2

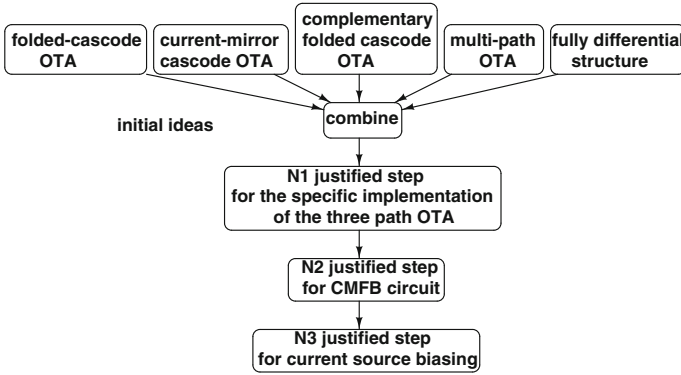


Fig. 1.8 Initial ideas and design sequences for circuit 2

Table 1.4 Performance trade-offs of OTA

Variables	CM gain	DC gain	Noise	Dominant pole
$g_{ms4}$	–	–	↓	–
$c_{gd4}$	–	–	↑	–
$c_{gs4}$	–	–	↓	–
$g_{md4}, g_{md10}$	↑	↓	↑	–
$g_{mg4}$	–	↑	–	–
$g_{mg6}$	–	↑	↑	–
$g_{ms6}$	–	–	↑	–
$g_{ms8}$	↓	↑	↓	–

Table 1.5 Performance trade-offs of OTA2

Variables	CM gain	DC gain	Noise	Dominant pole
$g_{md10}$	↓	↓	↑	–
$g_{mg6}$	↑	↑	↑	–
$g_{ms6}$	↓	–	↑	–
$g_{ms8}$	↑	↑	↓	–

two-path OTA without devices M4 and M5 (OTA2). The common structures in the two OTAs result in the same trade-offs on performance, which, for brevity, were not included in the tables. Regarding DC gain, the additional variables  $g_{mg4}$  and  $g_{md4}$  enhance gain by  $g_{mg4}/g_{md4}$  in the three-path OTA. Meanwhile,  $c_{gd4}/c_{gs4}$  degrades the noise performance.

Design step N2 adds the common-mode feedback CMFB to the circuit. A fully differential amplifier usually requires CMFB circuit to stabilize the common-mode level of the outputs. To justify using a CMFB circuit, sensitivity analysis was performed on the common-mode configured circuits. Sensitivity analysis studies the mapping of all circuit parameter variations onto the performance specifications of



the circuit [34]. The results show that devices M12, M6, M16, and M14 increase the common-mode gain, which degrades the common-mode performance. For comparison, in a circuit without CMFB circuit, devices M13, M1, M7, M16, M15, M6, M10, and M11 reduce the common-mode performance for an equal sensitivity value. Therefore, CMFB circuit is justified by improved common level of the outputs.

Design step N3 adds current source biasing to the circuit. Current source biasing is required as the circuit would not operate without biasing.

### 1.4.3 Circuit 3

Circuit 3 in Fig. 1.9 is a linearized operational transconductance amplifier (OTA) for low-voltage and high-frequency applications [16]. The complete set of structural features (set  $\Sigma$ ) includes the following features: fully differential, double cross-coupled pseudodifferential pair input, low voltage current mirror, linear region transistors, common-mode feedforward (CMFF) and common-mode feedback (CMFB) circuits, and current source biasing.

The starting features (set  $S$ ) include the following: combining common-mode control system (set  $I$ ), pseudodifferential input pair (set  $A$ ), fully differential structure (represents new insight of the designer), and nonlinearity cancelation (is new insight of the designer). The starting features correspond to the structures labeled as 1 and 2 in Fig. 1.9. Nonlinearity cancelation is among the starting ideas, representing the abstract idea use transconductance linearization.

Figure 1.10 shows the causal reasoning information for circuit 3. The design sequence starts from combining the features including the abstract idea of

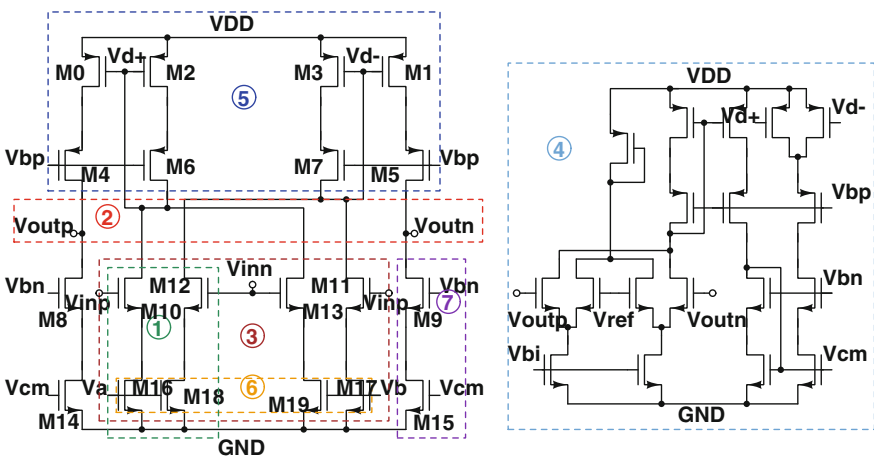
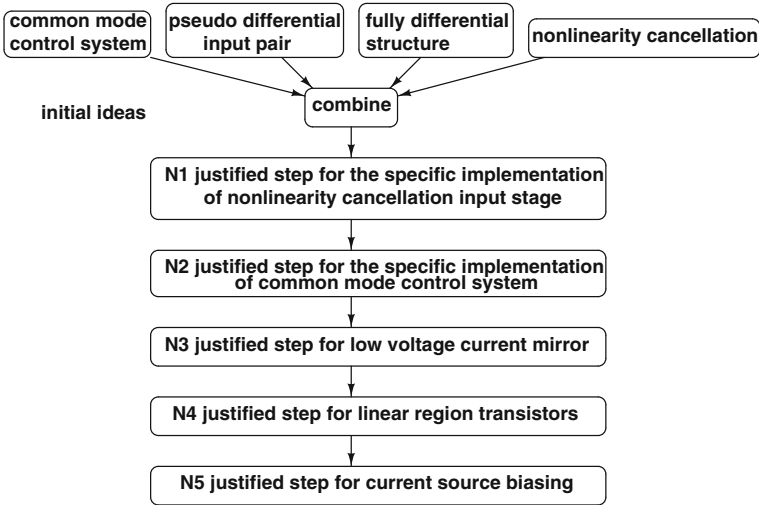


Fig. 1.9 Circuit 3



**Fig. 1.10** Initial ideas and design sequences for circuit 3

common-mode control system, pseudodifferential input pair, fully differential structure, and nonlinearity cancellation. The actual design steps include the justified steps for the specific input stage implementation for nonlinearity cancellation, the specific implementation for common-mode control system, low voltage current mirror, linear region transistors, and current source biasing circuit. The steps to create the design sequence are as follows.

Beginning with the starting features set  $S$ , the uncovered features of the circuit are computed as the difference set  $\Sigma$ —set  $S$ . The uncovered features are labeled as 3, 4, 5, 6, and 7 in Fig. 1.9. For each uncovered feature, justification starts first analyzing the more abstract feature as well as the more likely feature.

Design step N1 adds the double cross-coupled pseudodifferential pair with degenerated transistors' input to the set of initial features. Step N1 is justified by a previous design (cited by the paper) that uses pseudodifferential input pair. The cross-coupled structure implements the abstract idea of nonlinearity cancellation. To justify the resulting structure, the transient response of the structure was compared with a circuit with single differential pair input and with ideal current source (OTA2). Table 1.6 summarizes the total harmonic distortion for 10 MHz sine input. The circuit without the design feature has a total harmonic distortion of 7.093 %. The circuit proposed in Fig. 1.9 achieves better linearity performance. Tables 1.7 and 1.8 present the trade-offs of these two circuits for gain, noise, and bandwidth.

**Table 1.6** THD comparison between OTA and OTA2

Circuit	Input voltage ( $V_{pp}$ )	THD (%)
OTA	1	6.778
OTA2	1	7.093

**Table 1.7** Performance trade-offs for OTA

Variables	CM gain	DC gain	Noise	Dominant pole
$c_{gd13}$	–	–	↑	–
$g_{md13}$	↓	↓	↑	–
$g_{mg0}$	↑	↑	↓	–
$g_{ms0}$	–	–	↑	–
$g_{md4}, g_{md8}$	↓	↓	↑	↑
$g_{ms4}, g_{mg10}, g_{ms10}, g_{mg13}$	↑	–	↓	–
$g_{mg6}, g_{ms6}, g_{ms13}$	–	–	↓	–
$c_{gs10}, c_{gs13}$	–	–	↓	–
$g_{md16}, g_{md19}$	↑	–	↑	–

**Table 1.8** Performance trade-offs for OTA2

Variables	CM gain	DC gain	Noise	Dominant pole
$g_{mg0}, g_{mg10}$	↑	↑	–	–
$g_{ms0}, g_{ms4}, g_{mg6}, g_{ms6}$	–	–	↓	–
$g_{md4}, g_{md8}$	↓	↓	↑	↑

Circuit 3 has reduced gain performance because of the cross-coupled input stage. Linearity is achieved at the cost of gain reduction. The dominant pole position is the same for two circuits, but the cross-coupled stage introduces more noise.

Design step N2 adds the CMFF and CMFB circuits. This step is justified by the requirement of a proper common-mode control system in a pseudodifferential structure. The results for sensitivity analysis show that the circuit without common-mode control has the same sensitivity but with larger cost. Devices M19, M16, M2, M8, M14, M4, and M6 degrade the common-mode gain more in the circuit without common-mode control.

Design step N3 adds the low voltage current mirror to the circuit. Having a low voltage current mirror instead of a single current mirror is justified by its high output resistance and reduced output voltage (as only 0.4-V headroom is left for keeping devices M0 and M4 in saturation).

Design step N4 adds the linear region transistors to the circuit. The degenerated resistors in the pseudodifferential stage are implemented by transistors. The linear region transistors are justified by having the transconductance tuning ability that compensates for the variation caused by the fabrication process and temperature.

Design step N5 adds the current source biasing to the circuit. The current source biasing is a required step in the design sequence as otherwise the circuit would not operate.

### 1.5 Circuit 4

Circuit 4 in Fig. 1.11 is a recycling amplifier based on folded-cascode transconductance amplifier [14]. The complete set of features (set  $\Sigma$ ) includes the following structural features: double differential pair cross-coupled input, current mirror transistors as driving transistors, and transistors cascoded to single current mirror.

The starting features (set  $S$ ) include combining conventional folded-cascode amplifier, current mirror transistors work as driving transistors (set  $I$ ), multi-path OTA (set  $J$ ), and single-ended output structure (represents insight by the designer). The starting features correspond to the structures labeled as 1 in Fig. 1.11.

Figure 1.12 presents the causal reasoning information for circuit 4. The design sequence starts from combining conventional folded-cascode amplifier, current mirror transistors work as driving transistors, and multi-path OTA. The design steps include the justified step for adding to the circuit the double differential pair cross-coupled input, specific implementation of the current mirror, and the transistors cascoded to a single current mirror. The features that are added by the design sequence are computed by the difference set  $\Sigma$ —set  $S$  and are labeled as 2, 3, and 4 in Fig. 1.11.

Design step N1 adds the double differential pair cross-coupled input to the initial features. Step N1 is justified by the signal polarity since the output current is the sum of positive input path and negative input path.

Design step N2 adds the current mirror transistors as driving transistors. The recycling current mirror devices are justified by their additional current driving capability. In order to illustrate this advantage, Tables 1.9 and 1.10 present the trade-offs of the recycling folded-cascode and traditional folded-cascode circuit (OTA2). For brevity, only the variables having different effects on gain, noise, and bandwidth performance are shown. Table 1.10 indicates a higher flexibility of the

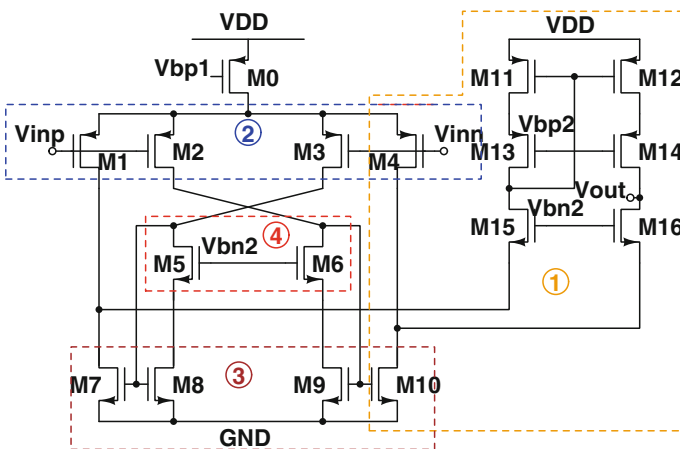


Fig. 1.11 Circuit 4

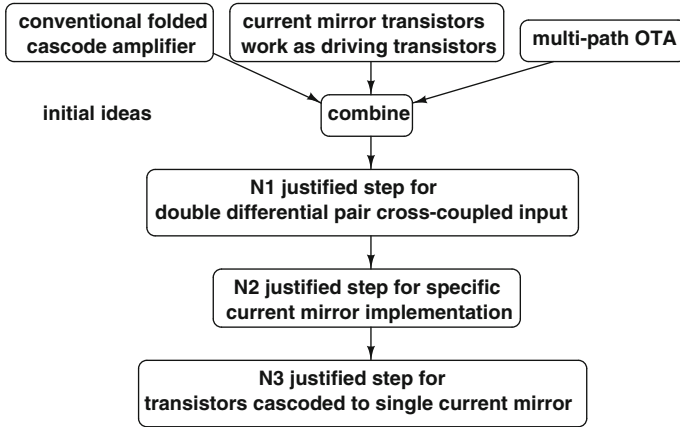


Fig. 1.12 Initial ideas and design sequences for circuit 4

Table 1.9 Performance trade-offs for OTA

Variables	CM gain	DC gain	Noise	Dominant pole
$g_{ms13}$	–	–	↓	–
$c_{gd1} + c_{db1}, c_{gs2} + c_{sb2}, c_{gd10} + c_{db10}, c_{gs16} + c_{sb16}$	–	–	–	↓
$g_{md1}$	↓	↓	↑	↑
$c_{gd1}, c_{gs1}, c_{gs2}, c_{gd10}$	–	–	↓	–
$g_{md2}, g_{md5}, g_{md12}, g_{md13}, g_{md15}$	↓	↓	↑	–
$g_{mg1}, g_{mg2}$	–	↑	↓	–
$g_{ms1}$	–	–	–	↑
$c_{gd2}, c_{gd8}, c_{gd11}, c_{gd12}$	–	–	↑	–
$g_{ms2}$	–	–	↓	↑
$g_{mg10}$	↑	↑	–	–
$g_{md10}$	↓	↓	↑	↑
$g_{mg12}$	–	↑	–	–
$g_{ms12}, g_{mg14}, g_{mg16}$	–	–	↑	–
$g_{md14}, g_{md16}$	↓	↓	↑	–
$g_{ms16}$	↓	↑	↓	↑

gain and bandwidth performance. DC gain is enhanced by devices M2 and M5, and the dominant pole is pushed further away by adjusting the input parameters  $g_{md1}$  and  $g_{ms2}$ .

Design step N3 adds the transistors cascoded to single current mirrors. Using additional cascoded transistors is due to the reduced DC mismatch since current mirrors have specific sizing. DC mismatch simulation is done on circuit 4 and the circuit without devices M5 and M6 (OTA3). Table 1.11 summarizes the simulation

**Table 1.10** Performance trade-offs for OTA2

Variables	CM gain	DC gain	Noise	Dominant pole
$g_{ms12}, g_{mg14}, g_{mg16}$	–	–	↓	–
$c_{gd11} + c_{db11}, c_{gs13} + c_{sb13}$	–	–	–	↓
$c_{gd1}, c_{gs1}, c_{gd11}, c_{gd12}$	–	–	↑	–
$g_{md1}, g_{md10}, g_{md12}, g_{md13}, g_{md14}, g_{md15}, g_{md16}$	↓	↓	↑	–
$g_{mg1}$	↑	↑	↓	–
$g_{ms1}$	↓	–	↓	↑
$g_{md11}$	–	–	–	↑
$g_{mg12}$	–	↑	–	–
$g_{ms13}$	–	–	↓	↑
$g_{ms16}$	↓	↑	↓	–

**Table 1.11** DC mismatch comparison between OTA and OTA2

Circuit	DC mismatch (V)
OTA	0.651
OTA2	0.824

results. Both circuits are configured by the same biasing current and input voltage. The cascoded transistors minimize the DC mismatch by 21 %, which justifies this design step.

## 1.6 Conclusions

This chapter presents an overview of our recent work toward devising a novel analog circuit synthesis method based on knowledge-intensive reasoning, similar to (at a basic level) to the tasks that humans conduct when solving new problems. The synthesis method mimics domain-specific knowledge-based reasoning. The reasoning procedure for synthesis is based on operators such as abstraction–instantiation, induction, and concept combination. The knowledge structure used for synthesis has two components: an associative part and a causal reasoning component. The associative part clusters the known circuit schematics into abstractions. Clustering uses the similarities and differences of the structural features of the circuits. Each abstraction is a summary of the common symbolic expressions that describe the behavior of the related instances. Also, the synthesis flow reuses the design sequences that have been previously used to solve similar design applications. Such sequences together with the starting design ideas are stored by the causal reasoning component of the knowledge structure. Experiments offer insight about the automated mining of the starting ideas and design sequences for four high-performance analog circuits.

Our ongoing research studies the effectiveness of the proposed domain knowledge structure and reasoning-based flow to address new applications and new design problems, e.g., by selecting or refining a circuit topology, identifying new design opportunities (by analyzing the combining of design features that have never been used together), and validating the design correctness by showing that all steps in a design sequence are justified. We also think that there is the opportunity of using the associative component of the knowledge structure to create in real-time comprehensive summaries of the state of the art on a certain design problem, such as by mining design information from electronic documents in databases, such as IEEE Xplore. Finally, this approach could create an initial path toward computationally mimicking some facets of human innovation and creativity, a well-known challenge in computing.

**Acknowledgement** This material is based upon work supported by the National Science Foundation under Major Collaborative Creative IT Grant No. 0856038 and Grant BCS No. 1247971. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## References

1. Chi, M., Feltovich, P., Glaser, R.: Categorization and representation of physics problems by experts and novices. *Cogn. Sci.* **3**, 121–152 (1981)
2. Smith, E., Osherson, D.: *An Invitation to Cognitive Science. Thinking.* MIT Press, Cambridge, MA (1995)
3. Thagard, P.: *The Cognitive Science of Science. Explanation, Discovery, and Conceptual Change.* MIT Press, Cambridge, MA (2012)
4. Vosniadou, S., Osherson, D.: *Similarity and Analogical Reasoning.* Cambridge University Press, Cambridge (1989)
5. Carley, R., Gielen, G., Rutenbar, R., Sansen, W.: Synthesis tools for mixed-signal ICs: progress on frontend and backend strategies. In: *Proceedings of the Design Automation Conference*, pp. 298–17303 (1996)
6. Doboli, A.: *Specification and design-space exploration for high-level synthesis of analog and mixed-signal systems.* Ph.D. Thesis, University of Cincinnati (2000)
7. Doboli, A., Dhanwada, N., Nunez, A., Vemuri, R.: A library-based approach to synthesis of analog systems from VHDL-AMS specifications. *ACM Trans. Des. Autom.* **9**(2), 238–271 (2004)
8. Gielen, G., Rutenbar, R.: Computer aided design of analog and mixed-signal integrated circuits. *Proc. IEEE* **88**, 1825–1852 (2000)
9. Tang, H., Zhang, H., Doboli, A.: Refinement based synthesis of continuous-time analog filters through successive domain pruning, plateau search and adaptive sampling. *IEEE Trans. CADICS* **25**(8), 1421–1440 (2006)
10. Ferent, C., Doboli, A.: Improving design feature reuse in analog circuit design through topological-symbolic comparison and entropy-based classification. In: Fakhfakh, M., Tlelo-Cuautle, E., Castro-Lopez, R. (eds.) *Analog/RF and Mixed-Signal Circuit Systematic Design.* Springer, Berlin (2013)
11. Hum, R.: Where are the dragons?. In *Frontiers in Analog Circuit Synthesis and Verification Workshop.* talk (2011)

12. Hawkins, J., Blakeslee, S.: *On Intelligence*. Times Books (2004)
13. Adut, J., Silva-Martinez, J., Rocha-Perez, M.: A 10.7-MHz sixth-order SC ladder filter in 0.35- $\mu\text{m}$  CMOS technology. *IEEE Trans. Circuits Syst. I Regul. Pap.* **53**(8), 1625–1635 (2006)
14. Assaad, R., Silva-Martinez, J.: The recycling folded cascode: a general enhancement of the folded cascode amplifier. *IEEE J. Solid-State Circuits* **44**(9), 2535–2542 (2009)
15. Koziel, S., Szczepanski, S.: Design of highly linear tunable CMOS OTA for continuous-time filters. *IEEE Trans. Circ. Syst. II Analog Digital Sig. Proc.* **49**(2), 110–122 (2002)
16. Lo, T.-Y., Hung, C.-C.: A 40-MHz double differential-pair CMOS OTA with IM3. *IEEE Trans. Circ. Syst. I Regul. Pap.* **55**(1), 258–265 (2008)
17. Ferent, C., Doboli, A.: An axiomatic model for concept structure description and its application to circuit design. *Knowl. Based Syst.* **45**, 114–133 (2013)
18. Ferent, C., Doboli, A.: Symbolic matching and constraint generation for systematic comparison of analog circuits. *IEEE Trans. CADICS* **32**(4), 616–629 (2013)
19. Ferent, C., Doboli, A.: Analog circuit design space description based on ordered clustering of feature uniqueness and similarity. *Integr. VLSI J.* **47**(2), 213–231 (2014)
20. Ferent, C., Doboli, A.: Novel circuit topology synthesis method using circuit feature mining and symbolic comparison. In: *Proceedings of Design, Automation and Test in Europe Conference (DATE)* (2014)
21. Jiao, F., Montano, S., Doboli, A., Doboli, S.: Analog circuit design knowledge mining: discovering topological similarities and uncovering design reasoning strategies. *IEEE Transactions on CADICS*. In Press. (2014)
22. Ferent, C., Doboli, A.: Formal representation of the design feature variety in analog circuits. In: *Proceedings of FDL Conference* (2013)
23. Ferent, C.: *Systematic modeling and characterization of analog circuits using symbolic and data mining techniques*. Stony Brook University. Ph.D. Thesis (2013)
24. Wei, Y., Doboli, A.: Structural macromodeling of analog circuits through model decoupling and transformation. *IEEE Trans. CADICS* **27**(4), 712–725 (2008)
25. Doboli, A., Vemuri, R.: Exploration-based high-level synthesis of linear analog systems operating at low/medium frequencies. *IEEE Trans. CADICS* **22**(11), 1556–1568 (2003)
26. Tang, H., Doboli, A.: High-level synthesis of delta-sigma modulators optimized for complexity, sensitivity and power consumption. *IEEE Trans. CADICS* **25**(3), 597–607 (2006)
27. Wei, Y., Tang, H., Doboli, A.: Systematic methodology for designing reconfigurable  $\Delta\Sigma$  modulator topologies for multimode communication systems. *IEEE Trans. CADICS* **26**(3), 480–496 (2007)
28. Koza, J., Bennett III, F., Andre, D., Keane, M.: Automated WYWIWYG design of both the topology and component values of analog electrical circuits using genetic programming. In: *Proceedings of the First Annual Conference on Genetic Programming*, pp. 28–1731 (1996)
29. McConaghy, T., Palmers, P., Gao, P., Steyaert, M., Gielen, G.: *Variation-Aware Analog Structural Synthesis: A Computational Intelligence Approach*. Springer, Berlin (2009)
30. Sripramong, T., Toumazou, C.: The invention of CMOS amplifiers using genetic programming and current-flow analysis. *IEEE Trans. CADICS* **21**, 1237–1252 (2002)
31. Ferent, C., Doboli, A.: Measuring the uniqueness and variety of analog circuit design features. *Integr. VLSI J.* **44**(1), 39–50 (2011)
32. Ferent, C., Doboli, A.: A prototype framework for conceptual design of novel analog circuits. In: *Proceedings of the International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design* (2012)
33. Wei, Y., Doboli, A.: Systematic development of analog circuit structural macromodels through behavioral model decoupling. In: *Proceedings of the 42nd annual Design Automation Conference*, pp. 57–62 (2005)
34. Das, T., Mukund, P.: Sensitivity analysis for fault-analysis and tolerance in RF front-end circuitry. In: *Proceedings of the Conference on Design, Automation and test in Europe (DATE)*, pp. 1277–1282 (2007)



# Chapter 2

## Automatic Synthesis of Analog Integrated Circuits Including Efficient Yield Optimization

Lucas C. Severo, Fabio N. Kepler and Alessandro G. Girardi

**Abstract** In this chapter, the authors show the main aspects and implications of automatic sizing, including yield. Different strategies for accelerating performance estimation and design space search are addressed. The analog sizing problem is converted into a nonlinear optimization problem, and the design space is explored using metaheuristics based on genetic algorithms. Circuit performance is estimated by electrical simulations, and the generated optimal solution includes yield prediction as a design constraint. The method was applied for the automatic design of a 12-free-variables two-stage amplifier. The resulting sized circuit presented 100 % yield within a 99 % confidence interval, while achieving all the performance specifications in a reasonable processing time. The authors implemented an efficient yield-oriented sizing tool which generates robust solutions, thus increasing the number of first-time-right analog integrated circuit designs.

### 2.1 Introduction

Analog integrated circuit (IC) design presents different characteristics from its digital counterparts in terms of number of devices, design methodologies, and design automation.

As digital electronic systems are modeled using hardware description languages (HDLs), digital design processes are largely removed from technology considerations and from actual physics of the devices. Digital IC design typically focuses on logical correctness, maximization of circuit density, placement, and routing of

---

L.C. Severo · F.N. Kepler · A.G. Girardi (✉)  
Alegrete Technology Campus, Federal University of Pampa, Alegrete-RS, Brazil  
e-mail: alessandro.girardi@unipampa.edu.br

L.C. Severo  
e-mail: lucas.severo@unipampa.edu.br

F.N. Kepler  
e-mail: fabio.kepler@unipampa.edu.br

circuits. The highly automated process produces variable, fab-independent netlists and easily generated layouts that are usually “right the first time.”

Analog/mixed-signal designs, on the contrary, are notorious for requiring more than just one prototyping cycle. They typically include a wide variety of primitive devices, such as digital MOS and mid- and high-voltage MOS and bipolar junction transistors (BJT), as well as a host of passive elements that include capacitors, resistors, inductors, varactors, and diodes. These devices are often required to operate under unfriendly environments, where they have to cope with high-temperature differences, high voltages, switching noise, and interference from neighboring elements. Analog circuits are also much more sensitive to noise than digital circuits, which can severely affect their performance.

Unlike digital designs, in which only a few device parameters, such as threshold voltage, leakage, and saturation currents, need to be considered, analog/mixed-signal design must cope with much more complex specifications. It is more concerned with the physics of devices. Parameters such as voltage gain, matching, power dissipation, and output resistance, for example, depend on voltage levels, device dimensions, and process parameters. Each device in the analog world must, therefore, be carefully characterized and modeled across a very large parameter space to allow for a reliable circuit design. This process usually leads to fab-specific designs that typically require more than one iteration to get the mask set right [14].

The success for first-time-right analog design in a traditional design flow can be summarized in three parts. The first is the experience of the design team. It can be acquired with a library of tried and tested design topologies and well-characterized devices, allowing a correct estimation of process variations. The second is the availability of good design kits and device models, providing an accurate characterization of transistor behavior in different operation points. Furthermore, a wide range of statistical models must be made available, including worst-case models, statistical corner models, and Monte Carlo mismatch models, making it possible for circuit design sizing and design centering techniques to achieve high-yielding and robust designs. The third is a good planning for optimizing the time necessary for a full design cycle, from initial specification to a functional prototype. A tight time-to-market, in general, plays a crucial role in the definition of the design schedule. In this context, a mandatory strategy for first-time-right silicon in analog design is the automation of critical design stages such as transistor sizing.

Automatic synthesis of analog integrated circuits is a very hard task due to the complex relationship between technology process parameters, device dimensions, and design specifications. The design of analog building blocks requires circuit parameters to be sized such that design specifications are met or even optimized. An efficient search in the design space is mandatory when hard specifications must be accomplished, mainly for low-voltage and low-power design. The exploration of all transistor operation regions is also fundamental for the search for an optimized circuit [18].

As devices shrink with the fabrication technology evolution, the impact of process variations on analog design becomes significant and can lead to circuit performance degradation and yield falling below specification [1, 20, 35]. Gate

oxide thickness, for example, approaches a few angstroms in state-of-the art technologies. Although  $V_{DD}$  scales to sub 1 V supply voltage headroom, threshold voltage does not scale in the same proportion due to leakage. Less headroom means more sensitivity to threshold voltage variation. This issue has led to the inclusion of yield prediction as a fundamental step in the analog design process. However, this prediction—estimated by Monte Carlo analysis—might demand high computational effort if included at each iteration of an optimization procedure [36].

With Monte Carlo simulation, one can find out how the distribution in circuit response relates to the specification. The aspects of yield considered here are the percentage of devices, which meet the specification and the design centering with respect to the specification.

Another important aspect is avoiding over-design, when the circuit characteristics are within specification but with a wide margin, which could be at the expense of area or power and ultimately, cost. Although not recommended, this strategy is still in use by most of the analog design teams because of the low level of design automation.

A typical design process is iterative, first for finding a solution which meets the nominal specification, and then moving on to a solution that meets yield and economic constraints, including the effects of variations in device characteristics. It helps to understand the relationship of the design parameters to the circuit response and the relationships of the different types of circuit response. However, it is a slow process, since it depends on the direct influence of the human designer.

The inclusion of yield prediction in the automatic circuit sizing procedure allows for a realistic modeling which contributes for a first-time-right design. The problem is that it often presents high complexity due to the long simulation time in the optimization process. Several hours is often required to optimize a typical-sized circuit.

In this work, we demonstrate the main aspects and implications of automatic sizing including yield. Different strategies for accelerating performance estimation are addressed.

In Sect. 2.2, we show the different strategies for automatic analog circuit sizing, while in Sect. 2.3 we show how it can be approached as an optimization problem. In Sect. 2.4, we present a tool for circuit sizing using optimization and considering yield. Then, in Sect. 2.5, we discuss the results of the automatic design of a Miller OTA using the tool. Finally, we draw conclusions in Sect. 2.6.

## 2.2 Strategies for Automatic Analog Integrated Circuit Sizing

In the analog design flow, the definition of transistor sizes, device values, and bias voltages and currents is called the sizing procedure. It can be implemented, in general, by two approaches: knowledge-based sizing or optimization-based sizing.

In the knowledge-based approach, the circuit sizing is performed based on the experience of the design team. This method uses analytic design equations that relate circuit performance to device characteristics. Although it is a good approach for older technologies, it is not suitable for designs in modern fabrication technologies, since the modeling of short-channel effects turns the design equations extremely complex and simplification leads to values far from the actual circuit response. Also, it is difficult to explore transistor operation regions other than strong inversion. An example of knowledge-based sizing tool is Procedural Analog Design (PAD) tool [33].

The optimization-based approach transforms the design procedure in a general optimization problem. The circuit performance is modeled in a cost function, and the design space is explored automatically by an optimization heuristic in the search for optimized solutions. According to Barros et al. [5], the optimization method is dependent on the design optimization model, which can be classified in equation based, simulation based, or learning based.

The equation-based method uses simplified equations originated from large- and small-signal analysis of the circuit topology. It allows for a fast estimation of circuit performance, but lacks in accuracy. The application of this method has been demonstrated in the literature, mainly with the use of geometric programming [15, 23]. The circuit performance is modeled by posynomial equations, which guarantee the finding of an optimal solution in a fast computational time. However, this modeling implies simplifications that compromise accuracy, since performance equations are not posynomials.

Simulation-based methods use electrical simulators such as SPICE to estimate circuit performance. This performance estimation method is purely numerical and tends to consume a large computational time, since several iterations are necessary to resolve the convergence algorithm implemented by SPICE. However, this method gives a very accurate performance estimation. Electrical simulation allows the calculation of all design specifications, in both time and frequency domains. Another advantage is that circuit variability and sensitivity can be estimated by corner models or Monte Carlo simulation.

The tool proposed by Phelps et al. [29] uses simulated annealing heuristic to explore a multi-objective cost function using Cadence Spectre simulator for performance estimation. The exploration of the design space using multi-objective genetic optimization is presented by De Smedt and Gielen [8], in which the calculation of the hypersurface of Pareto-optimal design points explores the trade-off between competing objectives.

Learning-based methods provide fast performance evaluation and good accuracy. It is obtained by using techniques such as support vector machines [5] and neural fuzzy networks [3]. The models are trained from electrical simulations. The drawbacks are the high effort necessary to train the models with the desired accuracy—a huge amount of simulation data is necessary—and the low configurability, since a simple modification in the circuit topology makes the trained model no more suitable for the application.

**Fig. 2.1** General optimization-based design flow

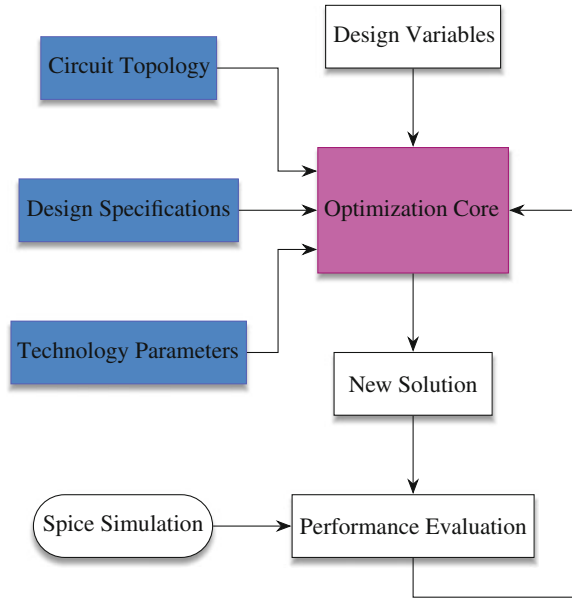


Figure 2.1 shows the execution scheme of an analog optimization-based tool with simulation-based performance evaluation. The tool takes as input the circuit topology, design specifications, and technology parameters. The optimization core generates solutions for the optimization problem according to the implemented optimization technique. For each iteration, it is necessary to evaluate the quality of the generated solution. It is quantified by a cost function, which gives an indication of the performance of the generated solution with respect to the desired specifications. The performance is estimated by SPICE simulation of a set of test benches in which design specifications can be extracted.

### 2.2.1 Sources of Process Parameter Variability

Submicrometer integrated circuit technologies present high incidence of variability in the fabrication process. These variations affect the performance of ICs, both analog and digital. In digital circuits, the effects are directly perceived in the propagation time of the digital signal. In analog circuits, process variations affect the operation point of the individual transistors and cause mismatch that can lead to loosing circuit functionality [10].

According to Orshansky et al. [27], the variations in the IC fabrication process can be classified in three categories: front end, back end, and variations caused by the environment. Front-end variations are caused by the first steps of device fabrication, such as ion implantation, oxidation, polysilicon deposition, and others.

In these stages, random variations occur in the transistor gate sizes ( $W$  and  $L$ ), threshold voltage ( $V_{TH}$ ), silicon oxide thickness ( $T_{ox}$ ), among others.

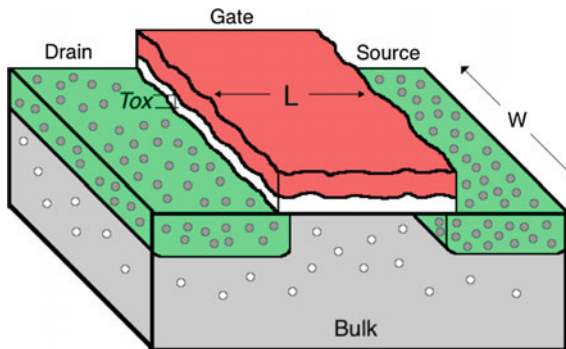
Back-end variations are characterized by the process variations caused in the metallization step of the fabrication process. The width of metal lines and interconnection vias, at different metal levels, are affected by random variations. At the same time, passive devices—such as capacitors and inductors—present variations around the nominal values, because they are implemented, in general, by metal lines.

The environment variations refer to the differences between the nominal and real operation conditions of the IC. We can cite temperature and supply voltage variations as example. These variations are systematic and can be treated at design level in order to attenuate their effects.

Figure 2.2 illustrates three of the main parameter variation sources present in an IC fabrication process. A random fluctuation of dopants occurs due to the difficulty in controlling the exact quantity and energy of the ion implantation in small devices. Some ions are located at undesired regions, and the concentration presents nonuniform patterns. At the same time, there is a random variation in the effective channel length and width, making it slightly different from the drawn dimensions. Devices with large gate length are less sensitive to process imperfections. Polysilicon width does not produce large variations in  $W$ , since this dimension is defined by the diffusion region, which, in general, has a large area [10]. Finally, the gate oxide thickness  $T_{ox}$  presents a random variation gradient along the wafer area.

According to Orshansky et al. [27], front-end variations are very relevant for an analog design. In order to exemplify the influence of a variation in the fabrication process over an integrated circuit, consider the threshold voltage  $V_{TH}$ . For large channel transistors with uniform doping [34], it can be estimated by

$$V_{TH} = V_{TH0} + \gamma(\sqrt{2|\phi_F| + V_{SB}} - \sqrt{2|\phi_F|}),$$



**Fig. 2.2** Representation of the main variation sources in the integrated circuit fabrication process: random doping fluctuation, effective gate dimensions, and gate oxide roughness

where  $V_{\text{TH0}}$  is the threshold voltage for a long-channel device with source-bulk voltage equal to zero,  $\phi_F$  is the Fermi level, and  $\gamma$  is obtained by

$$\gamma = \frac{\sqrt{2q\epsilon_{\text{Si}}N_{\text{sub}}}}{C_{\text{ox}}}.$$

Here,  $q$  is the fundamental charge of an electron,  $\epsilon_{\text{Si}}$  is the silicon permittivity,  $N_{\text{sub}}$  is the density of electrons in the substrate, and  $C_{\text{ox}}$  is the silicon oxide capacitance.

In this context, we can verify that a variation in the number of dopants in the substrate ( $N_{\text{sub}}$ ) has a great effect in the threshold voltage. The same occurs with a variation in  $C_{\text{ox}}$ , which is dependent on the gate oxide thickness  $T_{\text{ox}}$ :

$$C_{\text{ox}} = \frac{\epsilon_{\text{ox}}}{T_{\text{ox}}}.$$

Although well controlled, the gate oxide thickness can present relevant variations over different regions of the wafer.

The standard deviation of  $V_{\text{TH}}$  can be estimated by Orshansky et al. [27] as follows:

$$\sigma_{V_{\text{TH}}} = 3.19 \cdot 10^{-8} \frac{T_{\text{ox}} N_{\text{sub}}^{0.4}}{\sqrt{L_{\text{eff}} W_{\text{eff}}}}.$$

We can verify that, besides  $T_{\text{ox}}$  and  $N_{\text{sub}}$ , the variation of  $V_{\text{TH}}$  is related to the inverse square root of the effective gate area. Small devices are, therefore, more sensitive to process variations.

## 2.2.2 Estimating Circuit Yield

Yield is the ratio of accepted ICs over the total number of fabricated ICs. If the yield is significantly less than 100 %, this implies a financial loss to the IC manufacturer. Therefore, it is important to calculate and maximize the manufacturing yield already during the design stage [12]. This is called design for manufacturability, which implies techniques for yield estimation and yield optimization.

Failures that show up when the IC is in use in a product in the field are even more expensive, for instance when the IC is used under extreme operating conditions such as high temperatures. To try to avoid this, the design has to be made as robust as possible. This is called design for robustness or design for quality, which implies techniques for variability minimization and design centering.

Some works have been done in the analog automatic synthesis theme considering design for manufacturability. The strategy for calculating the yield-aware specification Pareto front is explored by Mueller-Gritschneider and Graeb [25].

Other approaches propose the use of simplified sampling techniques for yield estimation in order to reduce computational time [6, 13, 22]. The use of response surface method (RSM) for circuit performance modeled as quadratic functions of the process parameters is also reported [31]. All of them, however, face challenges in accuracy, compromising the search for the optimized circuit.

The problem is how to estimate yield with accuracy in a reasonable processing time. Monte Carlo is the standard technique for statistical simulation of circuits and for yield estimation during the design phase. The reason for this is that Monte Carlo is applicable to arbitrary circuits, arbitrary statistical models, and all performance metrics of interest, while allowing arbitrary accuracy. On the other hand, circuit size, nonlinearity, simulation time, and required accuracy often conspire to make Monte Carlo analysis expensive and slow. A single Monte Carlo run can cost a few thousand SPICE simulations, and higher accuracy requirements demand longer runs.

A detailed analysis of traditional pseudo-random Monte Carlo sampling, Latin hypercube sampling (LHS), and quasi-Monte Carlo (QMC) techniques is done by Singhee and Rutenbar [32]. The goal is to reduce the number of sample points while keeping the accuracy of the yield prediction. For high-dimensional problems, QMC presents advantages in terms of simulation speed, giving  $2\times$  to  $8\times$  speedup over conventional Monte Carlo for roughly 1 % accuracy levels.

## 2.3 Problem Formulation

The problem of analog integrated circuit sizing is modeled as an optimization problem by translating the circuit performance specifications to a cost function dependent on the transistor dimensions, capacitances, resistances, and bias voltages and currents (design free parameters). This cost function fully defines the performance space, which can be explored by an optimization heuristic for a minimum (or maximum) point. The optimized solution is contained in this point.

Consider a set of circuit performance functions (design specifications)  $\mathbf{X}(\mathbf{p}, \mathbf{q}) = \{S_1, S_2, \dots, S_k\}$ , which depends on a set of design parameter values  $\mathbf{p}$  and on a set of technology parameter values  $\mathbf{q}$ .  $S$  is an individual specification, and  $k$  is the number of design specifications. Performance functions for an operational amplifier can be the low-voltage gain ( $A_{v0}$ ), gain–bandwidth product (GBW), slew rate (SR), dissipated power ( $P_{\text{diss}}$ ), etc. Design parameters are the free variables the designer can handle in order to design the circuit, such as gate dimensions (length  $L$  and width  $W$ ), reference currents, and capacitor values. Technology parameters include electrical MOS model parameters (such as oxide thickness  $T_{\text{ox}}$  and threshold voltage of the long-channel device at zero substrate bias  $V_{\text{TH0}}$ ), supply voltages, and operation temperature range.



The acceptance of a circuit by a specification test can be expressed as follows:

$$\mathbf{X}(\mathbf{p}, \mathbf{q}) \in \Phi.$$

$\Phi$  is the region of acceptable performance specifications in the performance space. The acceptance region  $\Psi$  in the design parameter space is defined by

$$\mathbf{X}(\mathbf{p}, \mathbf{q}) \in \Phi \rightarrow \mathbf{p} \in \Psi.$$

A manufactured circuit will be considered acceptable if all of its actual performances fall within acceptable limits, i.e., if  $S_i^L \leq S_i \leq S_i^U$ , where the indexes  $L$  and  $U$  correspond to lower and upper specification limits, respectively.

The parameter values  $\mathbf{q}$  vary statistically around a nominal value, caused by unavoidable process fluctuations in manufacturing, with a joint probability density function (JPDF)  $g(\mathbf{p}, \mathbf{q})$ .

Two different types of parameter variation are present in a semiconductor fabrication process: global (interdie) and local (intradie) variation. Global variation of the electrical parameters is induced by process fluctuations in manufacturing, which affect all devices in a circuit in the same way. It is independent of length  $L$  and width  $W$ .

Local variation induces differences between identically designed devices caused by edge roughness, doping variation, boundary effects, etc. In this case, the variation of  $L$  depends on the width of the device. Other parameters such as sheet resistance, channel dopant concentration, mobility, and gate oxide thickness are inversely dependent on the gate area ( $W \cdot L$ ), since the parameters average over a greater distance or area [10]. Mismatch is dominated by local variation and affects the electrical behavior of input differential pairs and current mirrors, even for well-designed layouts.

The manufacturing yield  $Y$  of a circuit can be formulated by the number of accepted circuits that pass the specification test over the total number of considered circuits:

$$Y = \text{Prob}(\mathbf{X}(\mathbf{p}, \mathbf{q}) \in \Phi).$$

The manufacturing yield can be estimated by repeating circuit electrical simulation and performance specification evaluation. This is done by Monte Carlo analysis, which simulates the variation of the electrical parameters that affect all devices in a circuit. In order to simulate global variation, the process parameters are randomly selected in each simulation run and globally assigned to all device instances in a design. For local variation simulation, every instance of a device that contains matching-relevant parameters receives an individual random value around a typical mean.

There are some design techniques typically used to improve yield in analog circuits [7]. These techniques can be implemented—in combination or not—in three design stages: topology selection, transistor sizing, and physical synthesis

[24]. The focus of this chapter is restricted to yield maximization at the transistor sizing stage.

In our approach, yield maximization is performed by determining a set of nominal values of the design parameters,  $\mathbf{p}$ , that maximizes the probability of the random performances lying within  $\Phi$  [4]. However, there is not an explicit for estimating  $g(\mathbf{p}, \mathbf{q})$ . The yield optimization problem can be formulated in the space of independent statistical disturbances. The region of acceptability in the disturbance space contains all possible combinations of disturbances that can occur in the manufacturing of a circuit, which, for specific nominal parameter values, do not result in unacceptable performance. Yield optimization is, therefore, performed by modifying the acceptability region in a way that increases the coverage of a fixed probability distribution [9].

This yield can be calculated in both the device parameter space and the circuit performance space. This calculation, however, is complicated by the fact that, in either space, one of the two elements is not known explicitly: the statistical fluctuations are known in the device parameter space but not in the circuit performance space, whereas the acceptability region is known in the performance space but not in the parameter space [12]. Monte Carlo simulation, combined with an optimization procedure, is the most effective way to estimate acceptability region in the performance space.

### 2.3.1 Transistor Sizing as an Optimization Problem

The proposed approach for optimization of circuit performance explores the yield prediction as a design objective in an automatic sizing procedure. It is a nonlinear programming problem and requires the formulation of a single performance function (cost) to minimize subject to a set of inequality constraints, as in the following standard form [26]:

$$\begin{aligned} & \underset{\mathbf{p}, \mathbf{q}}{\text{minimize}} && F_m(\mathbf{p}, \mathbf{q}), \quad i = 1, \dots, M \\ & \text{subject to} && C_n(\mathbf{p}, \mathbf{q}) \leq C_{n(ref)}, \quad n = 1, \dots, N \end{aligned}$$

where  $M$  is the total number of  $F_m$  specifications to optimize, and  $N$  is the number of  $C_n$  constrained performance functions. Here,  $\mathbf{X}$  can be rewritten as a set of design objectives and design constraints:

$$\mathbf{X}(\mathbf{p}, \mathbf{q}) = \{F_1, \dots, F_M, C_1, \dots, C_N\}.$$

$C_n(\mathbf{p}, \mathbf{q})$  is a function that is dependent on the specification type: minimum required value ( $C_{\min}(\mathbf{p}, \mathbf{q})$ ) or maximum required value ( $C_{\max}(\mathbf{p}, \mathbf{q})$ ) [5].

These functions are shown in Fig. 2.3, where  $a$  is the maximum or minimum required value, and  $b$  is the bound value between acceptable and unacceptable performance values.

Acceptable but non-feasible performance values are the points between  $a$  and  $b$ . They return intermediate values for the constraint functions in order to allow the exploration of disconnected feasible design space regions. These functions return additional cost for the cost function if the performance is outside the desired range. Otherwise, the additional cost is zero.

The constrained problem can be transformed into an unconstrained minimization problem using the penalty function approach:

$$f_c(\mathbf{p}, \mathbf{q}) = \sum_{m=1}^M w_m \cdot \hat{F}_m(\mathbf{p}, \mathbf{q}) + \sum_{n=1}^N v_n \cdot \hat{C}_n(\mathbf{p}, \mathbf{q}). \quad (2.1)$$

Here,  $w_m$  and  $v_n$  are weights that indicate the relative importance of design objectives and design constraints, respectively.  $\hat{F}$  and  $\hat{C}$  are normalized design objective and design constraint functions, in order to keep all sum factors in the same order of magnitude.

Yield prediction can be easily included as a design objective in the cost function by adding a new term  $\hat{Y}(\mathbf{p}, \mathbf{q}, \epsilon_q)$  in the penalty function:

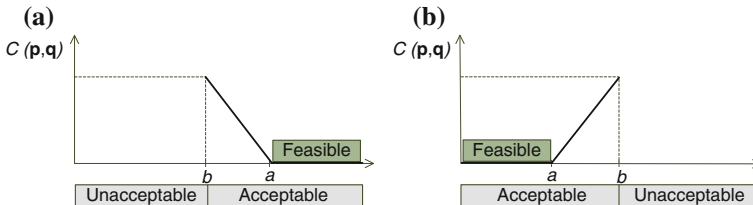
$$f_c^Y(\mathbf{p}, \mathbf{q}, \epsilon_q) = \sum_{m=1}^M w_m \cdot \hat{F}_m(\mathbf{p}, \mathbf{q}) + \sum_{n=1}^N v_n \cdot \hat{C}_n(\mathbf{p}, \mathbf{q}) + \hat{Y}(\mathbf{p}, \mathbf{q}, \epsilon_q). \quad (2.2)$$

This new term is dependent on the variability vector of technology parameters  $\mathbf{q}$  given by  $\epsilon_q$ .

We define the characteristic function of  $\Phi$  as

$$I_{\Phi}(\mathbf{X}) = \begin{cases} 1 & \text{if } \mathbf{X} \in \Phi \\ 0 & \text{if } \mathbf{X} \notin \Phi \end{cases}$$

which is 1 for pass and 0 for fail. This is also known as the indicator function.



**Fig. 2.3** Design constraint performance metrics: **a** minimum required value specifications and **b** maximum required value specifications

We can now define the circuit yield as the probability of a circuit instance lying in the acceptance region:

$$Y = \int_{\mathfrak{R}^*} I_{\Phi}(\mathbf{X})g(\mathbf{X})d\mathbf{X}.$$

There are other ways of calculating the yield, such as the use of process capability indexes ( $Cpk$ ). The key idea is normalizing the distance to the feasible boundary by the standard deviation ( $\sigma$ ) of its performance distribution to consider different process sensitivity. The application of these indexes, however, is out of the scope of this work.

In this work, the cost function is estimated by electrical simulations performed by Synopsys HSpice<sup>®</sup>. The high computational cost of Monte Carlo simulations is diminished by performing circuit statistical analysis only for a subset of design solutions in the optimization process, as detailed in the next section.

### 2.3.2 Monte Carlo in the Optimization Flow

According to Eq. 2.2, we need to estimate circuit yield for the calculation of the cost function in the optimization flow. However, it is computationally costly if done by Monte Carlo simulation. On the other hand, if yield prediction is not considered, the optimization algorithm tends to find optimal solutions close to the border of the performance space. At these points, a small variation in the process parameters makes the performance specifications fall outside the acceptable region. Thus, a strategy for dealing with this problem must be included in the optimization process. It is called design centering.

There are two approaches for improving processing time considering Monte Carlo simulations: to reduce the number of iterations in which Monte Carlo simulations are necessary; and to reduce the number of runs in a Monte Carlo simulation. Both strategies are essential for improving the processing time during the search for an optimal design solution.

#### 2.3.2.1 Reducing the Number of Monte Carlos Simulations

It is possible to reduce the number of iterations in which the calculation of  $Y$  is necessary by analyzing the influence of this term over the entire cost function of Eq. 2.2. Figure 2.4 illustrates this strategy. Consider first the cost function in Eq. 2.1, which does not include yield. If the current solution is not a best solution candidate, i.e., if it is already a worst solution even without the calculation of  $Y$ , this solution can be discarded and Monte Carlo simulation is not necessary. As  $f_c^Y(\mathbf{X})$  is unknown before the Monte Carlo simulation, the test for best solution candidate is

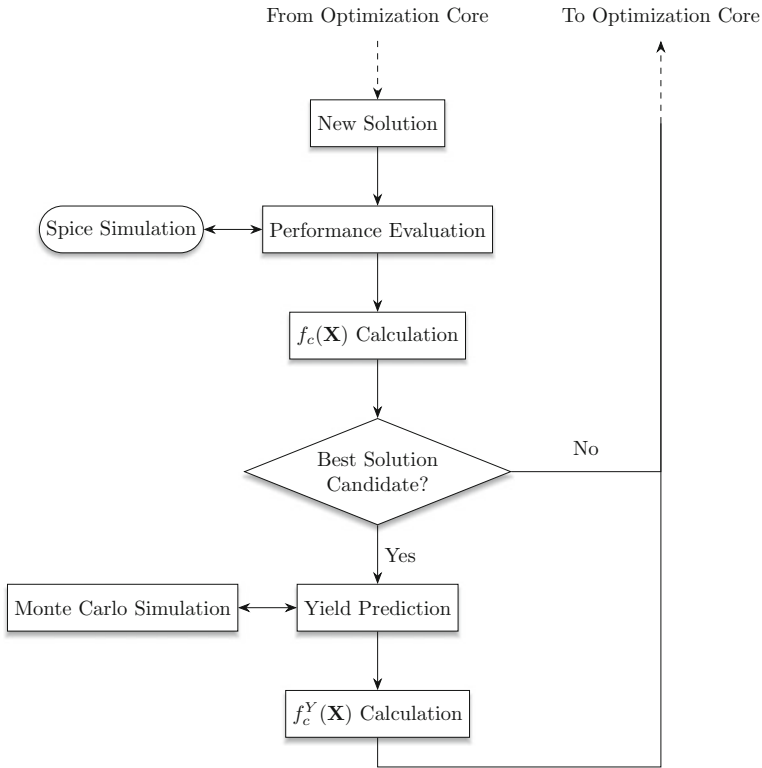


Fig. 2.4 Optimization flow including yield prediction only for best solution candidates

done by the calculation of  $f_c(\mathbf{X})$ . Monte Carlo simulation is executed only when  $f_c(\mathbf{X}) < \min(f_c^Y(\mathbf{X}))$ —where  $\min(f_c^Y(\mathbf{X}))$  is the current best solution.

At the start of the optimization process, the current best solution is frequently updated. However, the update frequency tends to reduce along iterations. Consequently, the number of iterations in which Monte Carlo simulation is necessary also reduces. Computational time is spared, since useless Monte Carlo simulations are avoided.

### 2.3.2.2 Reducing the Number of Runs in a Monte Carlo Simulation

The reduction in the number of runs in a Monte Carlo simulation can be implemented with the calculation of the number of samples ( $n$ ) necessary to achieve a desired confidence level in the yield estimation.

The expected value of a random variable  $p$  is  $\mu = E(p)$ . If we generate values  $p_1, \dots, p_n$  independently and randomly from the distribution  $p$ , we can estimate  $\mu$  as

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n p_i.$$

One of the great strengths of the Monte Carlo method is that the sample values themselves can be used to get a rough idea of the error  $\hat{\mu}_n - \mu$ . The average squared error in Monte Carlo sampling is  $\sigma^2/n$ . The most commonly used estimation of standard deviation  $\sigma^2$  is

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (p_i - \hat{\mu}_n)^2.$$

Monte Carlo sampling typically uses large values of  $n$  for guaranteeing that this estimation is a good approximation to the actual  $\sigma^2$ .

A variance estimate  $\sigma^2$  tells us that the error is on the order of  $\sigma/\sqrt{n}$ . We know that  $\hat{\mu}_n$  has mean  $\mu$  and we can estimate its variance by  $\sigma^2/n$ .

From the central limit theorem (CLT), we also know that  $\hat{\mu}_n - \mu$  has approximately a normal distribution with mean 0 and variance  $\sigma^2/n$ . The CLT can be used to get approximate confidence intervals for  $\mu$ . For 95 % confidence interval,

$$\mu_{95\%} = \hat{\mu}_n \pm 1.96 \frac{\sigma}{\sqrt{n}}.$$

For 99 % confidence interval,

$$\mu_{99\%} = \hat{\mu}_n \pm 2.58 \frac{\sigma}{\sqrt{n}}. \quad (2.3)$$

In a general way,

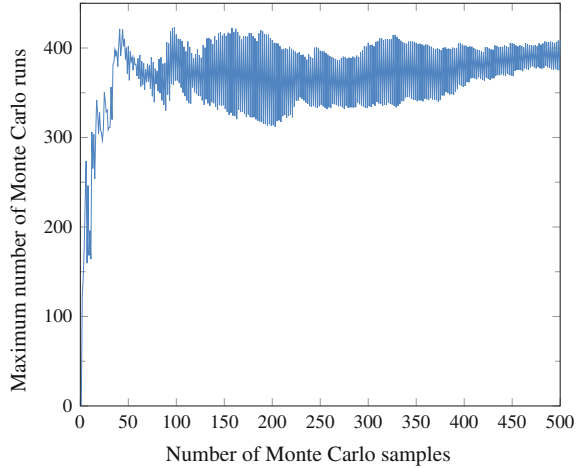
$$\mu_{c\%} = \hat{\mu}_n \pm \Phi^{-1}(1 - \alpha/2) \frac{\sigma}{\sqrt{n}},$$

where  $\alpha = 1 - c/100$  and  $\Phi(\cdot)^{-1}$  is the inverse cumulative distribution function (ICDF) of  $N(0, 1)$ , the standard normal distribution. It is not available in closed form, and computation requires careful use of numerical procedures. It is also called “probit function,” an acronym for “probability unit function.” The probit function can be calculated as

$$\text{probit}(p) = \sqrt{2} \cdot \text{erf}^{-1}(2p - 1).$$

In Matlab, the `erfinv` function is available for  $\text{erf}^{-1}$  (inverse error function).

**Fig. 2.5** Example of the maximum number of Monte Carlo runs calculated with Eq. 2.3. The circuit is a Miller amplifier, and the specification being measured is low-voltage gain. A good choice for  $n_0$  is 50, the value where the graph stabilizes around 400



The steps for determining the number of Monte Carlo runs that matches a 99 % confidence interval for a determined specification are the following:

```

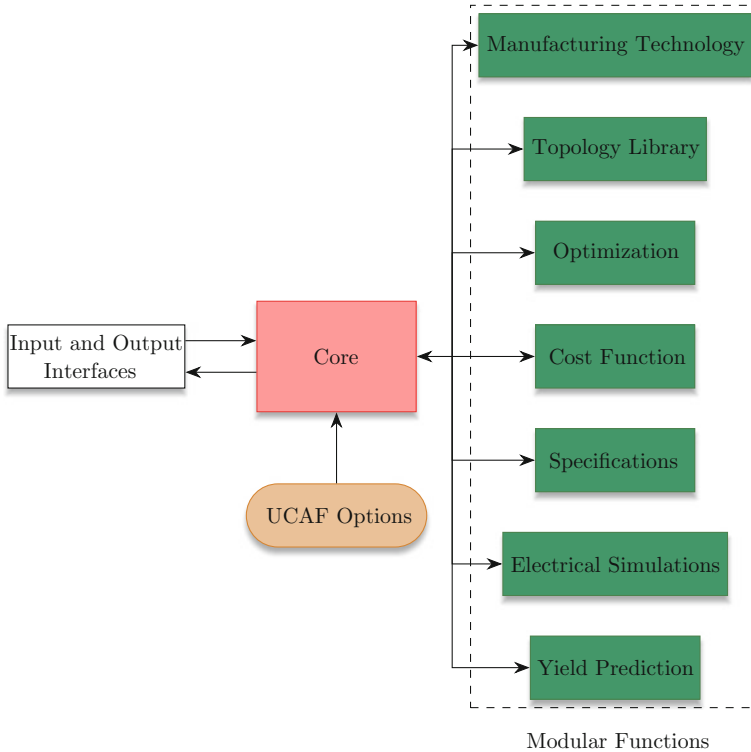
Initialize  $n_0$ ;
 $i = 0$ ;
repeat
     $i = i + 1$ ;
    Run Monte Carlo simulation;
    Calculate  $\hat{\sigma}_i$  and  $\Delta_\mu$ ;
     $n_i = (\frac{\hat{\sigma}_i}{\Delta_\mu} \cdot 2.58)^2$ ;
until  $n_i > n_{i-1}$ ;

```

The correct choice of  $n_0$  is fundamental for correctly estimating  $n_{i+1}$ , since the confidence of  $\sigma$  is dependent on  $n$ . With some simulations, we can infer that  $n_0 = 50$  is a good choice. Figure 2.5 shows the simulation of Eq. 2.3 for different values of  $n$  for estimating the low-voltage gain of a Miller OTA. One can note that the graph stabilizes for  $n = 50$ , indicating a maximum number of samples ( $n_{i+1}$ ) equal to 400.

## 2.4 The UCAF Tool

UCAF is a CAD tool we developed for the automatic design of analog basic blocks including yield optimization. The tool sizes an integrated circuit by modeling it as an optimization problem and exploring efficiently the design space searching for optimal solutions. The main design flow of this tool is the same shown in Fig. 2.1. This general design flow is implemented in Matlab, and Synopsys HSpice is used for performance estimation.



**Fig. 2.6** UCAF structure

The implementation of UCAF is made using modular functions to solve a generic analog IC design. These modular functions are shown in Fig. 2.6. Modularity allows a high degree of configurability, since each function can be substituted by a similar one without losing functionality. For example, the optimization algorithm can be changed independently of the remaining functions. The tool is configured with the aid of a specific script as the input interface. Also, a graphical interface guides the user through the basic configurations. The output interface presents the generated solutions.

The “Core” module is the main function, which creates and organizes a new design, creates design folders, sets the modular functions, writes the simulation file, and performs other important tasks.

The function “Manufacturing Technology” implements the interface between the design and the fabrication technology. It reads and configures the parameters of simulation models from the design kit provided by the foundry.

Each new analog block inserted in the tool is saved in a cell library. This task is performed by the “Topology Library” function. These cells can be reused for different design specifications.



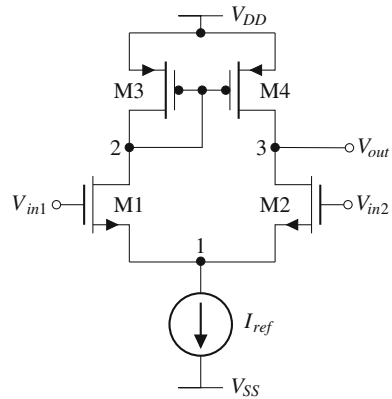
The “Optimization” function is responsible for the optimization algorithm that will guide the design space exploration. This function will be detailed further in Sect. 2.4.2.

The “Cost Function” implements Eq. 2.1 and is responsible for representing the design as an unconstrained minimization problem. In order to evaluate the cost function, it is necessary to estimate the values of the circuit specifications. Thus, the UCAF tool has the “Electrical Simulation” and “Specifications” functions. These functions are analyzed with more details in Sect. 2.4.3.

### 2.4.1 A Simplified Design Example

The design flow of the UCAF tool can be illustrated by the sizing of a simple active load differential amplifier circuit, as shown in Fig. 2.7. It is composed of four transistors and a tail current ( $I_{ref}$ ). In order to simplify the design, we assume all transistors have the same size and  $I_{ref}$  is fixed at 10  $\mu$ A. This is not of practical use, but reduces the design problem to only two variables: the transistors channel width ( $W$ ) and length ( $L$ ). It also allows the visualization of the design space and provides an intuitive understanding of the optimization procedure.

**Fig. 2.7** Active load differential amplifier



**Table 2.1** Required specifications for the differential amplifier of Fig. 2.7

Specification	Required value
Gate area	Minimize
Low-frequency gain ( $A_{vo}$ )	$\geq 40.00$ dB
Phase margin (PM)	$\geq 70.00$
Gain-bandwidth product (GBW)	$\geq 1.00$ MHz
Input common-mode range (ICMR <sup>+</sup> )	$\geq 0.40$ V

Table 2.1 summarizes the circuit performance specifications. The design objective is to minimize the gate area ( $W \cdot L$ ). The cost function is calculated by

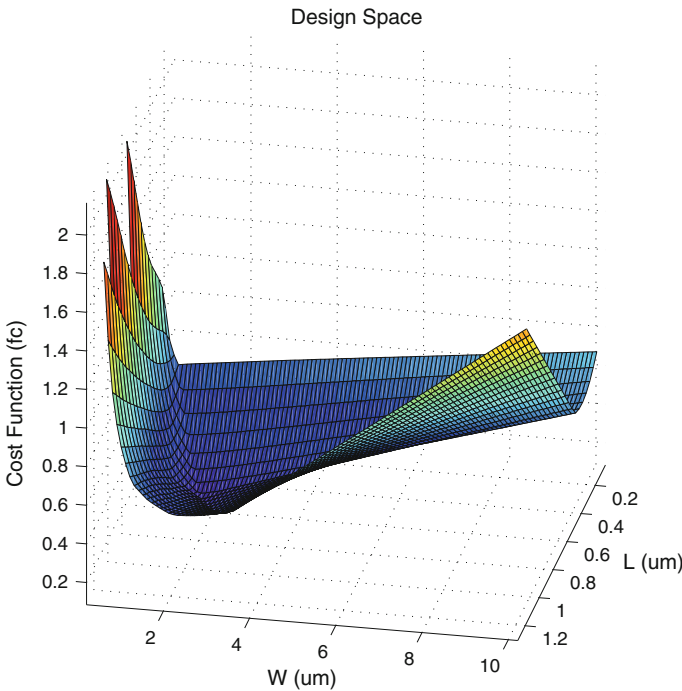
$$f_c(\mathbf{X}) = \frac{1}{\text{Area}_{ref}} \cdot \text{Area}(\mathbf{X}) + \sum_{n=1}^N C(\mathbf{X}), \quad (2.4)$$

where  $\mathbf{X}$  is the vector of free variables ( $\mathbf{X} = [WL]$ ),  $\text{Area}_{ref}$  is the weighting parameter of the gate area, and  $C_n(\mathbf{X})$  represents the constraint performance metric for the required specifications.  $\sum C(\mathbf{X})$  is calculated by

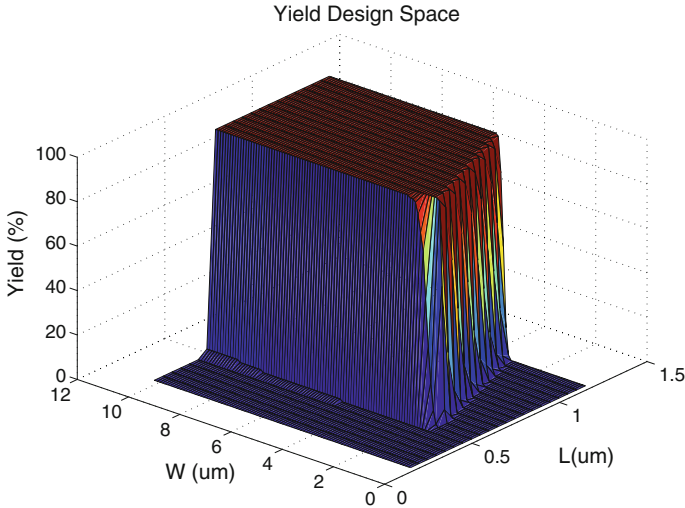
$$\begin{aligned} \sum C_n(\mathbf{X}) = & C_{\min}(A_v(\mathbf{X}), A_{vref}) + C_{\min}(\text{PM}(\mathbf{X}), \text{PM}_{ref}) \\ & + C_{\min}(\text{GBW}(\mathbf{X}), \text{GBW}_{ref}) + C_{\min}(\text{ICMR}^+(\mathbf{X}), \text{ICMR}_{ref}^+), \end{aligned}$$

where  $C_{\min}$  and  $C_{\max}$  are the constraint performance functions shown in Fig. 2.3.

Assuming  $W$  can vary between 0.22 and 10  $\mu\text{m}$  and  $L$  between 0.2 and 1  $\mu\text{m}$  in steps of 0.05  $\mu\text{m}$ , the optimization problem has 3120 possible solutions. We exhaustively calculated all 3120 solutions by electrical simulation. The resulting design space is shown in Fig. 2.8, where one can see the high nonlinearity of the



**Fig. 2.8** Resulting design space composed by 3120 possible solutions for the differential amplifier of Fig. 2.7



**Fig. 2.9** Yield design space estimated by Monte Carlo simulation

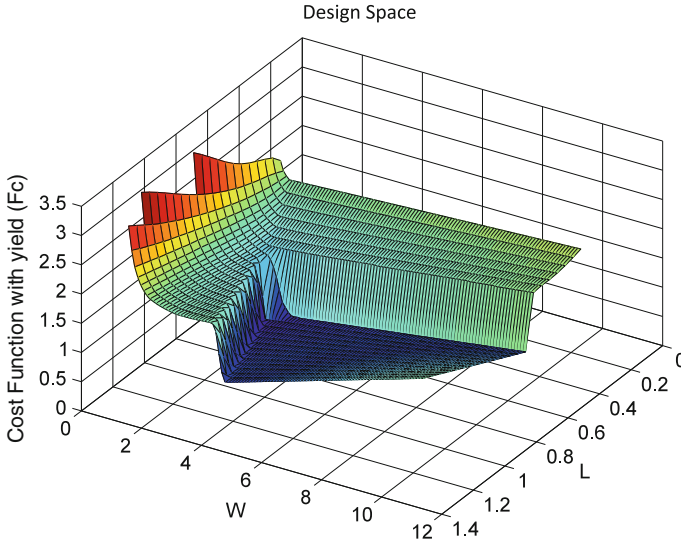
cost function with respect to the design free variables. The optimization method searches the design space in order to find the value of  $W$  and of  $L$  that make the cost function have the lowest value. The optimal solution is reached at the point  $W = 1.62 \mu\text{m}$  and  $L = 0.54 \mu\text{m}$ , with a cost value of 0.11958.

The yield of each solution is evaluated by Monte Carlo simulations with 200 samples. The yield design space is shown in Fig. 2.9. It is possible to see that the design space is abruptly deformed at the region where the yield moves from 100 to 0 %. The optimal solution not considering yield represents, in practice, an yield of 51.8 %, indicating that the solution is located in a performance region very sensitive to process variations.

Using Eq. 2.2, the design spaces of Figs. 2.8 and 2.9 can be joined to result in a design space including yield prediction, shown in Fig. 2.10. The optimal solution is now at  $W = 1.82 \mu\text{m}$  and  $L = 0.56 \mu\text{m}$ , with a cost value of 0.13658 and yield of 99.8 %. The optimal point moved just slightly, but enough for considerably increasing the yield. The difference between this optimal value and the optimal value without yield is  $\Delta W = 0.02 \mu\text{m}$  and  $\Delta L = 0.02 \mu\text{m}$ .

## 2.4.2 Optimization

This is the main function of the UCAF tool, because it is responsible for exploring the design space. Here we opted for using genetic algorithms (GA), available in the Matlab Genetic Algorithm Optimization Toolbox (GAOT) [17].



**Fig. 2.10** Design space considering yield prediction

The GA optimization approach is based on the biology theories of evolution and genetics. It is a non-deterministic meta-heuristic and can be used for optimizing nonlinear functions [11].

A specialized vocabulary is used for better reflecting the biological approach. A solution is called an “individual,” and since GA work with a number of solutions simultaneously, this set of solutions is called “population.” The iterations of the optimization process are called “generations,” and the cost function is referred as the “fitness” function.

In each generation, the individuals of the current population are crossed, generating new individuals that share characteristics from both parents (“crossover”) and that may suffer “mutation.” Each individual is represented as a chromosome, which in turn represents the optimization variables and their values.

Figure 2.11 shows the flowchart for optimizing the circuit size using GA. The GA core receives three inputs: the configuration parameters, the design specifications, and the technology parameters. The first step is creating an initial set of solutions, which is randomly performed by an initialization function. Each solution is then evaluated according to the fitness (cost) function, given by Eq. 2.4, but replicated here for convenience (recall that circuit specifications are estimated via electric simulations):

$$f_c(\mathbf{X}) = \frac{1}{\text{Area}_{\text{ref}}} \cdot \text{Area}(\mathbf{X}) + \sum_{n=1}^N C(\mathbf{X}).$$

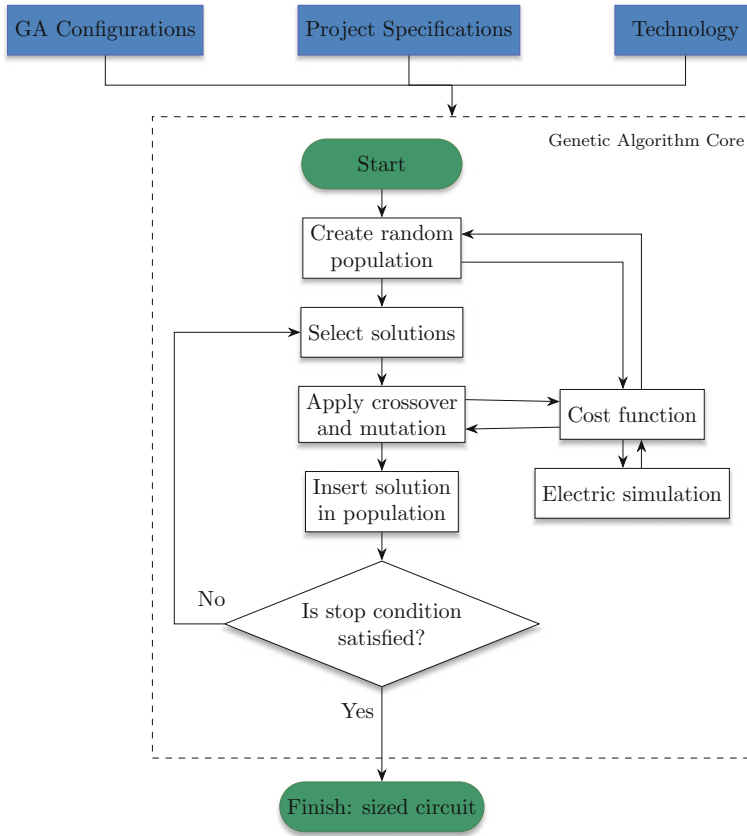


Fig. 2.11 Genetic algorithm flowchart for circuit sizing

Afterward, a subset of the solutions is chosen based on a selection function. The GAOT accepts the following selection functions: roulette wheel, normalized geometric rank selection, and tournament selection. The roulette wheel function first gives a normalized probability for each solution based on its fitness and then builds a roulette based on these probabilities. A random number is generated, and the solution with this number is selected. That way, the better the fitness, the higher the chance of being selected.

The selection by ranking orders the solutions based on their fitness and assigns a selection probability to each position. As with the roulette function, a solution is randomly chosen according to the probabilities. The difference here is that the probability of selection of a solution does not directly depend on its fitness, but only on its rank.

The third selection function chooses a number of solutions uniformly at random and keeps only the best solution. New tournaments are drawn, and the best overall solution is kept.

After selecting a subset of solutions, the next step in Fig. 2.11 is to perform crossover and mutation over the solutions. These operations are responsible for the state space search process of GA, since they generate new solutions. The crossover operation takes two solutions (chromosomes), splits them at some random point, and then combines one part from each chromosome, generating two new chromosomes. The mutate operation takes a single chromosome, selects a random point, and then inverts its value. Various functions for crossover and mutation are described in Houck et al. [16].

The new population is tested and, if the stop condition is satisfied, the circuit is sized and the process finishes. If it is not satisfied, a new generation (iteration) is performed. A maximum number of generations or a minimum cost function difference can be used as stop conditions.

### 2.4.3 Circuit Characterization

The optimization procedure has an interface with an external electrical simulator to estimate the circuit performance. For each specification, it is necessary to simulate a circuit test bench, performing AC, DC, or transient analysis. The current version of UCAF tool has some circuit standard test benches to measure the specifications of operational amplifiers [2, 30]. These test benches are shown in Fig. 2.12.

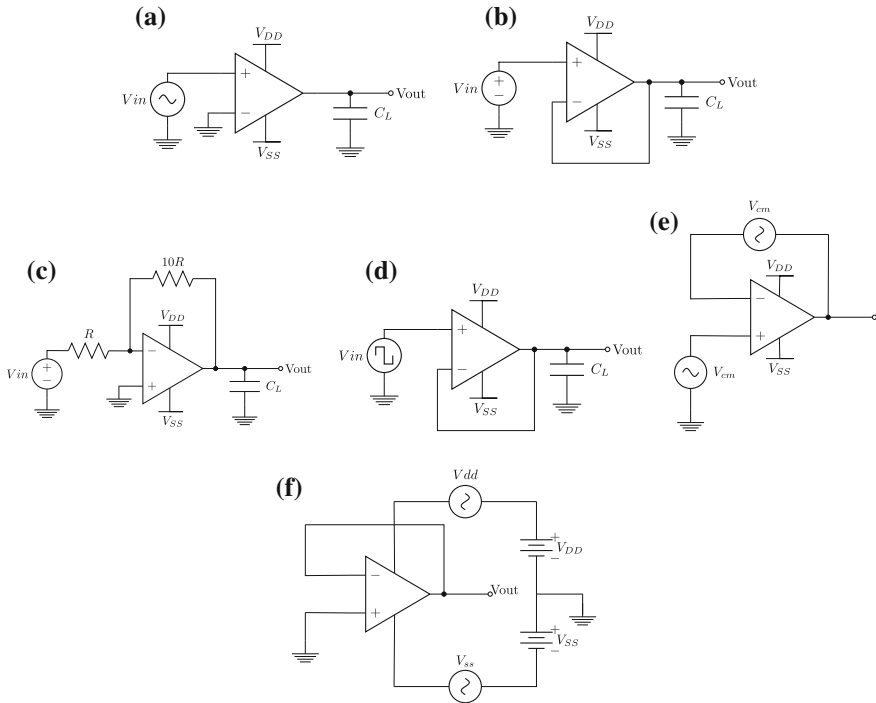
An AC analysis is performed for measuring the low-frequency gain ( $A_{v0}$ ), the gain–bandwidth product (GBW), and the phase margin (PM). The configuration is shown in Fig. 2.12a. The results of this simulation can be plotted as a Bode diagram. From the gain curve,  $A_{v0}$  and GBW specifications are extracted. In the same way, the phase margin is obtained in the phase curve, as shown in Fig. 2.13. In UCAF, this extraction is performed by the “Specification” modular function.

To obtain the input common-mode range (ICMR), the amplifier is connected in unity gain configuration, as shown in Fig. 2.12b. In this simulation, the input voltage is varied from a minimum to a maximum level through a DC analysis. Positive and negative values are obtained from simulation output when the gain is linear.

Figure 2.12c shows a circuit with a voltage gain of  $-10$ . This circuit is used for measuring the output swing (OS) with a DC analysis of input voltage sweep. As the gain is  $-10$ , the output level of saturation is obtained. The difference between the minimum and maximum output levels is the OS specification.

The response speed of an amplifier (Slew Rate) is measured with the same configuration as ICMR. The goal of this simulation is the transient analysis of a step response of the circuit through the verification of the raise or fall behavior of the output voltage level, as illustrated in Fig. 2.12d.

The common-mode rejection ratio (CMRR) is given by the ratio of the common voltage ( $V_{cm}$ ) by the generated output voltage. This specification represents the rejection amount of the input common-mode voltage due to the non-idealities of the amplifier. To measure this specification, an AC analysis is executed using the



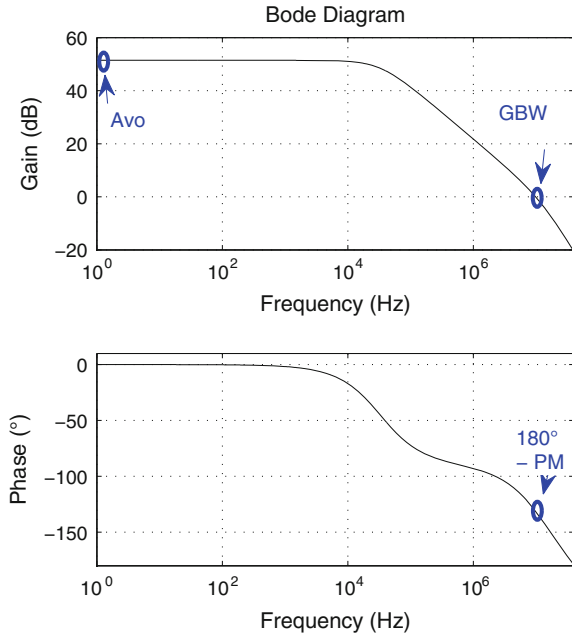
**Fig. 2.12** Implemented test benches for measuring the performance of operational amplifiers. **a** AC open loop. **b** ICMR. **c** Output swing. **d** Slew rate. **e** CMRR. **f** PSRR

configuration shown in Fig. 2.12e, varying the operation frequency of the common-mode voltage source.

Like the CMRR, the power supply rejection ratio (PSRR) indicates the amplifier rejection capacity with respect to the noise coming from the power supply of the circuit. The circuit used to measure the PSRR is shown in Fig. 2.12f. The noise comes from the  $V_{DD}$  and from the  $V_{SS}$  power supplies, resulting in positive (PSRR<sup>+</sup>) and negative (PSRR<sup>-</sup>) rejection ratio, respectively. An AC analysis is executed to sweep the frequency of the voltage sources, simulating the noise coming from the power supplies. It is important to notice that these two simulations are performed separately.

With a multi-core computer architecture, the electrical simulation task can be carried out in parallel in different cores, since each specification has an independent test bench. The UCAF implementation is capable of using all cores simultaneously to simulate the circuit, resulting in a relevant reduction in the overall processing time.

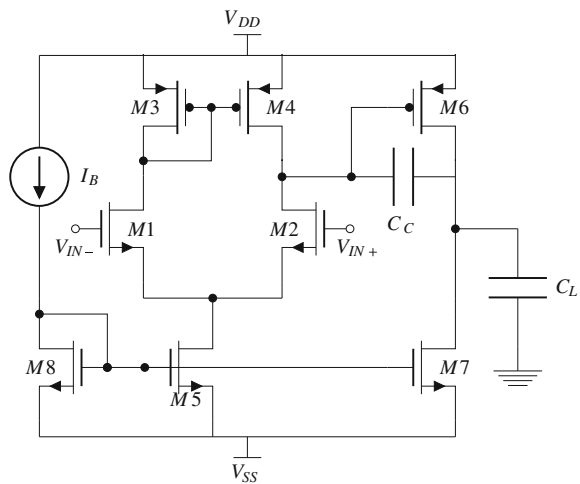
**Fig. 2.13** Bode diagram used to extract low-frequency gain ( $A_{v0}$ ), gain-bandwidth product (GBW), and phase margin (PM)



### 2.5 Automatic Design of a Two-Stage Amplifier

To illustrate the application of the optimization procedure described above, we performed the automatic design of a two-stage CMOS Miller operational transconductance amplifier (OTA). The schematics of this amplifier is shown in Fig. 2.14. It is composed of an input differential amplifier as first amplification stage

**Fig. 2.14** Schematics of a two-stage Miller amplifier





**Table 2.2** Upper and lower bounds for the free values in the design of a Miller OTA

Variable	Lower bound	Upper bound
$W_i$	0.22 $\mu\text{m}$	50.00 $\mu\text{m}$
$L_i$	0.18 $\mu\text{m}$	10.00 $\mu\text{m}$
$I_B$	0.10 $\mu\text{A}$	100.00 $\mu\text{A}$
$C_c$	0.10 pF	10.00 pF

and by an inverter amplifier in the second stage. A compensation capacitor ( $C_c$ ) is connected between stages for stability purposes [2].

This design has the following user specifications as input constraints: low-frequency gain ( $A_{v0}$ ), gain–bandwidth product (GBW), phase margin (PM), slew rate (SR), common-mode input range (ICMR), and output swing (OS). The current mirrors and the differential pair result in the following matched transistors:  $M_1 = M_2$ ,  $M_3 = M_4$  and  $M_5 = M_8$ . Designing this circuit requires calculating all CMOS gate sizes ( $W$  and  $L$ ), the bias current source ( $I_B$ ), and the compensation capacitor ( $C_c$ ), resulting in the following 12 free variables:  $W_1, L_1, W_3, L_3, W_5, L_5, W_6, L_6, W_7, L_7, I_B$ , and  $C_c$ .

The target fabrication technology is XFAB 0.18  $\mu\text{m}$ , which defines the minimal values of transistor sizes:  $L_{\min} = 0.18 \mu\text{m}$  and  $W_{\min} = 0.22 \mu\text{m}$ , with a grid ( $\lambda$ ) of 0.01  $\mu\text{m}$ . The variable bounds are shown in Table 2.2. The design space has 12 dimensions and  $2.76 \times 10^{40}$  possible solutions, and thus, cannot be exhaustively explored with current ordinary computational resources.

**Table 2.3** Design specifications and results for the Miller amplifier

Specification	Required value	Automatic design 1	Automatic design 2	Automatic design 3	Jafari et al. [19]	Liu et al. [21]
$A_{v0}$ (dB)	$\geq 70.00$	73.55	77.25	76.17	82.40	80.66
GBW (MHz)	$\geq 2.00$	2.32	2.39	4.17	9.77	2.04
PM ( $^\circ$ )	$\geq 50.00$	55.69	54.17	66.58	60.00	55.60
SR (V/ $\mu\text{s}$ )	$\geq 5.00$	5.19	8.17	6.01	5.07	1.50
ICMR <sup>+</sup> (V)	$\geq 0.60$	0.76	0.83	0.82	–	–
ICMR <sup>–</sup> (V)	$\leq -0.60$	-0.72	-0.68	-0.64	–	–
OS (V)	$\geq 1.00$	1.17	1.65	1.61	1.17	1.91
Power dissipation ( $\mu\text{W}$ )	Min	25.82	108.49	144.16	52.00	1114.40
Gate area ( $\mu\text{m}^2$ )	Min	16.88	299.61	589.60	236.25	1407.78
Yield (%)	Max	25.62	92.11	100	–	–
Execution	–	43.55	239.05	139.58	–	164.42
Time (min)						

The UCAF tool was set to use genetic algorithms with 100 binary individuals, simple mutation, simple crossover, and roulette wheel selection function. The stop criterion is the execution of 1000 generations, and the initial solution is randomly generated. The cost function has power dissipation and gate area as design objectives, and the remaining user specifications are the constraints. The required values of the specifications are shown in the second column of Table 2.3.

The tool was executed in an Intel i7 processor with 8 GB of main memory. Three different configurations were executed. The performance results are shown in the third to fifth columns of Table 2.3, and device sizes are presented in Table 2.4.

The Automatic Design 1 was performed by UCAF without yield analysis in the optimization procedure. This execution spent 43.55 min, and the final result satisfied all constraints. The resulting power and area are equal to 25.82  $\mu\text{W}$  and 16.88  $\mu\text{m}^2$ , respectively. Comparing with the designs presented by Jafari et al. [19] and Liu et al. [21] for the same circuit, the values obtained by UCAF achieve a considerable reduction in area and dissipated power.

To estimate yield of Automatic Design 1, we executed a Monte Carlo simulation with 2000 samples. The resultant yield was 25.62 %, which is a very low productivity index since approximately 3 of 4 of the fabricated circuits will not satisfy the required specifications. This low yield was expected, because we did not consider it in the optimization problem. The generated solution is very close to the border of the performance space and is very sensitive to random fabrication process variations in this region. Even a small variation causes the solution to move out of the acceptable performance space. For example, the slew rate specification is proportional to the bias current  $I_B$ :

$$\text{SR} = \frac{I_B}{C_c}.$$

At the same time, power dissipation depends on  $I_B$ :

$$P_{\text{diss}} = (V_{\text{DD}} - V_{\text{SS}}) \cdot (2 \cdot I_B + I_7).$$

The optimization algorithm tries to satisfy the minimum SR constraint with the smallest possible  $I_B$ . This implies in reducing SR to the minimum value, moving the

**Table 2.4** Generated solutions for the Miller amplifier designs

Variable	Automatic design 1	Automatic design 2	Automatic design 3
$W_1/L_1$ ( $\mu\text{m}/\mu\text{m}$ )	2.87/0.27	2.89/5.87	38.99/0.58
$W_3/L_3$ ( $\mu\text{m}/\mu\text{m}$ )	5.84/0.21	27.61/1.95	35.97/2.80
$W_5/L_5$ ( $\mu\text{m}/\mu\text{m}$ )	1.42/0.90	8.99/5.39	10.68/0.86
$W_6/L_6$ ( $\mu\text{m}/\mu\text{m}$ )	31.90/0.26	43.87/0.52	31.58/4.55
$W_7/L_7$ ( $\mu\text{m}/\mu\text{m}$ )	2.36/0.82	6.06/6.34	21.12/8.6
$I_B$ ( $\mu\text{A}$ )	2.05	35.02	35.73
$C_c$ (pF)	1.00	1.26	6.29

solution point close to the border of the performance space. When not considering yield prediction during the optimization procedure, all analog IC automatic design tools tend to generate solutions with very low yield.

The other two configurations shown in Table 2.3 include yield prediction in the optimization flow. The Automatic Design 2 was performed using MC simulation for yield prediction for best solution candidates. As formulated in Sect. 2.3, this strategy is used to reduce the number of Monte Carlo simulations. The number of MC samples was fixed in 400. The generated solution satisfied all design constraints with 92.11 % of yield. The optimization process consumed 239.05 min, which is 5.5 times slower than the process without considering yield. However, the final yield increased to a practical value.

The Automatic Design 3 was executed with the same previous configuration, but using the central limit theorem with 99 % confidence interval for determining the number of MC samples in each iteration. The resulting solution showed 100 % of predicted yield in 139.58 min of execution. With less processing time, this configuration allowed a better exploration of the design space, leading to a solution point with 100 % of yield.

The relevant increase in circuit yield in the Automatic Designs 2 and 3 is given at the expense of gate area and power dissipation. This agrees with the Pelgrom's law [28], since the variation in the circuit performance is inversely proportional to the square root of the gate area. Another characteristic of the yield optimization results is that the distances between constraint values and reached specifications are increased. This guarantees that specifications fall inside the performance space even concerning the variations in the fabrication process.

## 2.6 Conclusion

The automatic design of a two-stage Miller amplifier including yield prediction in the optimization flow generated a robust solution with 100 % of yield within a 99 % confidence interval. All performance specifications were met, and gate area and power dissipation were minimized.

The UCAF tool uses genetic algorithms for efficiently searching the design space and produces reliable solutions, since the performance is estimated by means of electrical simulation. The tool executes Monte Carlo simulations for yield prediction, providing a realistic estimation of the performance sensitivity with respect to fabrication process variations. The computational time is reduced by executing Monte Carlo simulation only for best solution candidates and by calculating the minimum number of MC samples for a given theoretical confidence interval.

We implemented an efficient yield-oriented sizing tool that generates robust solutions, contributing for the increase in the number of first-time-right analog integrated circuit designs.

The technique described in this work addresses the optimization of a single basic analog block (a subsystem of a large analog circuit). When maximizing the yield of

the whole circuit, losses might be caused by unmatched interconnections and parasitic effects might appear when integrating subsystems on the top level. Nevertheless, it is necessary to optimize the yield of each subsystem in order to achieve a maximized yield in the whole circuit.

One of the drawbacks of our approach is dealing with large circuits composed by several design variables. The computational cost rapidly increases with the number of design variables, since the design space to be explored grows exponentially. A practical strategy is to divide a large circuit into smaller parts and then size each subcircuit at a time using our proposed methodology. After that, mixed-mode simulation could be considered as an alternative to increase simulation speed for these circuits.

## References

1. Ali, S., Wilcock, R., Wilson, P., Brown, A.: Yield model characterization for analog integrated circuit using Pareto-optimal surface. In: 2008 15th IEEE International Conference on Electronics, Circuits and Systems, IEEE, vol. 2, pp. 1163–1166 (2008). doi:10.1109/ICECS.2008.4675065. url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4675065>
2. Allen, P.E., Holberg, D.R.: CMOS Analog Circuit Design, 2nd edn. Oxford University Press, Oxford (2002)
3. Alpaydin, G., Balkir, S., Dundar, G.: An evolutionary approach to automatic synthesis of high-performance analog integrated circuits. IEEE Trans. Evol. Comput. **7**(3), 240–252 (2003). url: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1206446](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1206446)
4. Antreich, K., Koblitz, R.: Design centering by yield prediction. IEEE Trans. Circ. Syst. **29**(2), 88–96 (1982). doi:10.1109/TCS.1982.1085115. url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1085115>
5. Barros, M., Guilherme, J., Horta, N.: Analog Circuits and Systems Optimization Based on Evolutionary Computation Techniques. Springer, Berlin (2010)
6. Beer, M., Spanos, P.: Neural network based Monte Carlo simulation of random processes. In: Proceedings of the ninth ..., 1995, pp. 2179–2186 (2005). url: [http://www.uncertainty-in-engineering.net/pdf/icossar2005\\_pdf\\_017.pdf](http://www.uncertainty-in-engineering.net/pdf/icossar2005_pdf_017.pdf)
7. Chiang, C.C., Kawa, J.: Design for Manufacturability and Yield for Nano-Scale CMOS. Springer, Netherlands (2007)
8. De Smedt, B., Gielen, G.: Watson: design space boundary exploration and model generation for analog and RF IC design. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **22**(2), 213–224 (2003). doi:10.1109/TCAD.2002.806598. url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1174096>
9. Director, S., Feldmann, P., Krishna, K.: Statistical integrated circuit design. IEEE J. Solid State Circ. **28**(3), 193–202 (1993). doi:10.1109/4.209985. url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=209985>
10. Drennan, P.G., McAndrew, C.C.: Understanding MOSFET mismatch for analog design. IEEE J. Solid State Circ. **38**(3), 450–456 (2003)
11. Floudas, C.A., Pardalos, P.M.: Encyclopedia of Optimization, vol. 1. Springer, Berlin (2008)
12. Gielen, G., Rutenbar, R.A.: Computer-aided design of analog and mixed-signal integrated circuits. Proc. IEEE **88**, 1825–1852 (2000)

13. Gong, F., Basir-Kazeruni, S., He, L., Yu, H.: Stochastic behavioral modeling and analysis for analog/mixed-signal circuits. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **32**(1), 24–33 (2013). doi:10.1109/TCAD.2012.2217961. url: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6387696](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6387696), <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6387696>
14. Herbig, V.: Getting to ‘First Time Right’ in Analog/Mixed-Signal Designs. *EE Times*, Europe (2008). url: [http://www.eetimes.com/document.asp?doc\\_id=1271601](http://www.eetimes.com/document.asp?doc_id=1271601)
15. Hershenson, M.D.M., Boyd, S.P., Lee, T.H.: Optimal design of a CMOS Op-Amp via geometric programming. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **20**(1), 1–21 (2001)
16. Houck, C.R., Joines, J., Kay, M.G.: A genetic algorithm for function optimization: a matlab implementation. *NCSU-IE TR* **95**(09) (1995)
17. Houck, C.R., Joines, J.A., Kay, M.G.: A Genetic Algorithm for Function Optimization: A Matlab Implementation. Tech. rep., North Carolina State University (1996)
18. Huss, S.: Analog circuit synthesis: a search for the Holy Grail? In: 2006 IEEE International Symposium on Circuits and Systems, IEEE, pp. 1463–1466. doi:10.1109/ISCAS.2006.1692872. url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1692872>
19. Jafari, A., Sadri, S., Zekri, M.: Design optimization of analog integrated circuits by using artificial neural networks. In: 2010 International Conference of Soft Computing and Pattern Recognition, IEEE, pp. 385–388 (2010). doi:10.1109/SOCPAR.2010.5686736, URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5686736>
20. Lin, Y., Chen, D., Geiger, R.: Yield enhancement with optimal area allocation for ratio-critical analog circuits. *IEEE Trans. Circ. Syst. I: Regul. Pap.* **53**(3), 534–553 (2006). doi:10.1109/TCSI.2005.858761. url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1610852>
21. Liu, B., Wang, Y., Yu, Z., Liu, L., Li, M., Wang, Z., Lu, J., Fernández, F.V.: Analog circuit optimization system based on hybrid evolutionary algorithms. *Integr. VLSI J.* **42**(2), 137–148 (2009). doi:10.1016/j.vlsi.2008.04.003. url: <http://linkinghub.elsevier.com/retrieve/pii/S0167926008000126>
22. Liu, B., Fernandez, F.V., Gielen, G.: Efficient and accurate statistical analog yield optimization and variation-aware circuit sizing based on computational intelligence techniques. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **30**(6), 793–805 (2011). doi:10.1109/TCAD.2011.2106850. url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5768139>
23. Mandal, P., Visvanathan, V.: CMOS Op-Amp sizing using a geometric programming formulation. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **20**(1), 22–38 (2001)
24. Martins, R.M.F., Lourenço, N.C.C., Horta, N.C.G.: Introduction. In: *Generating Analog IC Layouts with LAYGEN II*, SpringerBriefs in Applied Sciences and Technology. Springer, Berlin, pp. 1–7 (2013). doi:10.1007/978-3-642-33146-6\_1. url: [http://dx.doi.org/10.1007/978-3-642-33146-6\\_1](http://dx.doi.org/10.1007/978-3-642-33146-6_1)
25. Mueller-Gritschneider, D., Graeb, H.: Computation of yield-optimized Pareto fronts for analog integrated circuit specifications. In: 2010 Design, Automation & Test in Europe Conference & Exhibition, IEEE, pp. 1088–1093 (2010). doi:10.1109/DATE.2010.5456971. url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5456971>
26. Nye, W., Riley, D.C., Sangiovanni-Vincentelli, A., Tits, A.L.: DELIGHT.SPICE: an optimization-based system for the design of integrated circuits. *IEEE Tran. Comput. Aided Des.* **7**(4), 501–519 (1988)
27. Orshansky, M., Nassif, S.R., Boning, D.: *Design for Manufacturability and Statistical Design*. Springer, Berlin (2008). url: <http://link.springer.com/book/10.1007/978-0-387-69011-7/page/1>
28. Pelgrom MJM, Duinmaijer ACJ, Welbers APG (1989) Matching properties of MOS transistors. *IEEE J. Solid State Circ.* **24**(5), 1433–1439
29. Phelps, R., Krasnicki, M., Rutenbar, R.A., Carley, L.R., Hellums, J.R.: Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **19**(6), 703–717 (2000)

30. Razavi, B.: Design of Analog CMOS Integrated Circuits, vol 6. McGraw-Hill, New York City. doi:[10.1111/j.1151-2916.1994.tb07040.x](https://doi.org/10.1111/j.1151-2916.1994.tb07040.x)
31. Sengupta, M., Saxena, S., Daldoss, L., Kramer, G., Minehane, S.: Application-specific worst case corners using response surfaces and statistical models. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **24**(9), 1372–1380 (2005). doi:[10.1109/TCAD.2005.852037](https://doi.org/10.1109/TCAD.2005.852037). url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1501902>
32. Singhee, A., Rutenbar, R.: Why Quasi-Monte Carlo is better than Monte Carlo or latin hypercube sampling for statistical circuit analysis. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **29**(11), 1763–1776 (2010) doi:[10.1109/TCAD.2010.2062750](https://doi.org/10.1109/TCAD.2010.2062750). url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5605333>
33. Stefanovic, D., Kayal, M.: *Structured Analog CMOS Design*, 1st edn., Springer, Berlin (2008)
34. Tsvividis, Y.: *Operation and Modeling of the MOS Transistor*, 2nd edn. Oxford University Press, Oxford (1999)
35. Xu, Y., Hsiung, K.L., Li, X., Pileggi, L.T., Boyd, S.P.: Regular analog/RF integrated circuits design using optimization with recourse including ellipsoidal uncertainty. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **28**(5), 623–637 (2009)
36. Yu, G., Li, P.: Hierarchical analog/mixed-signal circuit optimization under process variations and tuning. *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.* **30**(2), 313–317 (2011). doi:[10.1109/TCAD.2010.2071250](https://doi.org/10.1109/TCAD.2010.2071250). url: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5689353>

# Chapter 3

## Application of Computational Intelligence Techniques to Maximize Unpredictability in Multiscroll Chaotic Oscillators

Victor Hugo Carbajal-Gómez, Esteban Tlelo-Cuautle  
and Francisco V. Fernández

**Abstract** This chapter applies and compares three computational intelligence algorithms—genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO)—to maximize the positive Lyapunov exponent in a multiscroll chaotic oscillator based on a saturated nonlinear function series based on the modification of the standard settings of the coefficient values of the mathematical description, and taking into account the correct distribution of the scrolls drawing the phase-space diagram. The experimental results show that the DE and PSO algorithms help to maximize the positive Lyapunov exponent of truncated coefficients over the continuous spaces.

### 3.1 Introduction

Some nonlinear systems show chaotic behavior, which is a bounded unstable dynamic behavior that exhibits sensitive dependence on initial conditions and includes infinite unstable periodic motions. Although it appears to be stochastic, it occurs in a deterministic nonlinear system under deterministic conditions.

Nonlinear science has had quite a triumph in all conceivable applications in science and technology. Generation of multiscroll chaotic attractors has received considerable attention for more than a decade; such interest is both theoretical and practical [17, 18] and has been an attractive field for research in various areas, among them, physics, communications, and electronics [4, 5, 19, 23].

Chaotic oscillators have been investigated to generate multiscroll attractors. Some of them can be modeled by piecewise-linear (PWL) approaches, so that the

---

V.H. Carbajal-Gómez · E. Tlelo-Cuautle  
INAOE, Puebla, Mexico  
e-mail: vhcabjal@inaoep.mx

F.V. Fernández (✉)  
IMSE-CNM, CSIC and Universidad de Sevilla, Sevilla, Spain  
e-mail: pacov@imse-cnm.csic.es

nonlinear problem can be transformed into a linear one. However, many research challenges remain, for example: how to understand when a deterministic dynamical system might exhibit chaotic behavior, the required conditions of this behavior, the ways available to control it, the ways to implement it with electronic devices, and the practical and theoretical implications that follow.

Every new chaotic system [17, 19, 27] is a candidate to improve engineering applications. The circuit implementation of reliable nonlinear circuits, for generating various complex chaotic signals, is a key issue for future potential application to communications, cryptography, and neural networks [5], particularly in designing secure communication systems [3, 9, 11].

The main characterizations of chaotic systems are fractal dimension, Kolmogorov–Sinai entropy, and Lyapunov spectrum [14, 20]. Among them, the Lyapunov exponents provide a means of ascertaining whether the behavior of a system is chaotic. In this manner, the presence of positive Lyapunov exponents has often been taken as a signature of chaotic motion. In addition, a high value of the positive Lyapunov exponent indicates a high degree of unpredictability of the system; therefore, the system has a more complex dynamic behavior [22].

This chapter is organized as follows. Fundamentals of multiscroll chaotic oscillators and their controlling parameters are presented in Sect. 3.2. The procedure for computing the positive Lyapunov exponent is given in Sect. 3.3. The exploration of the parameter space to maximize the positive Lyapunov exponent is performed by applying computational intelligence techniques, briefly introduced in Sect. 3.4. Section 3.5 presents experimental results of standard implementations of various optimization algorithms: genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO). Robustness of these algorithms is studied statistically, showing a high effectiveness in the maximization of the positive Lyapunov exponent. The phase-space portraits (PSP) of non-optimized and optimized chaotic oscillators are also compared. Finally, Sect. 3.6 summarizes the conclusions.

## 3.2 Multiscroll Chaotic Oscillator

A multiscroll chaotic oscillator can be described by the system of differential equations given in (3.1) [4, 9, 17], where  $a, b, c$ , and  $d_1$  are the positive constants and can have values in the interval  $[0, 1]$ . The system is controlled by the PWL approximation, e.g., a saturated function series  $f$ ,

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= -ax_1 - bx_2 - cx_3 + d_1f(x_1; m) \end{aligned} \tag{3.1}$$

Now, it will be described how the saturated function  $f$  in (3.1) is obtained in detail. Let  $f_0$  be the saturated function:



$$f_0(x_1; m) = \begin{cases} 1, & \text{if } x_1 > m \\ \frac{x_1}{m}, & \text{if } |x_1| \leq m \\ -1, & \text{if } x_1 < -m \end{cases} \quad (3.2)$$

where  $\frac{1}{m}$  is the slope of the middle segment and  $m > 0$ . The upper radial  $\{f_0(x_1; m) = 1 | x_1 > m\}$ , and the lower radial  $\{f_0(x_1; m) = -1 | x_1 < -m\}$  are called *saturated plateaus*, and the segment  $\{f_0(x_1; m) = \frac{x_1}{m} | |x_1| \leq m\}$  between the two saturated plateaus is called *saturated slope* [19]. Let us consider also the saturated functions  $f_h$  and  $f_{-h}$  defined as:

$$f_h(x_1; m, h) = \begin{cases} 2, & \text{if } x_1 > h + m \\ \frac{x_1 - h}{m}, & \text{if } |x_1 - h| \leq m \\ 0, & \text{if } x_1 < h - m \end{cases} \quad (3.3)$$

and

$$f_{-h}(x_1; m, -h) = \begin{cases} 0, & \text{if } x_1 > h + m \\ \frac{x_1 - h}{m}, & \text{if } |x_1 - h| \leq m \\ -2, & \text{if } x_1 < h - m \end{cases} \quad (3.4)$$

where  $h$  is called the *saturated delay time* and  $h > m$ . Therefore, a saturated function series for a chaotic oscillator with  $s$  scrolls is defined as the function:

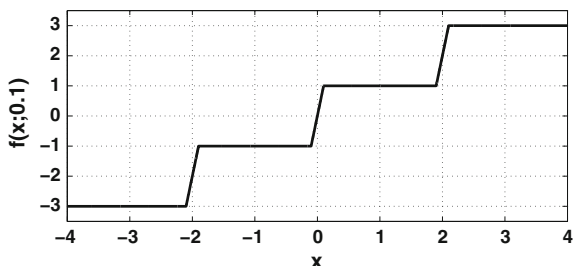
$$f(x_1; m) = \sum_{i=0}^{s-2} f_{2i-s+2}(x_1; m, 2i-s+2), \quad (3.5)$$

where  $s > 2$ .

For example, using  $f = f_0$  in (3.1), a 2-scroll chaotic oscillator can be generated. Therefore, the saturated function series to generate 3 scrolls is  $f(x_1; m) = f_{-1}(x_1; m, -1) + f_1(x_1; m, 1)$ . To generate a 4-scroll oscillator, it will be  $f(x_1; m) = f_{-2}(x_1; m, -2) + f_0(x_1; m) + f_2(x_1; m, 2)$ , and so on. This function series are shown in Fig. 3.1 for  $m = 0.1$ . Note that the value of  $h$  in (3.3) and (3.4) represents the center of the saturated slopes.

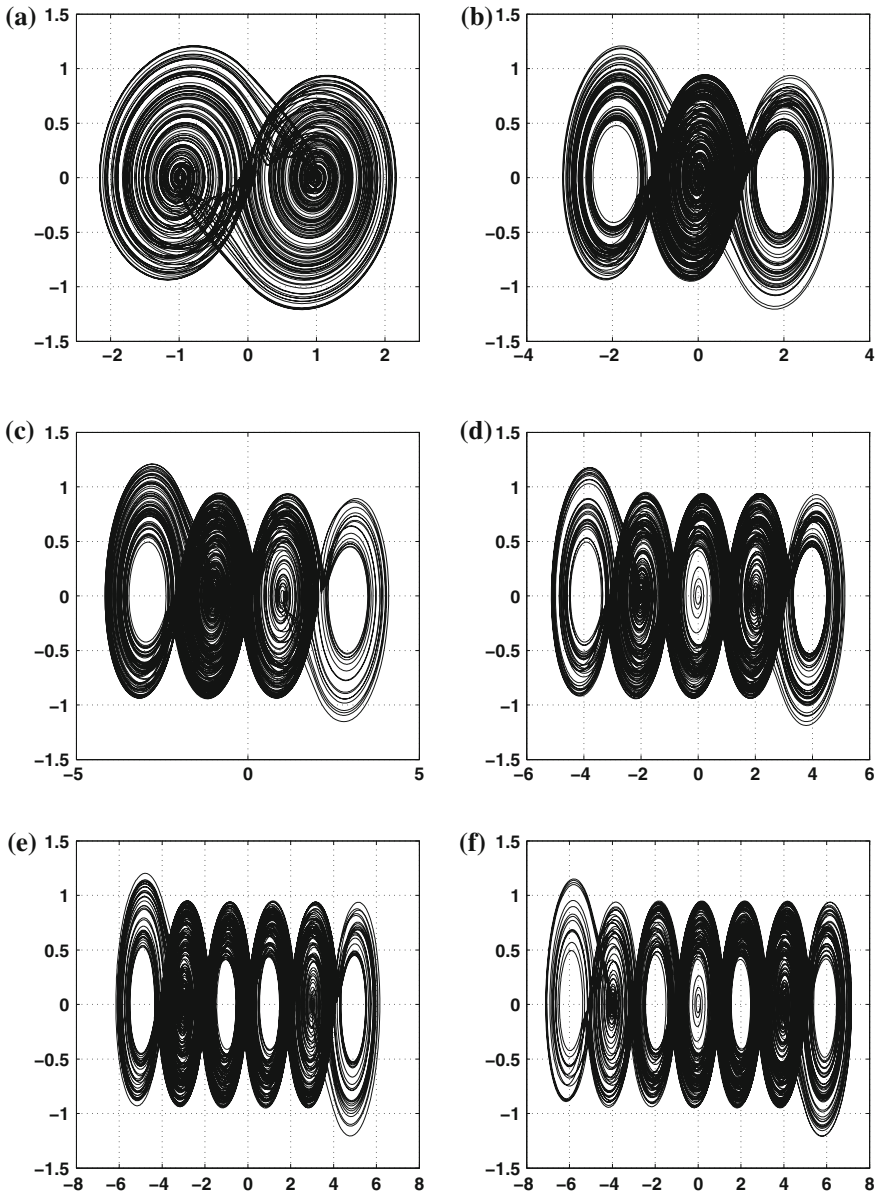
Figure 3.2 shows the simulation of 2- to 7-scroll chaotic oscillator attractors modeled by (3.1).

**Fig. 3.1** PWL-saturated function series to generate 4 scrolls



### 3.3 Computing Lyapunov Exponents

The deterministic, still unpredictable, behavior of nonlinear dissipative dynamical systems is an important subject in more and more fields of science, from mathematics to biology, and even in engineering. The Lyapunov exponents give the most



**Fig. 3.2** Phase-space portraits of chaotic oscillators with **a** 2-scrolls, **b** 3-scrolls, **c** 4-scrolls, **d** 5-scrolls, **e** 6-scrolls and **f** 7-scrolls

characteristic description of the presence of a deterministic nonperiodic flow. Therefore, Lyapunov exponents are asymptotic measures characterizing the average rate of growth (or shrinkage) of small perturbations to the solutions of a dynamical system [30]. Lyapunov exponents provide quantitative measures of response sensitivity of a dynamical system to small changes in initial conditions [10]. The number of Lyapunov exponents equals the number of state variables, and if at least one is positive, this is an indication of chaos [10, 20, 22]. That way, an algorithm capable of computing the Lyapunov exponents in a simple fashion is very much in need to guarantee chaotic regime.

Let us consider an  $n$ -dimensional dynamical system:

$$\dot{x} = f(x) \quad t > 0 \quad x(0) = x_0 \in \mathbb{R}^n \quad (3.6)$$

where  $x$  and  $f$  are  $n$ -dimensional vector fields. To determine the  $n$ -Lyapunov exponents of the system, one should find the long-term evolution of small perturbations to a trajectory, which are determined by the variational equation of (3.6),

$$\dot{y} = \frac{\partial f}{\partial x}(x(t))y = J(x(t))y \quad (3.7)$$

where  $J$  is the  $n \times n$  Jacobian matrix of  $f$ . A solution of (3.7) with a given initial perturbation  $y(0)$  can be written as:

$$y(t) = Y(t)y(0) \quad (3.8)$$

with  $Y(t)$  as the fundamental solution satisfying

$$\dot{Y} = J(x(t))Y \quad Y(0) = I_n \quad (3.9)$$

In (3.9),  $I_n$  is the  $n \times n$  identity matrix. By considering the evolution of an infinitesimal  $n$ -parallelepiped  $[p_1(t), \dots, p_n(t)]$  with the axis  $p_i(t) = Y(t)p_i(0)$  for  $i = 1, \dots, n$ , where  $p_i(0)$  denotes an orthogonal basis of  $\mathbb{R}^n$ , then the  $i$ th Lyapunov exponent, which measures the long-time sensitivity of the flow  $x(t)$  with respect to the initial data  $x(0)$  at the direction  $p_i(t)$ , is defined by the expansion rate of the length of the  $i$ th axis  $p_i(t)$  and is given by

$$\lambda_i = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \|p_i(t)\| \quad (3.10)$$

The Lyapunov exponents can be computed by applying the methods given in [10, 20, 22].

To measure the three Lyapunov exponents of the original chaotic oscillator system in (3.1), this original system is observed by expanding it with other three systems that change according to the derivative of (3.1). If  $\mathbf{u} = [\dot{x}, \dot{y}, \dot{z}]^T$ ,  $\mathbf{u} \in \mathbb{R}^3$ , represents one state of the original dynamical system at any  $t > 0$ , the state of the new observed system will be  $\mathbf{v} = [\mathbf{u}, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]^T$ ,  $\mathbf{v} \in \mathbb{R}^{12}$ , where  $\mathbf{u}_i$ , for  $i = 1, 2, 3$ ,

are the three added systems that will measure precisely the change of those small perturbations on each orthogonal directions, for each of the three state variables in (3.1). The initial state of the expanded system is set to

$$\begin{aligned} \mathbf{v}_0 &\in \mathbb{R}^{12} \\ \mathbf{v}_0 &= [\mathbf{u}_0^T, \mathbf{e}_1^T, \mathbf{e}_2^T, \mathbf{e}_3^T]^T \end{aligned} \quad (3.11)$$

where  $\mathbf{u}_0$  is the vector of initial conditions  $[x_0, y_0, z_0]^T$ ;  $[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3] = I$ ; and  $I$  is the identity matrix of size  $3 \times 3$ . Thus,  $\mathbf{e}_i$ , for  $i = 1, 2, 3$ , are each unitary column vectors of the identity matrix  $I$ .

The observational system is integrated by several steps until a period  $T_O$  is reached. After this step, the state of the variational system is orthonormalized by using the standard Gram–Schmidt method [14]. The next integration is carried out by using the new orthonormalized vectors as initial conditions.

The Lyapunov exponents measure the long-time sensitivity of the flow in  $\mathbf{u}$  with respect to the initial data  $\mathbf{u}_0$  at the directions of every orthonormalized vector. This measure is taken when the variational system is orthonormalized. If  $\mathbf{v} = [\mathbf{u}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]^T$  is the state after the matrix  $[\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]$  is orthonormalized, the LE  $\lambda_i$ , for  $i = 1, 2, 3$ , is calculated by

$$\lambda_i \approx \frac{1}{T} \sum_{j=1}^k \ln \|\mathbf{p}_i\| \quad (3.12)$$

where the number of summations  $k$  is calculated as  $\lceil \frac{T}{T_O} \rceil$ , and  $T$  is the simulation time.

For instance, in [2], the period of time  $T_O$  is selected by using the minimum absolute value of all the eigenvalues of the system as:

$$T_O = \frac{1}{l_{\min}}$$

where  $l_{\min}$  represents the value of the minimum eigenvalue of the system in (3.1) [26].

### 3.4 Optimization Algorithms

A global optimization problem can be formulated as the minimization of the function

$$f : \mathbb{R}^D \rightarrow \mathbb{R} \quad (3.13)$$

$$f(x), \quad \text{such that } x_j \in [l_j, u_j], \quad j = 1, \dots, D$$

where  $f$  is the objective function, and  $x$  is a continuous variable vector of  $D$  dimensions. The feasible domain of variable  $x$  is defined by specifying upper ( $u_j$ ) and lower ( $l_j$ ) limits of each component  $j$ .

To solve the optimization problem in (3.13), efficient search or optimization algorithms are needed. There are many optimization algorithms which can be classified in many ways, depending on the focus and characteristics [1, 12].

If the derivative or gradient of a function is the focus, optimization can be classified into gradient-based algorithms and derivative-free or gradient-free algorithms. Gradient-based algorithms use derivative information, and they are often very efficient. Derivative-free algorithms do not use any derivative information but the values of the function itself. Some functions may have discontinuities or it may be expensive to calculate derivatives accurately, and thus derivative-free algorithms become very useful [25]. From a different perspective, optimization algorithms can be classified into trajectory-based and population-based algorithms. A trajectory-based algorithm typically uses a single agent or one solution at a time, which will trace out a path as the iterations continue. On the other hand, population-based algorithms use multiple agents which will interact and trace out multiple paths [29]. Optimization algorithms can also be classified as deterministic or stochastic. If an algorithm works in a systematic deterministic manner without any random nature, it is called deterministic. For such an algorithm, it will reach the same final solution if we start with the same initial point. On the other hand, if there is some randomness in the algorithm, the algorithm will usually reach a different point every time the algorithm is executed, even though the same initial point is used.

Genetic algorithms (GA), differential evolution (DE), and particle swarm optimization (PSO) are evolutionary computation algorithms. They work with a population of tentative solutions to the problem, and new solutions are generated by somehow combining the information of the old ones and by surviving the ones with better fitness. These algorithms are used as solver for global optimizations problems, more commonly in problems with continuous representations [25, 29].

The usefulness of these evolutionary algorithms relies in the fact that they need only the value of function  $f$  to work, or in other words, it is not necessary that  $f$  be continuous or get any information about the derivative of function  $f$ .

### 3.4.1 Genetic Algorithm

Genetic algorithms (GAs) are probably the most popular evolutionary algorithms with a diverse range of applications. A vast majority of well-known optimization problems have been solved by genetic algorithms. In addition, genetic algorithms are population-based and many modern evolutionary algorithms are directly based

on, or have strong similarities to, genetic algorithms [13]. Genetic algorithms, developed by John Holland and his collaborators in the 1960s and 1970s [15], are a model or abstraction of biological evolution based on Charles Darwin's theory of natural selection. GAs operate on the principle of "survival of the fittest." In this manner, a GA has the capability to generate new design solutions from a population of existing solutions and to discard the solutions which have an inferior performance or fitness. Holland was the first to use recombination, mutation, and selection in the study of adaptive and artificial systems. These genetic operators are the essential components of genetic algorithms as a problem-solving strategy [15, 25].

This is often done through the following procedure: (1) definition of an encoding scheme; (2) definition of a fitness function or selection criterion; (3) creation of a population of chromosomes; (4) evaluation of the fitness of every chromosome in the population; (5) creation of a new population by performing fitness-proportionate selection, crossover, and mutation; (6) replacement of the old population by the new one. Steps (4), (5), and (6) are then repeated for a number of generations. At the end, the best chromosome is decoded to obtain a solution to the problem.

Each iteration, which leads to a new population, is called a generation. Fixed-length chromosomes are used in most of the genetic algorithms at each generation although there is substantial research on variable-length structures. The coding of the objective function is usually in the form of binary arrays or real-valued arrays in genetic algorithms. An important issue is the formulation or

---

### Algorithm 3.1 Genetic Algorithm

---

```

1:  $N$  is the number of individuals
2:  $G$  is the number of iterations (generations)
3: Variable bounds  $x_i \in [l_i, u_i]$ , for  $i = 1, 2, \dots, D$ 
4: Procedure GA ( $N, G, l_i, u_i$ )
5: for  $i=1:N$  do
6:   for  $d=1:D$  do
7:      $x_i[d] = l_d + (u_d - l_d) \cdot \text{rand}()$ 
8:   end for
9:    $Pop \leftarrow x_i[d]$ 
10:   $x_i.\text{fit} \leftarrow \text{evaluate } Pop$ 
11: end for
12: for  $i=1:G$  do
13:   Rank the best  $N/2$  solutions in  $Pop$  and save them in  $Pop_1$        $\triangleright$  Elitism based selection
14:   Randomly select two solutions  $x_A$  and  $x_B$  from  $Pop$                $\triangleright$  Crossover
15:   generate  $x_C$  and  $x_D$  by one-point crossover to  $x_A$  and  $x_B$ 
16:   save  $x_C$  and  $x_D$  to  $Pop_2$ 
17:   for  $i = 1 : N/2$  do
18:     Select a solution  $x_j$  from  $Pop_2$ 
19:     mutate each bit of  $x_j$  under the rate  $PM$  and generate a new solution  $x'_j$    $\triangleright$  Mutation
20:     if  $x'_j$  is unfeasible then
21:       update  $x'_j$  with a feasible solution by repairing  $x'_j$ 
22:     end if
23:     update  $x_j$  with  $x'_j$  in  $Pop_2$ 
24:   end for
25:    $Pop = Pop_1 + Pop_2$ 
26: end for
27: return the best solution  $x_i[d]$  in  $Pop$ 

```

---

choice of an appropriate fitness function that determines the selection criterion in a particular problem.

The structure and the steps that execute the proposed GA are highlighted in the pseudocode depicted in Algorithm 3.1.

### 3.4.2 Differential Evolution Algorithm

Differential evolution (DE) was developed by Storn and Price [24]. It is a vector-based evolutionary algorithm, and unlike genetic algorithms, differential evolution carries out operations over each component (or each dimension of the solution). Almost everything is done in terms of vectors, and DE can be viewed as a self-organizing search, directed toward the optimum.

DE is an evolutionary algorithm that works with a population of tentative solutions to the problem, and new solutions are generated by combining the old ones and by surviving the ones with better fitness [29].

The general convention used to denote the DE strategy is DE/ $x/y/z$ . DE stands for differential evolution algorithm,  $x$  represents a string denoting the vector to be perturbed,  $y$  is the number of difference vectors considered for perturbation of  $x$ , and  $z$  is the type of crossover being used (exp: exponential; bin: binomial).

We use the most common version of DE: DE/rand/1/bin. Hence, the perturbation is on any randomly chosen vector, for perturbation a single vector difference is used, and the type of crossover is binomial. For perturbation with a single vector difference, out of three distinct randomly chosen vectors, the weighted vector difference of any two vectors is added to the third one. In binomial crossover, the crossover is performed on each of the  $D$  variables whenever a randomly picked number between 0 and 1 is below a certain threshold  $R$ . The pseudocode of DE is shown in Algorithm 3.2. Each individual is represented by a vector  $x \in \mathbb{R}^D$ , and its fitness value is represented as  $x.fit$ . The location  $j$  of individual  $i$  is represented as  $x_i[j]$ .  $rand()$  is a function that returns a random number greater or equal to zero and less than one.  $evaluate()$  is a function that calculates de-fitness function (function to be optimized) [21]. The core of DE is in the loop on lines 15–24: A mutated individual is generated from three different randomly chosen individuals; each value of the new vector (a new individual) is calculated from the first individual, plus the difference of the other two individuals multiplied by  $F$ , the difference constant; the new vector value is calculated if random real number (between zero and one) is less than  $R$ , the DE's recombination constant. To prevent the case when the new individual is equal to the first reference individual, at least one vector component is forced to be calculated from the mutated vector, it is in line 16 of the pseudocode, when  $d = jrand$ , and  $jrand$  is an integer random number between 1 and  $D$ . Then, the new individual is evaluated. If it is better than the original one (in line 26), then the child replaces it (line 27).

---

**Algorithm 3.2** Differential Evolution algorithm
 

---

```

1:  $N$  is the number of individuals
2:  $G$  is the number of iterations (generations)
3: Variable bounds  $x_i \in [l_i, u_i]$ , for  $i = 1, 2, \dots, D$ 
4: Procedure DE ( $N, G, \{l_i\}, \{u_i\}$ )
5: for  $i = 1 : N$  do
6:   for  $d = 1 : D$  do
7:      $x_i[d] = l_d + (u_d - l_d) \cdot \text{rand}()$ 
8:   end for
9:    $x_i.\text{fit} \leftarrow \text{evaluate}(x_i)$ 
10: end for
11: for  $i = 1 : G$  do
12:   Let  $j_1, j_2$  and  $j_3$  be three random numbers in  $\{1, N\}$ 
13:   without replacement and also different to  $i$ .
14:    $j_{\text{rand}} \leftarrow [\text{rand}() \cdot D] + 1$ 
15:   for  $d = 1 : D$  do
16:     if  $\text{rand}() < R$  OR  $d = j_{\text{rand}}$  then
17:        $y[d] = x_{i2}[d] + F(x_{i0}[d] - x_{i1}[d])$ 
18:       if  $y[d] < l_d$  OR  $y[d] > u_d$  then
19:          $y[d] = l_d + (u_d - l_d) \cdot \text{rand}()$ 
20:       end if
21:     else
22:        $y[d] = x_i[d]$ 
23:     end if
24:   end for
25:    $y.\text{fit} = \text{evaluate}(y)$ 
26:   if  $y.\text{fit} < x_i.\text{fit}$  then
27:      $x_i \leftarrow y; x_i.\text{fit} \leftarrow y.\text{fit}$ 
28:   end if
29: end for
30: search  $\mathbf{q} = x_k | \min(x_k.\text{fit})$ , for  $k = 1, 2, \dots, N$ 
31:  $\mathbf{q}$  is the solution at iteration  $i$ 

```

---

### 3.4.3 Particle Swarm Optimization

Particle swarm optimization (PSO) was developed by Kennedy and Eberhart in 1995 [16] and was inspired on swarm behavior observed in nature such as fish and bird schooling. Since then, PSO has generated a lot of attention, and now forms an exciting, ever-expanding research subject in the field of swarm intelligence. PSO has been applied to almost every area in optimization, computational intelligence, and design applications. There are at least two dozens of PSO variants, as well as hybrid algorithms obtained by combining PSO with other existing algorithms, which are also increasingly popular [1, 25].



PSO searches the space of an objective function by adjusting the trajectories of individual agents, called *particles*. Each *particle* traces a piecewise path which can be modeled as a time-dependent positional vector. The movement of a swarming particle consists of two major components: a stochastic component and a deterministic component. Each particle is attracted toward the position of the current global best  $pbest_i.pos_d$  and its own best known location  $pbest_{gbest[i]}.pos$  in history.

When a particle finds a location that is better than any previously found locations, then it updates this location as the new current best for particle  $i$ . There is a current best for all  $N$  particles at any time  $t$  at each iteration. The aim is to find the global best among all the current best solutions until the objective no longer improves or after a certain number of iterations [6].

The pseudocode for PSO is shown in Algorithm 3.3. Each particle  $p_i$  has three associated values: position  $p_i.pos$ , velocity  $p_i.vel$ , and value of the fitness function  $p_i.fit$ . Particle  $pbest$  has only position and fitness function value.  $gbest[]$  is a vector that stores indexes to reference  $pbest$  particles.  $rand()$  is a function that returns a random number greater or equal to zero and less than one.  $evaluate()$  is a function that calculates the value of the fitness for the problem to solve. This PSO version was inspired from [6, 7, 16]. The main advantage of this PSO algorithm (not using extra parameters) consists on having only the essential parameters, i.e., the number of individuals (particles) and the number of iterations (generations). Particles position  $p_i$  are initialized randomly and also their velocities (in lines 5–10 and 11–15 in Algorithm 3.3, respectively). Each particle is evaluated and  $pbest_i$  particles are initialized equal to the  $p_i$  ones. For a given number of iterations, the following process is applied: (1) three random numbers are calculated in  $[1, N]$  ( $N = \text{population size}$ ) with replacement;  $gbest[i]$  points to the best particle inside this cluster of three particles. (2) A new particle is calculated and its velocity is updated (line 22–23). If this new particle is better than its associated  $pbest$ , then  $pbest$  particle takes the values of the new particle. The core of PSO is in the loop of lines 17–23. The update rules are:

$$p_i \cdot pos_d \leftarrow wp_i \cdot vel_d + \varphi_1 U_1 (pbest_i \cdot pos_d - p_i \cdot pos_d) \\ + \varphi_2 U_2 (pbest_{gbest[i]} \cdot pos_d - p_i \cdot pos_d)$$

where  $w$  is a parameter called inertia weight,  $\varphi_1$  and  $\varphi_2$  are two parameters so-called acceleration coefficients,  $U_1$  and  $U_2$  are two random numbers uniformly distributed in the interval  $[0, 1)$ .

**Algorithm 3.3** Particle swarm optimization algorithm.

---

```

1:  $N$  is the number of particles
2:  $G$  is the number of iterations (generations)
3: Variable bounds  $x_i \in [l_i, u_i]$ , for  $i = 1, 2, \dots, D$ 
4: Procedure PSO ( $N, G, \{l_i\}, \{u_i\}$ )
5: for  $i = 1 : N$  do                                     ▷ Initialize particles positions
6:   for  $d = 1 : D$  do
7:      $p_i.pos_d = l_d + (u_d - l_d) \cdot rand()$ 
8:      $pbest_i.pos_d \leftarrow p_i.pos_d$ 
9:      $p_i.fit \leftarrow evaluate(p_i.pos)$ 
10:     $pbest_i.fit \leftarrow p_i.fit$ 
11:   for  $i = N : D$  do                                     ▷ Initialize particles velocities
12:     for  $d = 1 : D$  do
13:        $vmin = l_d - p_i.pos_d$ 
14:        $vmax = u_d - p_i.pos_d$ 
15:        $p_i.vel_d = vmin + (vmax - vmin) \cdot rand()$ 
16:     for  $g = 1 : G$  do                                     ▷ Iterate G generations
17:       for  $i = 1 : N$  do                                     ▷ For each particle
18:         Let  $j_1, j_2$  and  $j_3$  be three random numbers in  $\{1, N\}$ 
19:          $gbest[i] = k | \min(pbest_k.fit)$ , for  $k \in \{i, j_1, j_2, j_3\}$ 
20:       for  $i = N : D$  do                                     ▷ For each particle
21:         for  $d = 1 : D$  do                                     ▷ For each dimension
22:            $p_i.pos_d \leftarrow w p_i.vel_d + \phi_1 U_1(pbest_i.pos_d - p_i.pos_d) + \phi_2 U_2(pbest_{gbest[i]}.pos_d - p_i.pos_d)$ 
23:            $p_i.vel_d \leftarrow p_i.pos_d + p_i.vel_d$ 
24:           if  $p_i.pos_d < l_d$ 
25:              $p_i.pos_d = l_d$ ;  $p_i.vel_d = 0$ 
26:           if  $p_i.pos_d > u_d$  then
27:              $p_i.pos_d = u_d$ ;  $p_i.vel_d = 0$ 
28:            $f = evaluate(p.pos_i)$ 
29:           if  $f < pbest_i.fit$  then
30:              $pbest_i.pos \leftarrow p_i.pos$ 
31:              $pbest_i.fit \leftarrow f$ 
32:           search  $q = pbest_k.pos - \min(pbest_k.fit)$ , for  $k = 1, 2, \dots, N$ .
33:            $q$  is the solution at iteration  $g$ .

```

---

### 3.5 Maximizing the Unpredictability in Multiscroll Chaotic Oscillators

Unpredictability is an important property of chaotic systems, since it means that the future events cannot be forecasted from the past events. Detecting the presence of chaos in a dynamical system is an important problem that is solved by measuring the largest Lyapunov exponent. The Lyapunov exponents give the most characteristic description of the presence of a deterministic nonperiodic flow. Therefore, Lyapunov exponents not only provide a qualitative characterization of dynamical behavior but also the exponent itself determines the measure of predictability. This means that a large Lyapunov exponent is equivalent to a high unpredictability [22].

**Table 3.1** Calculated positive Lyapunov exponent (L.E.) with coefficients values ( $a, b, c, d_1 = 0.7$ )

No. scrolls	L.E. with ( $a, b, c, d_1 = 0.7$ )
2	0.105422
3	0.138087
4	0.142087
5	0.134534
6	0.147785
7	0.148159

Hence, the estimation of the Lyapunov exponents as the useful dynamical classifier for deterministic chaotic systems is an important issue in nonlinear chaotic systems optimization.

The calculation of the Lyapunov exponents for the saturated nonlinear function series-based chaotic oscillator described by (3.1) can be performed by simply setting:  $a = b = c = d_1 = 0.7$ ,  $m = 0.1$  [2, 17].

In most reported approaches using saturated nonlinear function series-based chaotic oscillator [11, 17, 19, 28], the coefficients of the system are fixed to 0.7, but the positive Lyapunov exponent is relatively small, as shown in Table 3.1. Furthermore, in this section, we present the application and comparison of three computational intelligence algorithms: genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO), to maximize the positive Lyapunov exponent in a multiscroll chaotic oscillator based on saturated nonlinear function series with a modification of the standard settings of the coefficient values of the mathematical description and taking into account the correct distribution of the scrolls drawing the phase-space portrait.

For this chaotic oscillator, the optimization problem tries to find the values of the four coefficient variables  $a, b, c$ , and  $d_1$  in (3.1) that maximize the Lyapunov exponent. Those four coefficients can take values within the range  $[0.0, 1.0]$ . In our investigation, we use a resolution of 4 decimal digits for those variables, i.e., from 0.0001 to 1.0000 [2, 9].

The maximal Lyapunov exponent (MLE) was measured like it was described in Sect. 3.3. In addition, a very new procedure reported in [8] to measure the dispersions of the PSP coverage among all generated scrolls was included in the optimization loop. The procedure consists on counting the number of occurrences of the state trajectory in generating each scroll. Figure 3.1 shows the PWL function to generate 4 scrolls. The procedure for distributing the trajectories in the PSP basically counts how many times the state variable, e.g.,  $x$ , crosses the center of saturated levels (horizontal zones) of function  $f$  in Fig. 3.1 at the set of values  $x = \{-3, -1, 1, 3\}$ . The quantitative measure taken in [8] is the standard deviation among all crossing values at the end of the simulation time. In our case, we take the average among all crossing values to decide if it is a feasible or unfeasible solution. For a feasible solution, we take into account that 70 % of the average of the crosses is reached for each saturated region, e.g., for 4-scrolls crosses =  $\{224, 315, 301, 210\}$ ,

the average of these crosses is  $mean = 262.5$ , then, if in each case 183 crosses are reached, it is a feasible solution.

GA, DE, and PSO are stochastic algorithms in nature; hence, different results can be obtained between different runs and such results may also depend on the parameter settings [3, 9, 27]. For instance, Table 3.2 shows the MLE obtained for 2- to 7-scroll chaotic oscillator by applying the three selected algorithms. Each algorithm was executed 30 times, so Table 3.2 shows the best value of MLE, the mean Lyapunov exponent value, and the standard deviation.

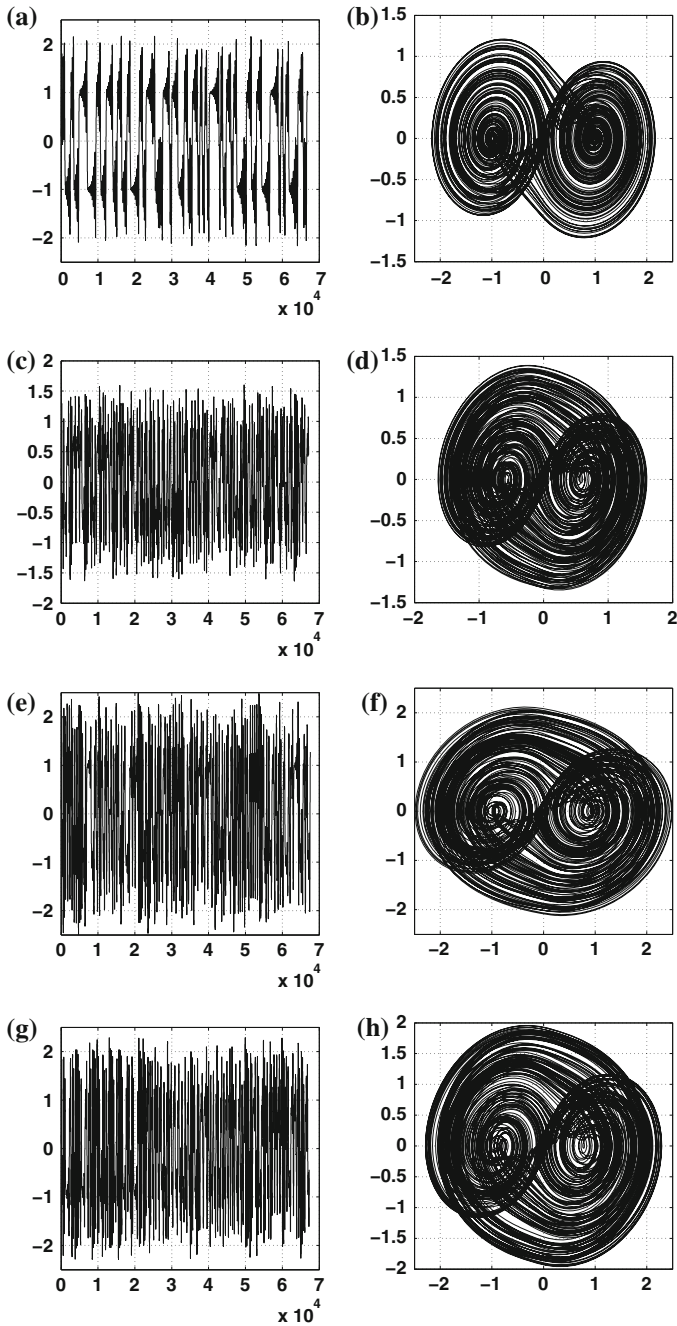
GA was executed with the crossover probability  $Pc = 0.9$  and mutation probability  $Pm = 0.1$ . A population of 80 individuals and 50 generations were used. DE was executed with the recombination constant  $R = 0.8$  and difference constant  $F = 0.6$ , a population of 40 individuals and 100 generations. Finally, PSO was executed with the inertia weight  $w = 0.721$  and the acceleration coefficients  $\varphi_1 = \varphi_2 = 1.193$ . A population of 20 particles and 200 generations were used. Therefore, a total of 4000 fitness evaluations were allowed for each algorithm.

Figures 3.3, 3.4, 3.5, 3.6, 3.7, and 3.8 show the transient evolution and phase-space portraits for the cases listed in Table 3.2 and those corresponding to the non-optimized coefficients in Table 3.1.

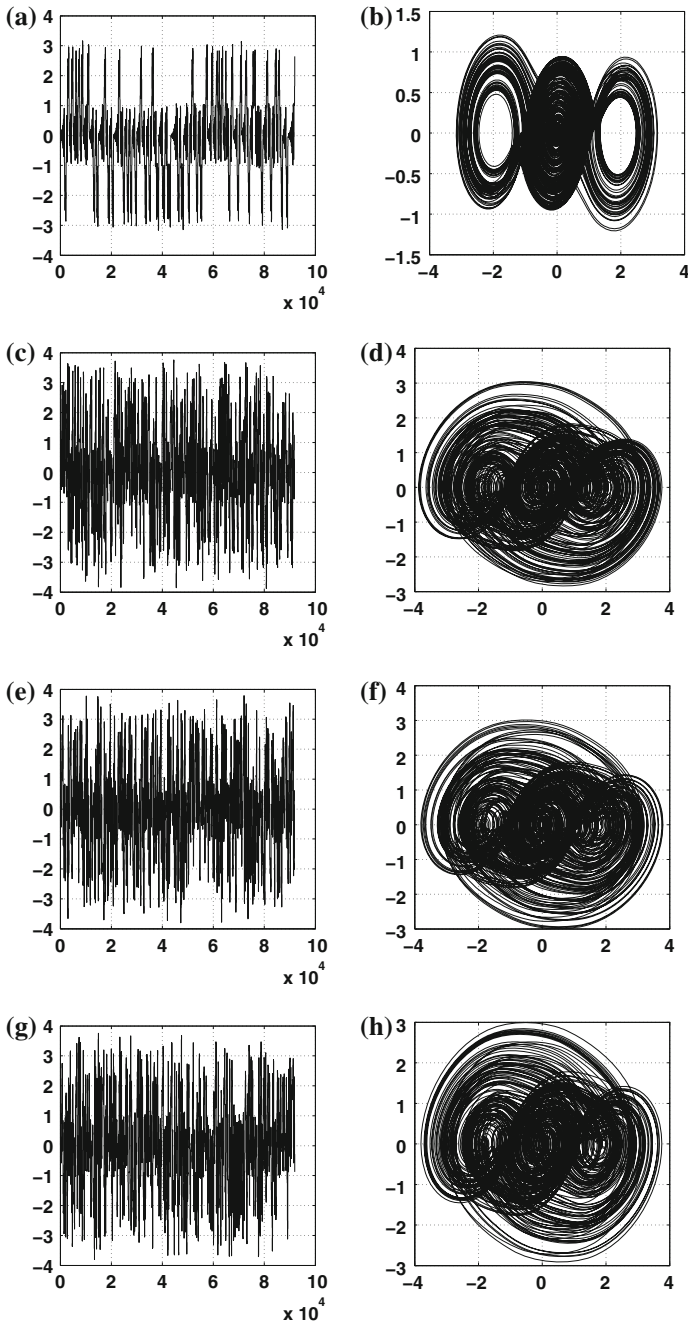
As it can be seen, the dynamic behavior of the chaotic system is more complex as the positive Lyapunov exponent increases, because it achieves greater unpredictability.

**Table 3.2** MLE values, mean values, standard deviation, and coefficient value of 30 executions of each heuristic against number of scrolls

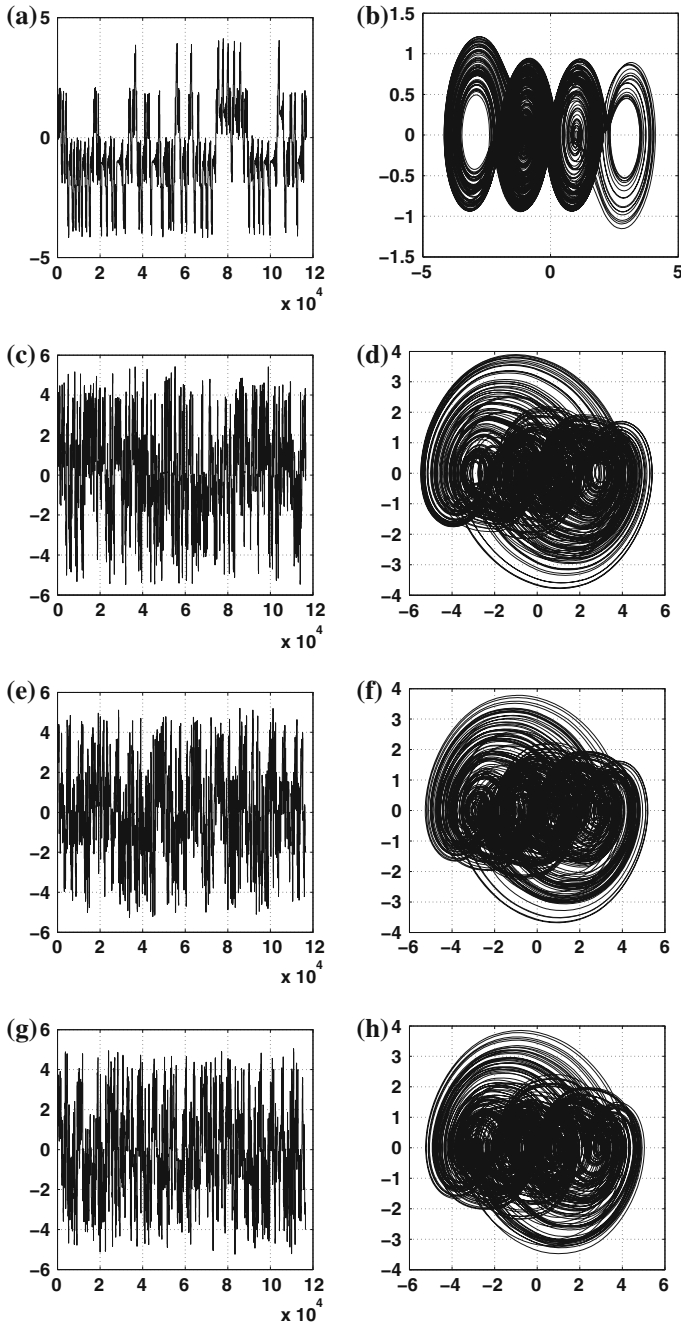
No. scrolls	Algorithm	M.L.E.	Mean	St. dev.	Constants— $[a, b, c, d_1]$
2	GA	0.221986	0.216023	0.005391	[0.9816, 0.8410, 0.4988, 0.6540]
2	DE	0.222767	0.218224	0.001765	[1.0000, 0.8284, 0.5321, 1.0000]
2	PSO	0.223114	0.219041	0.002024	[0.9970, 0.8469, 0.5098, 0.9221]
3	GA	0.298260	0.283042	0.011624	[0.9895, 0.7774, 0.3560, 1.0000]
3	DE	0.297813	0.290483	0.002884	[1.0000, 0.7782, 0.3416, 1.0000]
3	PSO	0.301033	0.294377	0.003385	[1.0000, 0.7724, 0.3618, 0.9927]
4	GA	0.303209	0.289411	0.014313	[0.9367, 0.6894, 0.3204, 0.9896]
4	DE	0.310734	0.300321	0.006029	[0.9399, 0.7037, 0.2854, 0.9660]
4	PSO	0.315349	0.306306	0.004998	[0.9607, 0.7028, 0.2728, 0.9880]
5	GA	0.296158	0.281553	0.012683	[0.9810, 0.8134, 0.2931, 1.0000]
5	DE	0.321793	0.302033	0.009817	[0.9770, 0.6622, 0.2180, 1.0000]
5	PSO	0.322885	0.309523	0.007469	[0.9497, 0.6494, 0.2749, 0.9966]
6	GA	0.313739	0.298833	0.008199	[0.9520, 0.5422, 0.2819, 1.0000]
6	DE	0.323515	0.307036	0.006663	[0.9167, 0.5410, 0.2467, 0.9521]
6	PSO	0.324055	0.310436	0.009127	[0.9502, 0.5745, 0.2395, 0.9916]
7	GA	0.322424	0.304251	0.016513	[0.9815, 0.7355, 0.1961, 1.0000]
7	DE	0.323100	0.307249	0.009793	[0.9692, 0.5269, 0.2312, 1.0000]
7	PSO	0.332127	0.320217	0.009676	[0.9391, 0.5217, 0.2172, 0.9699]



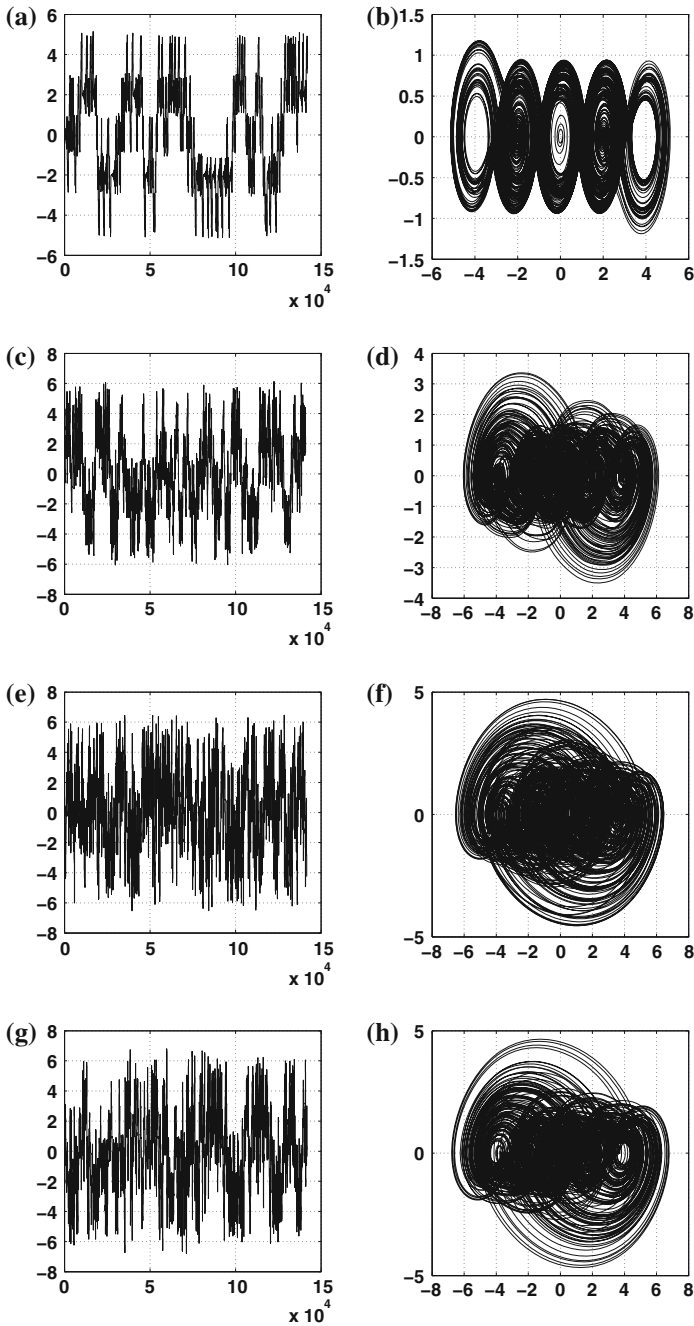
**Fig. 3.3** Diagrams of the cases listed in Table 3.2 for a 2-scroll chaotic oscillator: **a** Non-optimized  $x_1$  value versus time, **b** non-optimized  $x_1$  versus  $x_2$  values, **c** GA-optimized  $x_1$  value versus time, **d** GA-optimized  $x_1$  versus  $x_2$  values, **e** DE-optimized  $x_1$  value versus time, **f** DE-optimized  $x_1$  versus  $x_2$  values, **g** PSO-optimized  $x_1$  value versus time, **h** PSO-optimized  $x_1$  versus  $x_2$  values



**Fig. 3.4** Diagrams of the cases listed in Table 3.2 for a 3-scroll chaotic oscillator: **a** Non-optimized  $x_1$  value versus time, **b** non-optimized  $x_1$  versus  $x_2$  values, **c** GA-optimized  $x_1$  value versus time, **d** GA-optimized  $x_1$  versus  $x_2$  values, **e** DE-optimized  $x_1$  value versus time, **f** DE-optimized  $x_1$  versus  $x_2$  values, **g** PSO-optimized  $x_1$  value versus time, **h** PSO-optimized  $x_1$  versus  $x_2$  values

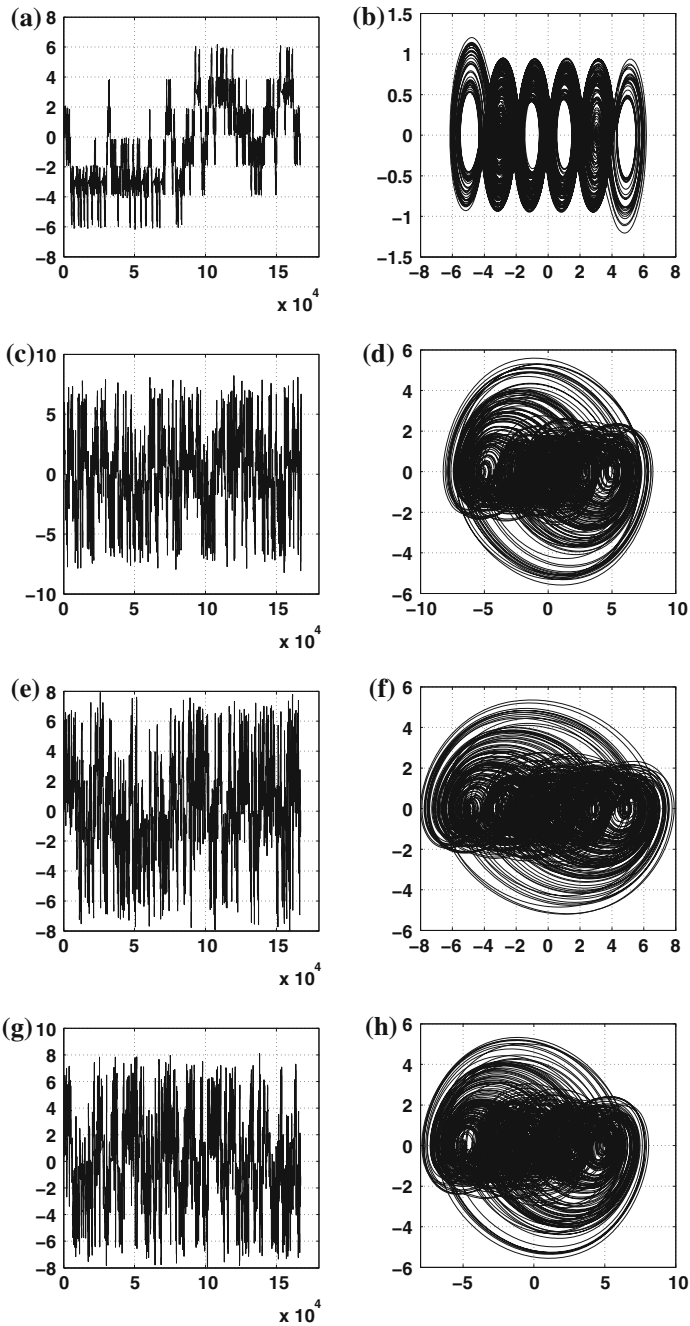


**Fig. 3.5** Diagrams of the cases listed in Table 3.2 for a 4-scroll chaotic oscillator: **a** Non-optimized  $x_1$  value versus time, **b** non-optimized  $x_1$  versus  $x_2$  values, **c** GA-optimized  $x_1$  value versus time, **d** GA-optimized  $x_1$  versus  $x_2$  values, **e** DE-optimized  $x_1$  value versus time, **f** DE-optimized  $x_1$  versus  $x_2$  values, **g** PSO-optimized  $x_1$  value versus time, **h** PSO-optimized  $x_1$  versus  $x_2$  values

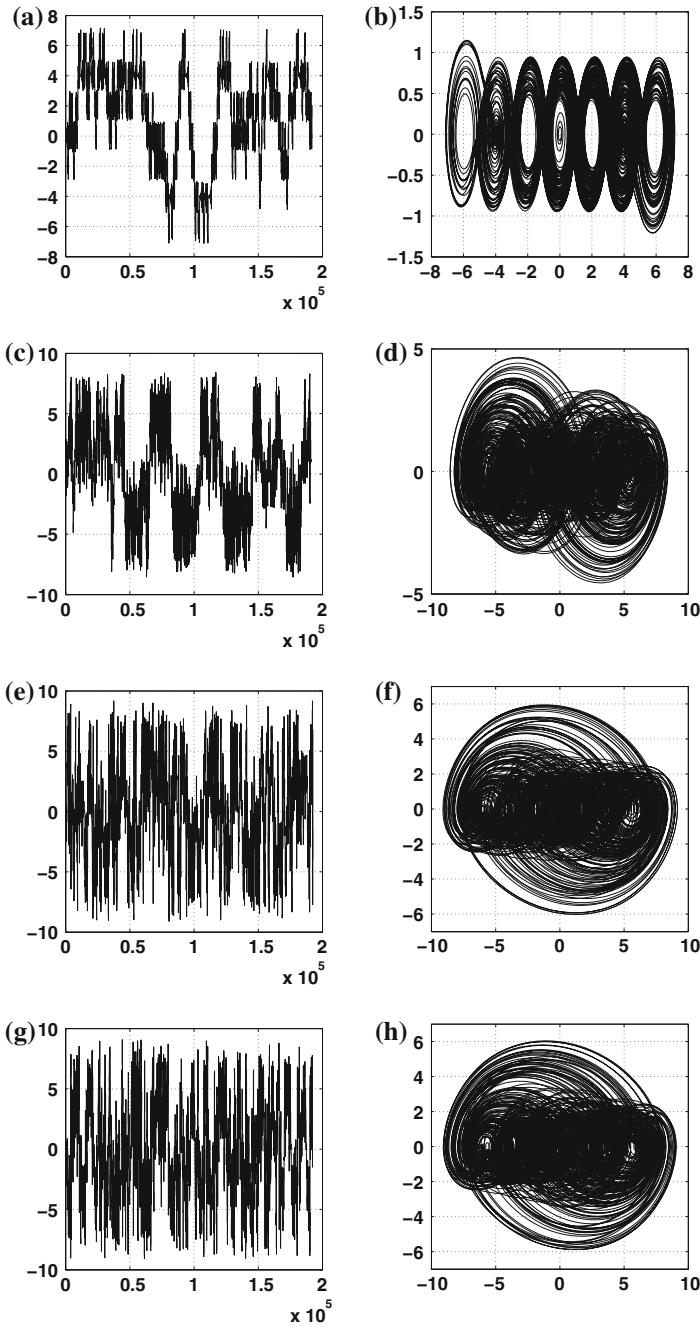


**Fig. 3.6** Diagrams of the cases listed in Table 3.2 for a 5-scroll chaotic oscillator: **a** Non-optimized  $x_1$  value versus time, **b** non-optimized  $x_1$  versus  $x_2$  values, **c** GA-optimized  $x_1$  value versus time, **d** GA-optimized  $x_1$  versus  $x_2$  values, **e** DE-optimized  $x_1$  value versus time, **f** DE-optimized  $x_1$  versus  $x_2$  values, **g** PSO-optimized  $x_1$  value versus time, **h** PSO-optimized  $x_1$  versus  $x_2$  values





**Fig. 3.7** Diagrams of the cases listed in Table 3.2 for a 6-scroll chaotic oscillator: **a** Non-optimized  $x_1$  value versus time, **b** non-optimized  $x_1$  versus  $x_2$  values, **c** GA-optimized  $x_1$  value versus time, **d** GA-optimized  $x_1$  versus  $x_2$  values, **e** DE-optimized  $x_1$  value versus time, **f** DE-optimized  $x_1$  versus  $x_2$  values, **g** PSO-optimized  $x_1$  value versus time, **h** PSO-optimized  $x_1$  versus  $x_2$  values



**Fig. 3.8** Diagrams of the cases listed in Table 3.2 for a 7-scroll chaotic oscillator: **a** Non-optimized  $x_1$  value versus time, **b** non-optimized  $x_1$  versus  $x_2$  values, **c** GA-optimized  $x_1$  value versus time, **d** GA-optimized  $x_1$  versus  $x_2$  values, **e** DE-optimized  $x_1$  value versus time, **f** DE-optimized  $x_1$  versus  $x_2$  values, **g** PSO-optimized  $x_1$  value versus time, **h** PSO-optimized  $x_1$  versus  $x_2$  values

The Runge–Kutta method of fourth order [20] was used to solve (3.1) and also to calculate the Lyapunov exponents. It was coded in C programming language, while the GA, DE, and PSO algorithms were coded in MATLAB. The integration step was selected as  $t_{\text{step}} = \frac{T_0}{50}$ . For this type of oscillator, the simulation was executed first for 400 s and then for another  $500 * s$  ( $s$  = number of scrolls) seconds, where the Lyapunov exponents were measured. Also, the initial condition was set for all the simulations to  $x_0 = [0.1, 0.0, 0.0]^T$ .

### 3.6 Conclusions

The optimization problem for the saturated nonlinear function series-based chaotic oscillator is encoded to search for the feasible values of variables  $a, b, c$ , and  $d_1$  to design a more robust and unpredictable chaotic oscillator. The search space in our problem can be calculated from the size of a variable: each coefficient in (3.1) has one significant digit that can be 0 or 1 and four decimal places can have values in  $\{0, 9\}$  (each one). Then, the number of possible solutions is  $10 \times 10 \times 10 \times 10 = 10^4$ . For the whole problem, the search space justifies the use of metaheuristics to solve the problem of maximizing the Lyapunov exponent. As shown in Table 3.2, DE and PSO algorithms produced a slightly greater value for the positive Lyapunov exponent and a smaller standard deviation than the GA algorithm. Both algorithms, DE and PSO, provide similar results in this application to maximize the positive Lyapunov exponent. Particle swarm optimization (PSO) tends to find higher values for the maximum Lyapunov exponent with a smaller population. It is noteworthy that in the original implementation [2, 9], the Lyapunov exponent values are larger than those obtained in this work and the scrolls will not appear uniformly in the phase-space portrait. In this work, the number of feasible solutions has been limited with the method described in Sect. 3.5. Therefore, there are lots of solutions that have been discarded if the distribution in phase-space portraits is not such that at least 70 % of the average of the crosses is reached for each saturated region. It is pretty clear from Figs. 3.3, 3.4, 3.5, 3.6, 3.7, and 3.8 that the optimized oscillator not only presents a better chaotic behavior, but also the distributions of the trajectories among the scrolls are well balanced. In this manner, one can select the appropriate one according to the application at hand, chaos is guaranteed and the oscillator will show all scrolls uniformly. This fact highlights the usefulness of applying computational intelligence techniques to maximize unpredictability in multiscroll chaotic oscillators.

**Acknowledgments** The first author wants to thank CONACyT-Mexico for the scholarship 331697. This work has been partially supported by CONACyT-Mexico under grant 131839-Y, in part by the TEC2013-45638-C3-3-R, funded by the Spanish Ministry of Economy and Competitiveness and ERDF, by the P12-TIC-1481 project, funded by Junta de Andalucía, and by CSIC project PIE 201350E058.

## References

1. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput. Surv. (CSUR)* **35**(3), 268–308 (2003)
2. Carbajal-Gómez, V.H., Tlelo-Cuautle, E., Fernández, F.V.: Optimizing the positive Lyapunov exponent in multi-scroll chaotic oscillators with differential evolution algorithm. *Appl. Math. Comput.* **219**(15), 8163–8168 (2013)
3. Carbajal-Gómez, V.H., Tlelo-Cuautle, E., Fernández, F.V., de la Fraga, L.G., Sánchez-López, C.: Maximizing Lyapunov exponents in a chaotic oscillator by applying differential evolution. *Int. J. Nonlin. Sci. Numer. Simul.* **15**(1), 11–17 (2014)
4. Carbajal-Gómez, V.H., Tlelo-Cuautle, E., Trejo-Guerra, R., Muñoz-Pacheco, J.M.: Simulating the synchronization of multi-scroll chaotic oscillators. In: 2013 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1773–1776. IEEE (2013)
5. Carbajal-Gómez, V.H., Tlelo-Cuautle, E., Trejo-Guerra, R., Sánchez-López, C., Muñoz-Pacheco, J.M.: Experimental synchronization of multiscroll chaotic attractors using current-feedback operational amplifiers. *Nonlinear Sci. Lett. B: Chaos Fractal Synchronization* **1**(1), 37–42 (2011)
6. Clerc, M.: From theory to practice in particle swarm optimization. In: *Handbook of Swarm Intelligence*, pp. 3–36. Springer, Berlin (2010)
7. Clerc, M.: Confinements and biases in particle swarm optimisation. Technical report. <https://hal.archives-ouvertes.fr/hal-00122799>. (2006)
8. de la Fraga, L.G., Tlelo-Cuautle, E.: Optimizing the maximum Lyapunov exponent and phase space portraits in multi-scroll chaotic oscillators. *Nonlinear Dyn.* **76**(2), 1503–1515 (2014)
9. de la Fraga, L.G., Tlelo-Cuautle, E., Carbajal-Gómez, V.H., Muñoz-Pacheco, J.M.: On maximizing positive Lyapunov exponents in a chaotic oscillator with heuristics. *Revista mexicana de física* **58**(3), 274–281 (2012)
10. Dieci, L.: Jacobian free computation of Lyapunov exponents. *J. Dyn. Diff. Equat.* **14**(3), 697–717 (2002)
11. Duarte-Villaseñor, M.A., Carbajal-Gómez, V.H., Tlelo-Cuautle, E.: Design of current-feedback operational amplifiers and their application to chaos-based secure communications. In: *Analog Circuits: Applications, Design and Performance*. NOVA Science Publishers (2012)
12. Glover, F., Kochenberger, G.A. (Eds.). *Handbook of Metaheuristics*. Springer, Berlin (2003)
13. Goldberg D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co, Reading, MA (1989)
14. Golub, G.H., Van Loan, C.F.: *Matrix computations (Johns Hopkins studies in mathematical sciences)*. (1996)
15. Holland, J.H.: *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, Ann Arbor, IL (1975)
16. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, vol. 4. Piscataway, NJ, pp. 1942–1948 (1995)
17. Lü, J., Chen, G.: Generating multiscroll chaotic attractors: theories, methods and applications. *Int. J. Bifurcat. Chaos* **16**(04), 775–858 (2006)
18. Lü, J., Yu, S., Leung, H., Chen, G.: Experimental verification of multidirectional multiscroll chaotic attractors. *IEEE Trans. Circuits Syst. I Regul. Papers* **53**(1), 149–165 (2006)
19. Muñoz-Pacheco, J.M., Tlelo-Cuautle, E.: *Electronic Design Automation of Multi-Scroll Chaos Generators*. Bentham Sciences Publishers (2010)
20. Parker, T.S., Chua, L.O.: *Practical Numerical Algorithms for Chaotic Systems*. Springer, Berlin (1989)
21. Price, K., Storn, R.M., Lampinen, J.A. *Differential Evolution: A Practical Approach to Global Optimization*. Springer, Berlin (2005)

22. Rugonyi, S., Bathe, K.J.: An evaluation of the Lyapunov characteristic exponent of chaotic continuous systems. *Int. J. Numer. Meth. Eng.* **56**(1), 145–163 (2003)
23. Sánchez-López, C., Fernández, F.V., Carbajal-Gómez, V.H., Tlelo-Cuautle, E., Mendoza-López, J.: Behavioral modeling of SNFS for synthesizing multi-scroll chaotic attractors. *Int. J. Nonlinear Sci. Numer. Simul.* **14**(7–8), 463–469 (2013)
24. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11**(4), 341–359 (1997)
25. Talbi, E.G.: *Metaheuristics: from Design to Implementation* (vol. 74). Wiley, London (2009)
26. Tlelo-Cuautle, E., Muñoz-Pacheco, J.M., Martínez-Carballido, J.: Frequency scaling simulation of Chua’s circuit by automatic determination and control of step-size. *Appl. Math. Comput.* **194**(2), 486–491 (2007)
27. Trejo-Guerra, R., Tlelo-Cuautle, E., Muñoz-Pacheco, J.M., Sánchez-López, C., Cruz-Hernández, C.: On the relation between the number of scrolls and the Lyapunov exponents in PWL-functions-based n-scroll chaotic oscillators. *Int. J. Nonlinear Sci. Numer. Simul.* **11**(11), 903–910 (2010)
28. Trejo-Guerra, R., Tlelo-Cuautle, E., Sánchez-López, C., Muñoz-Pacheco, J.M., Cruz-Hernández, C.: Realization of multiscroll chaotic attractors by using current-feedback operational amplifiers. *Revista mexicana de física* **56**(4), 268–274 (2010)
29. Yang, X.S.: *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley, London (2010)
30. Yang, C.J., Zhu, W.D., Ren, G.X.: Approximate and efficient calculation of dominant Lyapunov exponents of high-dimensional nonlinear dynamic systems. *Commun. Nonlinear Sci. Numer. Simul.* **18**(12), 3271–3277 (2013)

# Chapter 4

## Optimization and Co-simulation of an Implantable Telemetry System by Linking System Models to Nonlinear Circuits

Yao Li, Hao Zou, Yasser Moursy, Ramy Iskander, Robert Sobot  
and Marie-Minerve Louërat

**Abstract** This chapter presents a platform for modeling, design, optimization, and co-simulation of mixed-signal systems using the SystemC-AMS standard. The platform is based on a bottom-up design and top-down simulation methodologies. In the bottom-up design methodology, an optimizer is inserted to perform a knowledge-aware optimization loop. During the process, a PEANO trajectory is applied for global exploration and the Nelder–Mead simplex optimization method is applied for local refinement. The authors introduce an interface between system-level models and their circuit-level realizations in the proposed platform. Moreover, a transient simulation scheme is proposed to simulate nonlinear dynamic behavior of complete mixed-signal systems. The platform is used to design and verify a low-power CMOS voltage regulator for an implantable telemetry system.

### 4.1 Introduction

With the development of system on chip (SoC), the increasing complexity of mixed-signal systems makes their simulation and validation a demanding task. There is a trend toward hierarchical analog synthesis, automation, optimization, mixed-signal systems, etc. For most systems, the simulation needs to take into

---

Y. Li (✉) · H. Zou · Y. Moursy · R. Iskander · M.-M. Louërat  
Université Pierre et Marie Curie, Paris, France  
e-mail: yao.li@lip6.fr

M.-M. Louërat  
e-mail: Marie-Minerve.Louerat@lip6.fr

R. Sobot  
The University of Western Ontario, London, ON, Canada  
e-mail: Rsobot@uwo.ca

R. Sobot  
ENSA/ETIS, University of Cergy-Pontoise, Cergy-Pontoise, France

account both system and circuit levels, and the challenge is to create a co-simulation environment that allows synchronization and interaction between the two levels. Recently, the Accellera Systems Initiative releases an open source SystemC-AMS [1, 2]. As an extension to the SystemC [3], SystemC-AMS provides an extended set of capabilities for system-level mixed-signal modeling.

Many existing co-simulation approaches are based on SystemC, SystemC-AMS, or SPICE. In [4], co-simulation-refined models with timed data flow (TDF) paradigm of SystemC-AMS are presented. SystemC-AMS acts as master controlling VHDL testbench. In [5], the proposed solution relies on the integration between an instruction set simulator (ISS) and the SystemC simulation kernel to analyze the performances of embedded systems. In [6], it addresses a method for simulator coupling allowing a transient time simulation of SPICE and the mixed-signal language VHDL-AMS within one simulation process. Another attempt to achieve analog and mixed-signal simulation using loose coupling between SystemC and SPICE is presented in [7]. Nevertheless, all of these attempts lack a clear implementation to establish a link between system-level description and circuit-level realization.

This chapter presents a novel co-simulation framework used for modeling, design, and verification of mixed-signal systems based on knowledge-aware optimization engine. The complete system can be described using only the AMS extension of SystemC with some parts described in SPICE netlists. With this method, we can verify the impact of a circuit block (transistor netlist) on the system level. At the same time, the circuit-level non-idealities propagate upward and affect the system-level ideal behavior. In this co-simulation environment, the SystemC-AMS simulation and the circuit SPICE simulation engines are synchronized in order to perform a nonlinear time-domain analysis and to exchange data at the end of each time step.

Moreover, the optimization engine is used to perform an automatic sizing and biasing of the circuit level. It is a fast design space exploration of analog firm intellectual properties (IP). The main contribution is to propose a knowledge-aware optimization approach, instead of knowledge-based synthesis, which assumed that performance equations are provided by the designer for the underlying topology. We replace the performance equations by traditional SPICE-like netlists that are much easier to provide. Besides, the new optimization algorithm is combined with the hierarchical sizing and biasing methodology [8].

In summary, the advantages of the system-level to circuit-level co-simulation and optimization approach can be summarized as follows:

1. Proposing a very fast sizing and biasing engine to implement the analog IPs.
2. Achieving an automation sizing and biasing based on circuit performances.
3. Presenting a transient simulation scheme to allow the simulation of system-level non-conservative ideal models along with conservative non-ideal circuit-level netlists.

4. Basing only on the C/C++ language, our approach can be used both in high-level modeling (SystemC-AMS) and low-level design (SPICE, optimization engine).

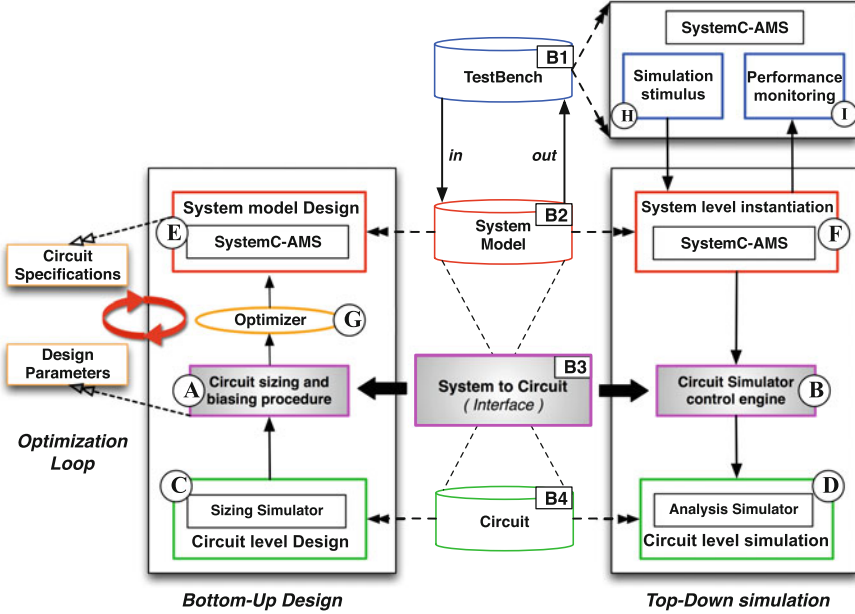
The chapter is organized as follows. Section 4.2 describes the co-simulation and optimization platform architecture by introducing the AMS extensions of SystemC and the hierarchical sizing and biasing procedure that are part of the platform. Section 4.3 gives a detailed explanation of the optimization engine. The implantable telemetry system is selected as the case study and shown in Sect. 4.4. The simulation cycle in co-simulation environment is introduced in Sect. 4.5. The simulation results of the circuit in different model are reported in Sect. 4.6. Section 4.7 concludes the demonstrated work.

## 4.2 Platform Architecture

Figure 4.1 represents the proposed platform architecture to link system models to nonlinear circuit. This platform is composed of a *bottom-up design* path as well as a *top-down simulation* path.

1. The bottom-up design path consists of the following:
  - A SPICE simulator © is used for sizing purpose.
  - The sizing simulator is controlled by the circuit sizing and biasing procedure Ⓐ.
  - An optimizer Ⓒ is called during the **end\_of\_elaboration** phase of a TDF module Ⓔ, defined by the SystemC standard.
  - The optimizer takes the circuit specifications as input parameters, calls the sizing and biasing procedure, and compares the circuit performances with specifications at each optimization iteration.
  - The whole design procedure provides an optimized, sized circuit to be used in the following top-down simulation.
2. The top-down simulation path consists of the following:
  - The *testbench* Ⓑ1 instantiates the SystemC-AMS models, generates the stimuli, and monitors the simulation results.
  - The instantiation of TDF models Ⓕ.
  - The circuit simulator control engine Ⓖ is called by the **processing** phase of a TDF module and applies the stimuli to the circuit netlist.
  - A SPICE simulator is used for analyzing the complete circuit netlist behavior.





**Fig. 4.1** Proposed modeling, design, optimization, and co-simulation platform architecture

As shown in Fig. 4.1, the system to circuit interface [B3] consists of two main parts: *the circuit sizing and biasing procedure* (A) and *the circuit simulator control engine* (B).

A complete system can be described using only the AMS extension of SystemC with some parts described in SPICE netlists. The proposed platform is capable to simulate the whole system with different levels of abstraction. Along with it, the circuit-level non-idealities propagate from upward and affect the system-level ideal behavior.

### 4.2.1 SystemC-AMS (Analog and Mixed-Signal System Design)

SystemC-AMS [1, 9] provides a framework for functional modeling [10], integration validation, and virtual prototyping [11] of *embedded analog and mixed-signal systems*. SystemC-AMS has three different models of computation: TDF, linear signal flow (LSF), and electrical linear networks (ELN).

Unlike the TDF modeling style, the LSF and ELN modeling styles can only be composed from their own linear primitives. Therefore, in the proposed approach, the TDF model of computation is selected. TDF is a discrete-time modeling style, which considers data as signals sampled in time. These signals are tagged at discrete

points in time and carry discrete or continuous values, such as voltages. Besides, TDF can be used with great efficiency to model complex non-conservative behaviors at system, functional, and macromodel level. Figure 4.2 shows the principle of the TDF modeling. The basic entities found in the TDF model of computation are as follows: the TDF modules, the TDF ports, and the TDF signals. The set of connected TDF modules form a directed graph, called a TDF cluster as defined below:

- TDF modules are the vertices of the graph.
- TDF signals correspond to its arcs.

Each TDF module involved in the cluster contains a specific C++ member function, named **processing()**, that computes a value at each time step.

If enough data samples are available at its input ports, depending on the involved port rates, the samples computed by a TDF module are written to the related output ports and describe continuous-time behaviors.

## 4.2.2 CHAMS Sizing and Biasing Engine

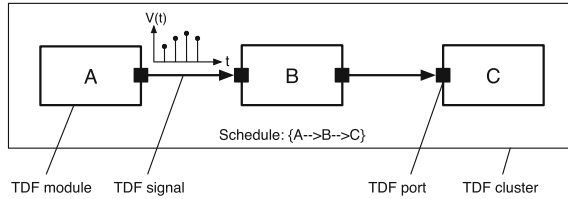
CHAMS [8, 12, 13] is a tool that provides assistance to the designer for the design of analog firm IP [14, 15]. It allows to generate the analog IP sizing and biasing procedure. It consists of the following three parts: *sizing and biasing operators*, *graph representation*, and *simulator encapsulation*.

### 4.2.2.1 Sizing and Biasing Operators and Graph Representation

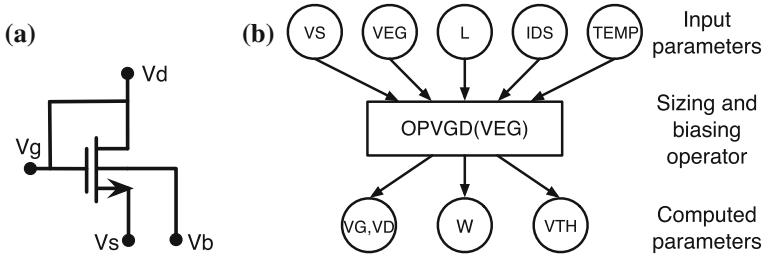
To size and bias a reference transistor, a bipartite directed acyclic graph (DAG) is associated with it. The bipartite graph [16] for the sizing and biasing of the diode-connected transistor using operator  $OPVGD(V_{EG})$  is shown in Fig. 4.3b. A set of input parameters are defined for the diode-connected transistor. The sizing and biasing operator  $OPVGD(V_{EG})$  is then called to compute the set of output parameters.

### 4.2.2.2 Simulator Encapsulation

Sizing and biasing operators use a specific simulator encapsulation that allows to interface with industrial design kits to ensure very accurate computed results. The simulator encapsulation is illustrated in Fig. 4.4. At the bottom is an electrical netlist that specifies the suitable technology and contains only 2 transistors: one PMOS and one NMOS, entirely sizable and biasable through simulator interactive commands. It is loaded by the electrical simulator launched in interactive mode.

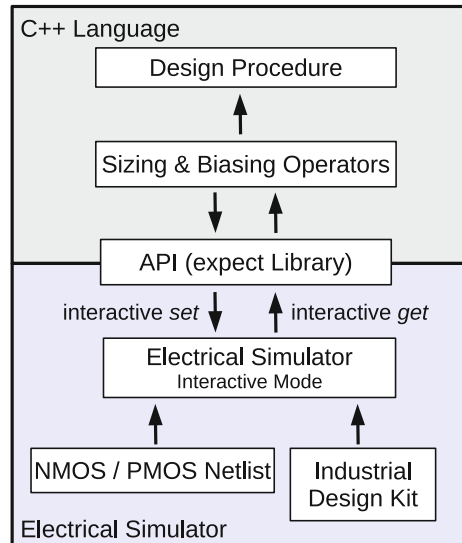


**Fig. 4.2** A basic TDF model with 3 TDF modules and 2 TDF signals



**Fig. 4.3** a NMOS reference transistor. b Graph representing the input parameters and output parameters of the operator OPVGD

**Fig. 4.4** CHAMS sizing engine: electrical simulator encapsulation within sizing and biasing operators



Three types of interactive commands are evaluated: *set*, *get*, and *run*. The first one allows to set all transistor known parameters (sizes and biases) inside the simulator. The second one enables to get all currents, voltages, and small-signal parameters computed by the simulator. After a set command, a simulation must be run using

run command, in order to compute the DC operating point of the transistor. An API is developed using *expect* library [17] to automate *set*, *get*, and *run* commands execution using simulator interactive mode. Sizing and biasing operators are optimized to minimize the number of calls to the simulator, which can reach several thousands during sizing.

### 4.3 Knowledge-Aware Simulation-Based Optimization Method

Simulation-based synthesis encapsulating a simulator within an optimization loop is presented in Fig. 4.5. Since the simulator is a verification tool, it starts with a set of sizes and biases (vector V2). First, it computes small-signal parameters (vector V3) by evaluating transistor models such as BSIM3v3 [18], BSIM4 [18], PSP [19], and EKV [20]. Second, linear and nonlinear performances (vector V4) are evaluated using a set of testbenches. We point that *performance evaluation* is performed by the simulator, and performances are then compared with the specifications that are specified by the designer.

Generally, the designer would like to use more meaningful design parameters (vector V1) to design analog circuits. The mapping to sizes and biases (vector V2)

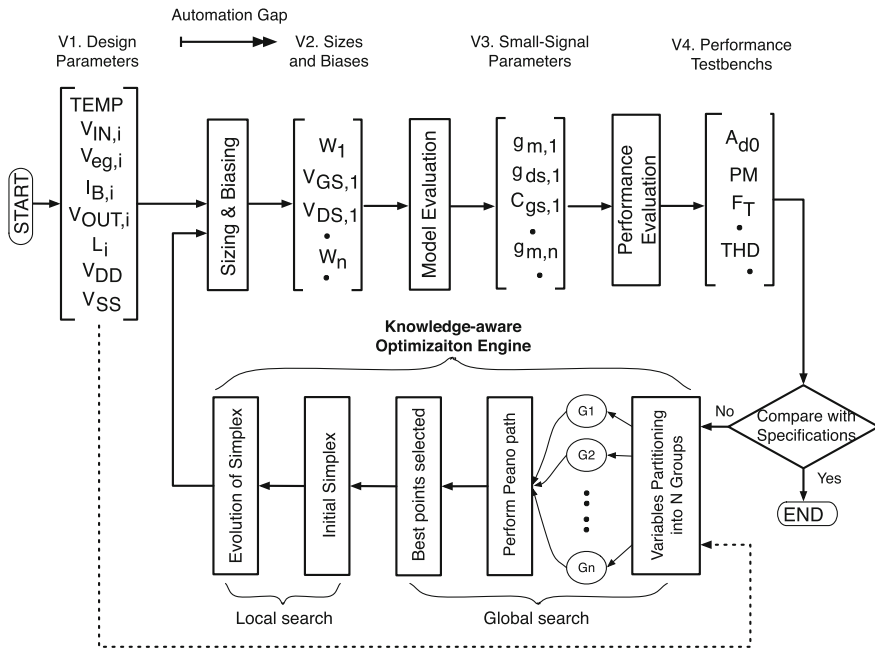


Fig. 4.5 Proposed loop for simulation-based synthesis with circuit optimization engine

becomes a laborious task that has to be repeated for each newly introduced circuit topology. This step depends mainly on the designer expertise and the complexity of circuit topology. Today, this step is not yet formalized; therefore, an *automation gap* is identified in the analog design flow, as illustrated in Fig. 4.5. This use of a formal representation favors the increase of analog design reuse, hence the reduction in design time. The automation gap is filled by generating design procedures using the *hierarchical sizing and biasing methodology*, already presented in the previous section.

Another major point is the performance evaluation. In general, performances are classified into 3 categories: *linear*, *weakly nonlinear*, and *strongly nonlinear*. Linear and weakly nonlinear performances may be easily modeled using mature symbolic analysis techniques [21, 22]. Strongly, nonlinear performances may be modeled using various techniques such as *model-order reduction* [23], *support vector machines* [24], and many others. In [8], we assumed that performance equations were mainly provided by the designer. Therefore, in this work, we propose to use the testbenches for circuit performance evaluation. Besides, we propose a very practical optimization method that is more adapted to the graph presentation as expected in [25].

The architecture of the optimizer is depicted as follows. The optimization variables comprise the set of design parameters chosen by the designer from vector  $V1$ . In order to break the curse of dimensionality, a partitioning scheme is selected where the  $n$  variables are partitioned into  $n/p$  groups of  $p$  variables each. Several variable groups are formed, and each group is globally optimized using a PEANO-like path exploration. During this global exploration, the best points are retained. Then, each point is used to start a local search by defining an initial simplex from this starting point and propagating this simplex until a convergence criterion is fulfilled. These schemes are explained in detail in the following sections.

### 4.3.1 Global Exploration: PEANO Trajectory

The trajectory used during the global search to compute the objective function was invented by the Italian mathematician PEANO [26] to establish a one-to-one correspondence between the number of points on a straight line and the number of points inside a square. This piecewise linear trajectory changes only 1 variable per step, helping optimization engine to converge faster since each point is taken as a prediction for the next one, based on the following Taylor expansion:

$$f(x_{1,next}, x_2, \dots) = f(x_{1,prev}, x_2, \dots) + \frac{\delta f}{\delta x_1} \cdot (x_{1,next} - x_{1,prev}) \quad (4.1)$$

**Fig. 4.6** PEANO trajectory for only 3 variables (X, Y, Z)

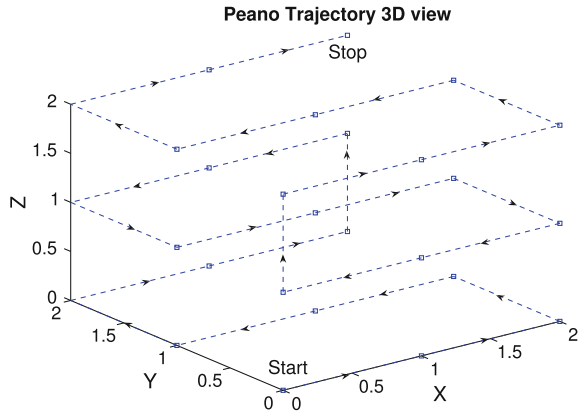


Figure 4.6 visualizes a PEANO trajectory for 3 variables (X, Y, Z). It is clear from the figure that moving on the PEANO path makes only 1 step change in 1 variable at a time.

### 4.3.2 Global Exploration: $p$ Variable Partitioning of an $n$ -Dimensional Design Space ( $p \ll n$ )

In order to break the *Curse of dimensionality* described by Richard Bellman in [27], a partitioning scheme for the  $n$ -dimensional space is proposed as follows: If we have  $n$  variables' optimization problem, we are interested in calculating the objective function at  $N$  points of a PEANO trajectory for each variable. In this case, the number of objective function evaluations  $N_{OBJ1}$  without partitioning is as follows:

$$N_{OBJ1}(\text{PEANO}, N, n) = N^n \tag{4.2}$$

Let us assume we make a partitioning by dividing randomly the  $n$  variables into  $n/p$  groups of  $p$  variables each. We repeat the partitioning process until a score of  $M$  is obtained for each variable.  $M$  is defined as the total number of times a given variable appears in all groups. In this case, the number of objective function evaluation  $N_{OBJ2}$  with partitioning is

$$N_{OBJ2}(\text{PEANO}, N, M, n, p) = M \cdot \frac{n}{p} \cdot N^p \tag{4.3}$$

Equations 4.2 and 4.3 indicate that the number of function evaluation provided by the  $p$  variable splitting is greatly reduced.

### 4.3.3 Local Exploration: Nelder–Mead Simplex

The Nelder–Mead simplex algorithm is the most widely used direct search method for solving the unconstrained optimization problem.

$$\min f(x) \quad (4.4)$$

where  $f(x)$  is called the objective function. A simplex is a geometric figure in  $n$  dimensions that is the convex hull of  $n + 1$  vertices. We denote a simplex with vertices  $x_1, x_2, \dots, x_{n+1}$ . The vertices satisfy the following relation:

$$f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1}) \quad (4.5)$$

where  $x_1$  refers as the best vertex, and  $x_{n+1}$  refers as the worst vertex. We eliminate the worst point of the simplex by using the four possible operations: *reflection*, *expansion*, *contraction*, and *shrink*, which are well defined in [28, 29].

The purpose of the global search is to extract lowest possible value points of the objective function to start the simplex search in a better area of interest. An initial simplex [30] placed symmetrically over these variables is an intuitive and reasonable choice.

### 4.3.4 The Cost Function

The objective function measures the deviation of the current solution with respect to objectives to minimize and constraints to meet. In our proposed formulation, the objective function is not a weighted function. It is the sum of two 2 types of contributions: *hard constraints* and *soft constraints*. A hard constraint must be satisfied to produce a feasible solution. A soft constraint has no guarantee to be satisfied. It may be minimized as best as possible.

A *hard constraint* is put off through a Heaviside function  $H(X)$  whenever it is exceeded, while a *soft constraint* is always active, as long as at least one *hard constraint* is exceeded.

We define the expression for a hard constraint as follows:

$$\text{Hard}_C = (1 - H(\text{spec}(i) - \text{spec}_{\text{lim}}(i))) \cdot \left( \frac{\text{spec}(i) - \text{spec}_{\text{lim}}(i)}{\text{spec}_{\text{lim}}(i)} \right)^2 \quad (4.6)$$

$H(X) = 1$  if  $X \geq 0$ , and  $H(X) = 0$  if  $X < 0$ .

We define the expression for a soft constraint as follows:

$$\text{Soft}_C = \alpha \cdot \left( \frac{\text{spec}(i) - \text{spec}_{\text{lim}}(i)}{\text{spec}_{\text{lim}}(i)} \right)^2$$

$$\text{with } \alpha = \frac{1}{n_{\text{hard}}} \sum_{i=1}^{n_{\text{hard}}} H(\text{spec}(i) - \text{spec}_{\text{lim}}(i)) \quad (4.7)$$

The general expression for the objective function  $F_{\text{obj}}$  is as follows:

$$F_{\text{obj}} = \alpha \cdot \sum_{i=1}^{n_{\text{soft}}} \left( \frac{\text{spec}(i) - \text{spec}_{\text{lim}}(i)}{\text{spec}_{\text{lim}}(i)} \right)^2$$

$$+ \sum_{i=1}^{n_{\text{hard}}} \left[ (1 - H(\text{spec}(i) - \text{spec}_{\text{lim}}(i))) \cdot \left( \frac{\text{spec}(i) - \text{spec}_{\text{lim}}(i)}{\text{spec}_{\text{lim}}(i)} \right)^2 \right] \quad (4.8)$$

**Note:** The spec represents the performance extracted from SPICE simulator, while  $\text{spec}_{\text{lim}}$  is the target specification.

$n_{\text{soft}}$  is the number of soft constraints, while  $n_{\text{hard}}$  is the number of hard constraints. The objective function is a summation of squared terms; therefore, its value is minimized when the specification spec reaches its target  $\text{spec}_{\text{lim}}$  at  $F_{\text{obj}} = 0$ .

#### 4.4 Case Study: Implantable Telemetry System

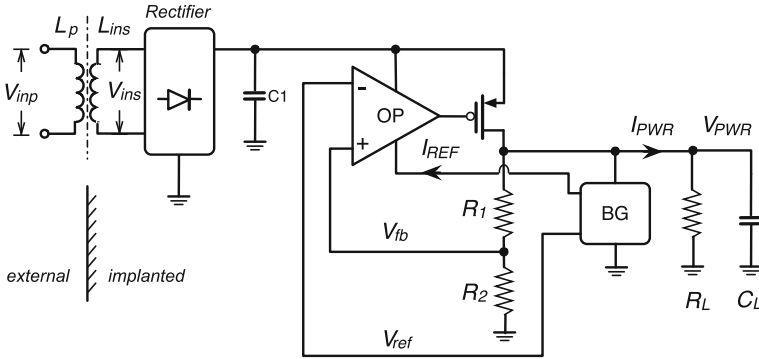
Recently, methodologies for energy harvesting received extensive attention in the research community and gained significant momentum. Especially, in the case of small animal subjects, rats and mice in particular, the coupling inductive of RF energy has become the primary method to transmit energy to the implantable telemetry system. However, the level of available internal energy varies by several orders of magnitude at the receiving side because of the subject's movement. Implication is that some form of AC/DC regulation is required for implantable telemetry systems [31].

The case study selected is an analogue IC of an implantable telemetry system. It is a RF-based CMOS voltage regulator for electromagnetic (EM) energy harvesting, which consists of a rectifier/charge pump, a folded-cascode amplifier, and a bandgap voltage reference sub-circuit, as shown in Fig. 4.7.

As the input RF power from the receiver is limited and constantly changing as the receiver moves, it requires the rectifier to be power efficient and the regulated supply to be stable when the given power supply changes. Consequently, it is important to design an efficient implantable voltage regulator that also consumes a minimal amount of energy for its own operation while providing continuous power to the load.

For a wireless power transmission system, the RF power to DC power conversion is realized in a rectifier. The generated steady DC voltage level depends on input RF signal.





**Fig. 4.7** Block diagram of energy-harvesting front-end circuit showing the inductors, rectifier/charge pump, and closed-loop regulation with folded-cascode amplifier (OP in the figure) and bandgap voltage reference (BG in the figure)

The regulation feedback loop is formed by the folded-cascode amplifier, the PMOS driver  $M_0$  and the voltage divider  $R_1$  and  $R_2$  network, which sets the ratio between  $V_{pwr}$  and  $V_{ref}$  voltages as

$$V_{pwr} = \left( 1 + \frac{R_1}{R_2} \right) \cdot V_{ref} \tag{4.9}$$

High DC gain in folded-cascode amplifier helps to suppress the difference between the  $V_{ref}$  and the feedback voltage.

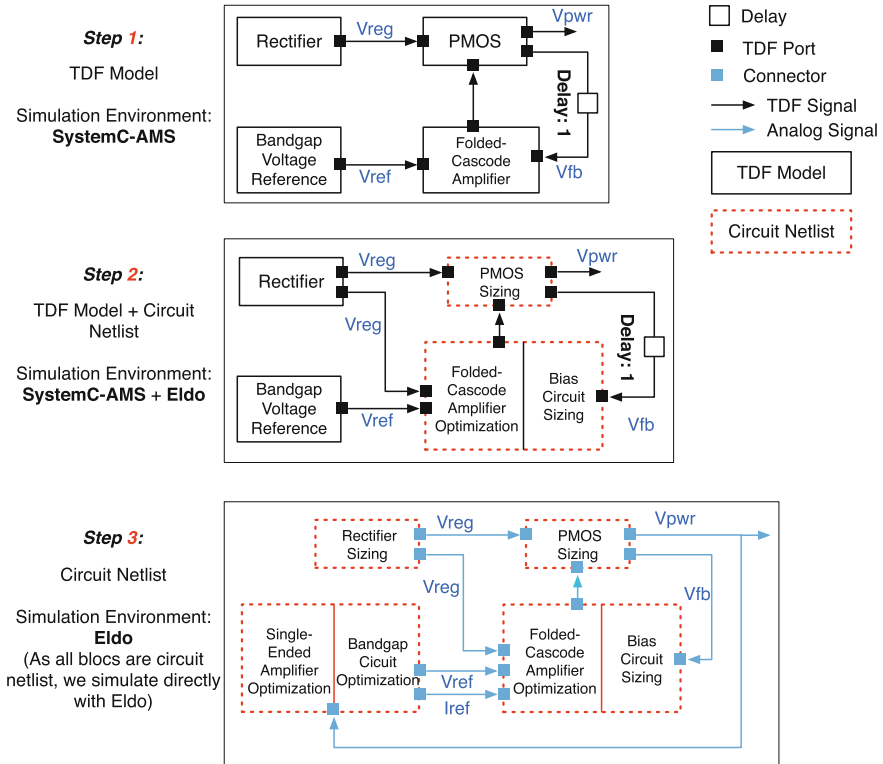
There are two main purposes to present this circuit:

1. Design: two blocks, bandgap voltage reference, and folded-cascode amplifier are extremely important in order to guarantee the feedback system to work as expected. Hence, we use the optimization engine to design and verify each blocks by meeting their specifications.
2. Simulation: we want to show our proposed platform can be used to co-simulate and verify a complete system that contains a feedback loop by propagating circuit non-idealities to system performances.

### 4.4.1 Design Process and Model Evolution

Figure 4.8 presents the design process and model evolution of the voltage regulator. It is a top-down structure and can be done in 3 steps:

- **Step 1:** Demonstrating the SystemC-AMS modeling environment, a set of TDF modules (**rectifier**, **PMOS**, **bandgap voltage reference**, **folded-cascode amplifier**) are organized to build the voltage regulator. Each module is



**Fig. 4.8** Synthesis flow of the voltage regulator, it is based on design concept, model evolution, optimization, combination, and co-simulation

integrated in a separate file. As shown in Fig. 4.8, the model contains a loop; therefore, a mandatory port delay assignment with delay value 1 (D: 1) has been performed on the output port of *PMOS*. This assignment allows the **folded-cascode amplifier + PMOS** loop to adjust the output signals  $V_{pwr}$  and keep it constant. The impact of the insertion of one delay can be neglected as a result of the sampling frequency is very high (500 MHz).

- **Step 2:** Establish a SystemC-AMS, Eldo co-simulation environment. Firstly, we optimize the folded-cascode amplifier by meeting their specifications. And then, we replace the ideal **folded-cascode amplifier** and **PMOS** model with circuit netlist and keep the **rectifier** and **bandgap voltage reference** as ideal model. At last, with the co-simulation platform, we simulate the whole system and propagate the nonlinearities of the folded-cascode amplifier at system level.
- **Step 3:** Optimize the bandgap voltage reference circuit and replace **Rectifier** and **Bandgap voltage reference** model with circuit netlist. Since all the blocs are in circuit netlist, the simulation can be done directly with Eldo.

### 4.4.2 Folded-Cascode Amplifier

The high DC gain of the operational amplifier can be achieved by using a single-stage folded-cascode structure. The diagram of folded-cascode amplifier is shown in Fig. 4.9. It contains two parts: folded-cascode amplifier and its bias circuit. Biasing voltages,  $V_2$ ,  $V_3$  and  $V_4$  need to be carefully calculated to ensure that the associated devices operate in the saturation region over the load variation. They are generated by the bias circuit which is associated with the left part of Fig. 4.9.

The sizing procedure of the whole folded-cascode amplifier circuit can be separated into two parts:

1. Firstly, we apply the optimizer engine to optimize the folded-cascode amplifier.
2. Secondly, with the desired biasing voltage ( $V_2$ ,  $V_3$  and  $V_4$ ), we size the bias circuit to meet these biasing voltages.

Instead of optimizing the whole circuit, we optimize the core part of the circuit and size the remain part using the sizing and biasing procedure without optimization. Such kind of optimization procedure can dramatically reduce the optimization complexity by decreasing the number of variables.

The sizing and biasing procedure of the folded-cascode amplifier is shown in Fig. 4.10 for a 130-nm process (sizing procedure of the bias circuit is shown in Fig. 4.11). It is a *bipartite graph* [13] that contains the designer’s knowledge to size and bias the amplifier. The folded-cascode amplifier is composed of five devices:  $D_1, D_2, D_3, D_4, D_5$  and a transistor  $M_b$ . The designer’s knowledge is represented by

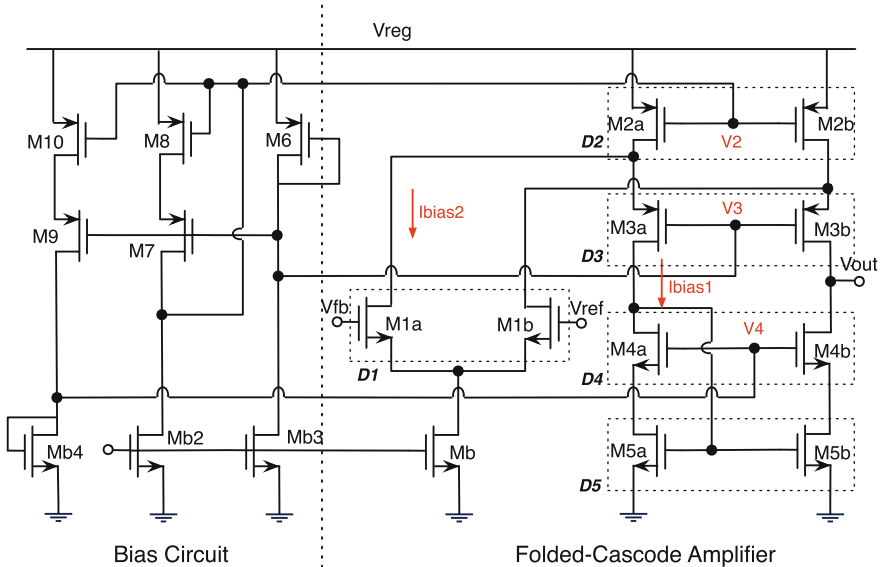
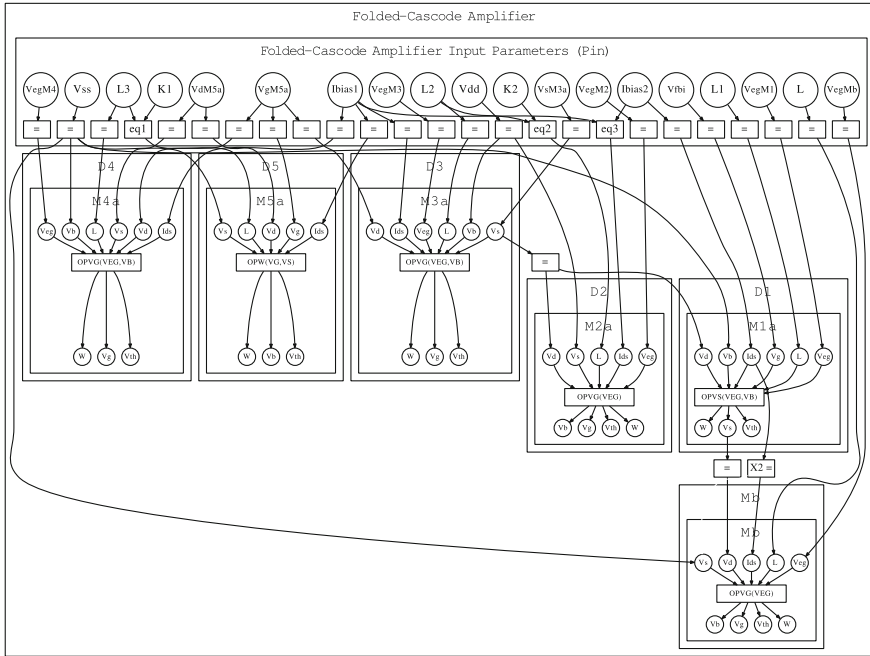


Fig. 4.9 Schematic diagram of the folded-cascode amplifier

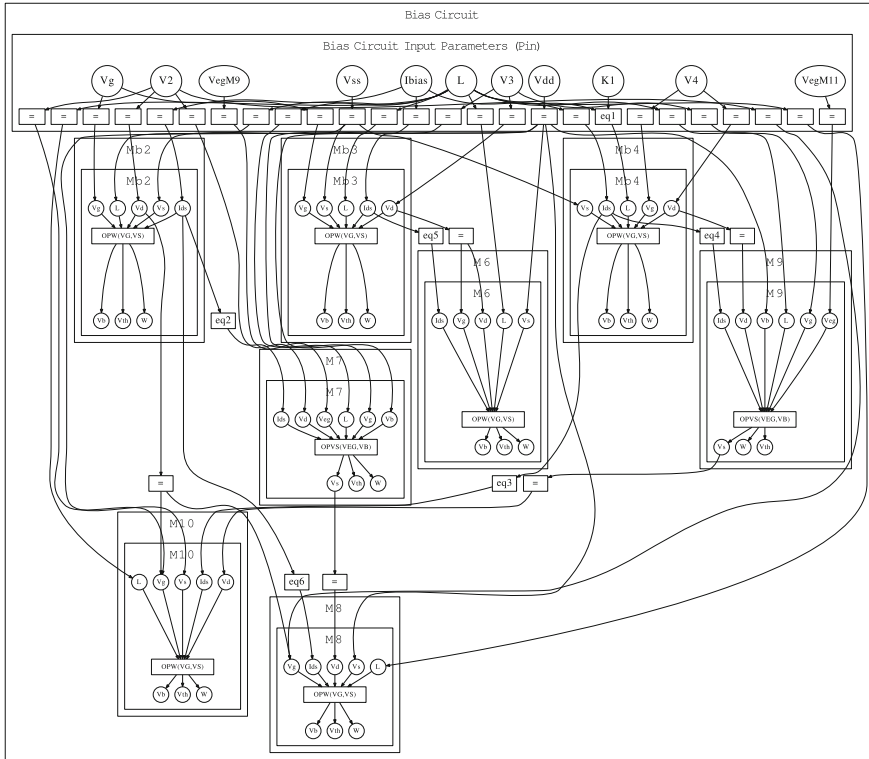


**Fig. 4.10** The bipartite graph (i.e., the design procedure) associated with the folded-cascode amplifier. Sizing and biasing operators are part of the bipartite graph

$P_{in}$  set of input parameters (at the top of the graph). Parameters in  $P_{in}$  (see in Tables 4.3 and 4.4, these present fixed variables and optimized variables, respectively) are spread in the graph and used by the sizing and biasing operators to compute unknown sizes and biases. Rectangle nodes named “eq” represent designer’s defined equations, and an example of equation is given with eq3:  $I_{Bias\_D2} = I_{Bias1} + I_{Bias2}$ , ( $I_{Bias1}$  and  $I_{Bias2}$  are given from input parameters, and the result  $I_{Bias\_D2}$  is passed to device  $D_2$ ). The resulting output parameters  $P_{out}$  are listed in Table 4.5. The bipartite graph is a sequence of sizing and biasing operators, and it is evaluated from top to bottom to get the sizes and biases of all transistors.

Figure 4.11 shows the bipartite graph of the bias circuit of the folded-cascode amplifier. The input parameters of the bias circuit are related to the output parameters of the folded-cascode amplifier. In other words, the whole design procedure can be seen as a hierarchical sizing and biasing method.

We want a high gain to avoid any discrepancy in the DC input voltage on positive and negative terminals of the amplifier. Table 4.1 gives the specifications to be met and displays the optimized performances. The global search boundaries for optimizing folded-cascode amplifier are shown in Table 4.2, and nine variables are optimized: 3 lengths ( $L_{M1a}$ ,  $L_{M3a}$ , and  $L_{M4a}$ ); 4 overdrive voltages ( $V_{EG,M4a}$ ,  $V_{EG,M3a}$ ,  $V_{EG,M2a}$ , and  $V_{EG,M1a}$ ); 2 currents ( $I_{BIAS1}$  and  $I_{BIAS2}$ ). The search boundary for each variable has been selected arbitrarily.



**Fig. 4.11** The bipartite graph (i.e., the design procedure) associated with the bias circuit of the folded-cascode amplifier

**Table 4.1** Specifications for folded-cascode amplifier circuit in 130-nm technology

Specification	Requirement	Constraint type	Performances
Gain	$\geq 75$ dB	Hard	75.2 dB
Unity-gain frequency	$\geq 1$ MHz	Hard	1.421 MHz
Phase margin	$\geq 80^\circ$	Hard	$87.9^\circ$
Power (10 k $\Omega$ load)	$\leq 10$ $\mu$ W	Soft	10.5 $\mu$ W

**Table 4.2** Global search boundaries for optimizing folded-cascode amplifier

Parameter	Boundaries values	Parameter	Boundaries values
$L_{M1a}$ ( $\mu$ m)	$0.5 \leq L_{M1a} \leq 3$	$L_{M3a}$ ( $\mu$ m)	$0.5 \leq L_{M3a} \leq 3$
$L_{M4a}$ ( $\mu$ m)	$0.5 \leq L_{M4a} \leq 3$	$I_{BIAS1}$ ( $\mu$ A)	$2.0 \leq I_{BIAS1} \leq 5.0$
$I_{BIAS2}$ ( $\mu$ A)	$2.0 \leq I_{BIAS2} \leq 5.0$	$V_{EG,M1a}$ (V)	$0.06 \leq V_{EG,M1a} \leq 0.15$
$V_{EG,M2a}$ (V)	$0.06 \leq V_{EG,M2a} \leq 0.15$	$V_{EG,M3a}$ (V)	$0.06 \leq V_{EG,M3a} \leq 0.15$
$V_{EG,M4a}$ (V)	$0.06 \leq V_{EG,M4a} \leq 0.15$		

**Table 4.3** Input parameters ( $P_{in}$ ) of the folded-cascode amplifier (fixed variables)

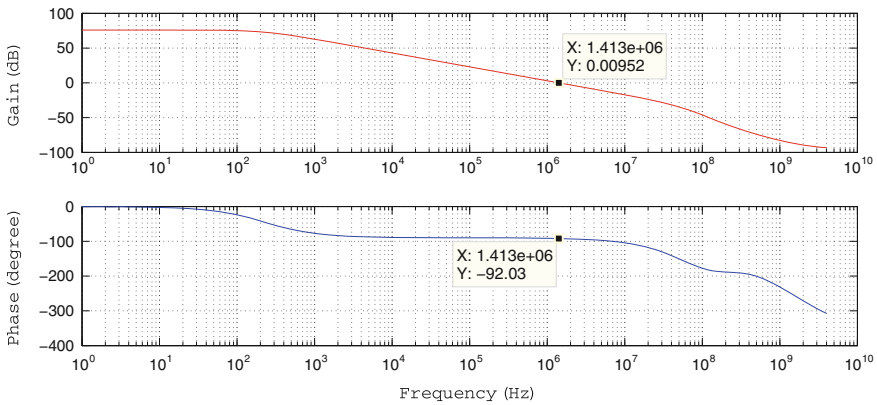
Parameter	Value	Parameter	Value	Parameter	Value
$V_{DD}$ (V)	1.2	$V_{D,M5a}$ (V)	0.3	$L_{M_b}$ ( $\mu\text{m}$ )	1.0
$V_{SS}$ (V)	0.0	$V_{D,M4a}$ (V)	0.6	$V_{EG,M_b}$ (V)	0.10
$V_{REF}$ (V)	0.585	$V_{S,M3a}$ (V)	0.9	$K_2 = L_{M2a}/L_2$	5
$K_1 = L_{M5a}/L_3$	5				

**Table 4.4** Input parameters ( $P_{in}$ ) of the folded-cascode amplifier (optimized variables)

Parameter	Value	Parameter	Value	Parameter	Value
$L_{M1a}$ ( $\mu\text{m}$ )	0.655	$L_{M3a}$ ( $\mu\text{m}$ )	0.595	$V_{EG,M2a}$ (V)	0.0766
$L_{M4a}$ ( $\mu\text{m}$ )	0.880	$I_{BIAS1}$ ( $\mu\text{A}$ )	2.071	$V_{EG,M3a}$ (V)	0.0695
$I_{BIAS2}$ ( $\mu\text{A}$ )	2.198	$V_{EG,M1a}$ (V)	0.0812	$V_{EG,M4a}$ (V)	0.0545

**Table 4.5** Computed width for the folded-cascode amplifier ( $P_{out}$ )

Transistor	Width	Transistor	Width	Transistor	Width
$W_{D1}$ ( $\mu\text{m}$ )	1.325	$W_{D4}$ ( $\mu\text{m}$ )	4.785	$W_{D3}$ ( $\mu\text{m}$ )	3.730
$W_{D2}$ ( $\mu\text{m}$ )	38.380	$W_{D5}$ ( $\mu\text{m}$ )	5.855	$W_{M_b}$ ( $\mu\text{m}$ )	1.405



**Fig. 4.12** Simulated gain and phase margin of the folded-cascode amplifier

Figure 4.12 illustrates the AC simulation results from optimized circuit. The DC gain is equal to 75.2 dB, phase margin is equal to 87.9°, the transition frequency is 1.421 MHz, and the power consumption is about 10.5  $\mu\text{W}$  with a 10 k $\Omega$  loaded resistor. The load capacitance is set to 50 pF.

### 4.4.3 Bandgap Voltage Reference Circuit

A key target for an integrated voltage reference is to provide adequate temperature stability and high rejection to power supply variations. These features are typically achieved by using a bandgap-based reference.

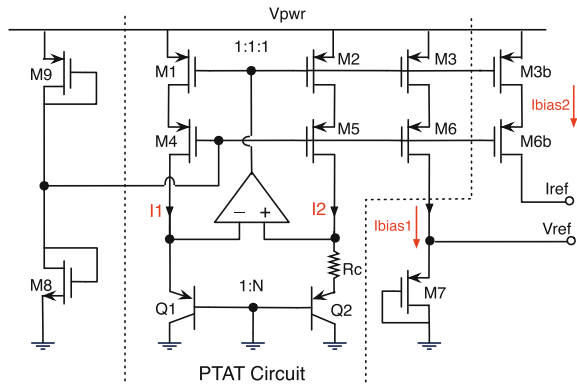
In this section, we will present a low-voltage low-power temperature-insensitive voltage reference. The schematic diagram of the circuit is presented in Fig. 4.13. To be more specific, an amplifier implements the weighted sum between a complementary to absolute temperature (CTAT) voltage (generated by means of a forward-biased diode) and a proportional to absolute temperature (PTAT) voltage. In the CMOS 0.13  $\mu\text{m}$  process, with this implementation, it is possible to work with power supply voltage as low as 1.0 V. The bandgap voltage reference circuit consists of a single-ended two-stage amplifier, which is detailed in the next sub-section. The optimization of the bandgap voltage reference circuit is done in 2 steps:

1. The first step consists in optimizing the single-ended two-stage amplifier by meeting some specifications.
2. The second step aims at optimizing the bandgap voltage reference circuit structure using the previously optimized amplifier, by sizing the remaining bandgap voltage reference circuit transistors.

#### 4.4.3.1 Single-Ended Two-Stage Amplifier

A single-ended two-stage amplifier is used inside the bandgap voltage reference circuit, as shown in Fig. 4.14. As the emitter–base voltage of Q1 varies from 0.5 to 0.8 V over the full temperature range, an NMOS input differential pair is used in input stage of the amplifier.

**Fig. 4.13** Schematic diagram of bandgap voltage/current reference circuit that consists of PTAT core,  $V_{\text{ref}}$  generator,  $I_{\text{ref}}$  generator, and a biasing voltage generator



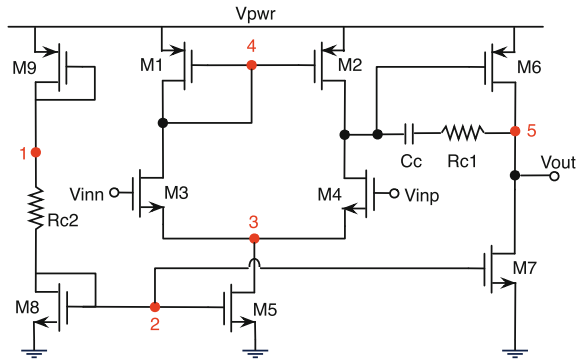


Fig. 4.14 Schematic diagram of the single-ended two-stage amplifier

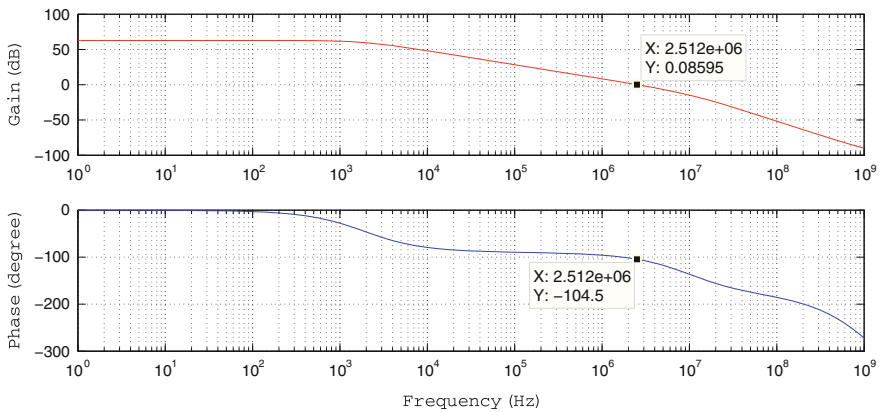


Fig. 4.15 Simulated gain and phase margin of the two-stage amplifier in the PTAT circuit

We use the same methodology presented in Sect. 4.4.2 to optimize this amplifier. With a 50-pF load capacitance, optimization results show that the DC gain is 67.8 dB, phase margin  $75.5^\circ$  with the transition frequency of 2.518 MHz. Figure 4.15 presents the AC simulation results.

#### 4.4.3.2 Temperature Independent of Bandgap Voltage Reference Circuit

In the analysis of bandgap voltage reference circuit, shown in Fig. 4.13, assuming for simplicity that  $(M_1-M_2-M_3)$ ,  $(M_4-M_5-M_6)$  are identical pairs, where  $I_1 = I_2$ , yielding the same behavior for  $I_{BIAS1}$ , we note that the transistor  $M_7$  works in the saturation region. The current  $I_{BIAS1}$  therefore equals



$$I_{\text{BIAS}1} = I_{\text{SD}} = \frac{1}{2} \mu_p C_{\text{ox}} \frac{W_7}{L_7} (V_{\text{SG}} - |V_{\text{TP}}|)^2 \quad (4.10)$$

we get

$$V_{\text{ref}} = V_{\text{SG}} = |V_{\text{TP}}| + \sqrt{\frac{2I_{\text{SD}}}{\mu_p C_{\text{ox}} \frac{W_7}{L_7}}} \quad (4.11)$$

Note:  $|V_{\text{TP}}|$  is the PMOS threshold voltage,  $\mu_p$  is the carrier mobility,  $C_{\text{ox}}$  is the unit gate oxide capacitance,  $I_{\text{SD}}$  is the bias current, and  $(W/L)$  is the gate width to length ratio. Therefore, a temperature-independent voltage/current reference is required.

In this equation, the  $I_{\text{SD}}$  and  $\mu_p$  are two parameters related to the temperature and hence:

$$\begin{aligned} \frac{\partial V_{\text{ref}}}{\partial T} &= \frac{\partial |V_{\text{TP}}|}{\partial T} + \frac{\partial \sqrt{\frac{2I_{\text{SD}}}{\mu_p C_{\text{ox}} \frac{W_7}{L_7}}}}{\partial T} \\ &= \frac{\partial |V_{\text{TP}}|}{\partial T} + \sqrt{\frac{1}{2 C_{\text{ox}} \mu_p I_{\text{SD}} \frac{W_7}{L_7}}} \cdot \frac{\partial I_{\text{SD}}}{\partial T} - \sqrt{\frac{I_{\text{SD}}}{2 C_{\text{ox}} \mu_p^3 \frac{W_7}{L_7}}} \cdot \frac{\partial \mu_p}{\partial T} \end{aligned} \quad (4.12)$$

For a bipolar device, we can write  $I_C = I_S \exp(V_{\text{BE}}/V_T)$ , where  $V_T = kT/q$ , thus:

$$\begin{aligned} I_2 &= \frac{\Delta V_{\text{EB}}}{R_C} = \frac{V_{\text{BE}2} - V_{\text{BE}1}}{R_C} \\ &= \frac{V_T \ln \frac{I_{C2}}{I_{S2}} - V_T \ln \frac{I_{C1}}{I_{S1}}}{R_C} = \frac{V_T \ln N}{R_C} = \frac{KT}{q R_C} \ln N \end{aligned} \quad (4.13)$$

Now, returning to Eq. 4.11 and including  $\partial I_{\text{SD}}/\partial T$ , we have

$$\frac{\partial V_{\text{ref}}}{\partial T} = \frac{\partial |V_{\text{TP}}|}{\partial T} + \frac{1}{g_{m7}} \frac{K \ln N}{q R_C} - \sqrt{\frac{I_{\text{SD}}}{2 C_{\text{ox}} \mu_p^3 \frac{W_7}{L_7}}} \cdot \frac{\partial \mu_p}{\partial T} \quad (4.14)$$

To get a temperature-independent voltage, it should have a positive temperature coefficient as well as a negative temperature coefficient. The above analysis helps to select the global search boundaries for optimizing bandgap voltage reference circuit, as shown in Table 4.6, and five variables are optimized:  $I_{\text{BIAS}1}$ ,  $N$ ,  $R_C$ ,  $L$ , and  $V_{M7, \text{VS}}$ .

We choose the specifications of the bandgap voltage reference circuit. Firstly, we expect the reference voltage is restricted to a very narrow range between 0.57 and 0.61 V. Secondly, the variation of  $V_{\text{ref}}$  with temperature (between 0 and 100 °C) in

**Table 4.6** Global search boundaries for optimizing bandgap voltage reference circuit

Parameter	Boundaries values	Parameter	Boundaries values
$L$ ( $\mu\text{m}$ )	$0.5 \leq L \leq 3$	$N$	$6 \leq N \leq 15$
$R_C$ ( $\text{k}\Omega$ )	$10 \leq R_C \leq 50$	$V_{M7,V_S}$ (V)	$0.5 \leq V_{M7,V_S} \leq 0.7$
$I_{BIAS}$ ( $\mu\text{A}$ )	$1 \leq I_{BIAS} \leq 4$		

**Table 4.7** Specifications for bandgap voltage reference circuit in 130-nm technology

Specification	Mode	Performances	Constraint type
$V_{\max}$ of $V_{\text{ref}}$	Typical	$\geq 0.57$ (V)	Hard
$V_{\min}$ of $V_{\text{ref}}$	Typical	$\leq 0.61$ (V)	Hard
$\Delta V_{\text{ref}}$	Typical	$\leq 0.0005$ (V)	Hard
$\Delta V_{\text{ref}}$	Corner	$\leq 0.0015$ (V)	Hard
Power (10 $\text{k}\Omega$ load)	Typical	$\leq 10$ ( $\mu\text{W}$ )	Soft

**Table 4.8** Input parameters ( $P_{\text{in}}$ ) of the bandgap voltage reference circuit (fixed variable)

Parameter	Value	Parameter	Value	Parameter	Value
$I_{BIAS2}$ ( $\mu\text{A}$ )	4.15	$V_{SS}$ (V)	0.0	$V_{M6b,V_S}$ (V)	0.46
$V_{M6,V_{EG}}$ (V)	-0.12	$V_{DD}$ (V)	1.0	$K = L_{M8,M9}/L$	5
$V_{M6b,V_{EG}}$ (V)	-0.12	$V_{M3,V_{EG}}$ (V)	-0.12		

**Table 4.9** Input parameters ( $P_{\text{in}}$ ) of the bandgap voltage reference circuit (optimized variable)

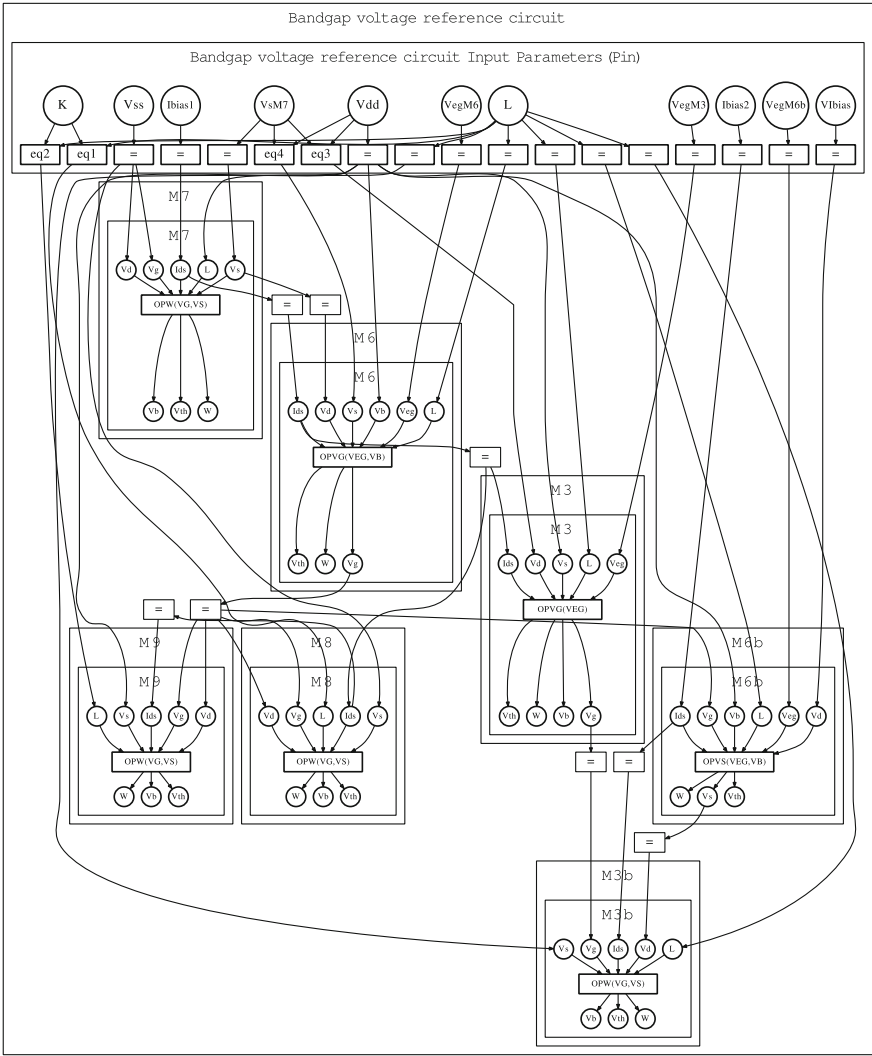
Parameter	Value	Parameter	Value	Parameter	Value
$L$ ( $\mu\text{m}$ )	0.325	$N$	9	$I_{BIAS1}$ ( $\mu\text{A}$ )	1.2198
$R_C$ ( $\Omega$ )	26,610	$V_{M7,V_S}$ (V)	0.5137		

typical mode and corner mode should be less than 0.5 and 1.5 mV, respectively. Thirdly, as we design a low-power circuit, the power dissipation should be less than 10 mW with a 10  $\text{k}\Omega$  load. Table 4.7 gives the all the specifications to be met.

The sizing and biasing procedure of the bandgap voltage reference circuit is presented in Fig. 4.16, note that the input parameters  $I_{BIAS2}$  and  $V_{M6b,V_S}$  are inherited from the sizing and biasing procedure of the folded-cascode amplifier, which are equal to  $I_{M_b,BIAS}$  and  $V_{M_b,V_G}$  respectively. The computed width values for the bandgap voltage reference circuit are listed in Table 4.10.

#### 4.4.3.3 Simulation Results of the Bandgap Voltage Reference Circuit

The optimization is performed using 3 SPICE netlists to simulate each of the corner cases for the 130-nm technology. In our bandgap voltage reference circuit, we have 3



**Fig. 4.16** The bipartite graph (i.e., the design procedure) associated with the bandgap voltage reference circuit. Sizing and biasing operators are part of the bipartite graph. Input parameters  $p_{in}$  (see Tables 4.8 and 4.9) are on the top of the graph

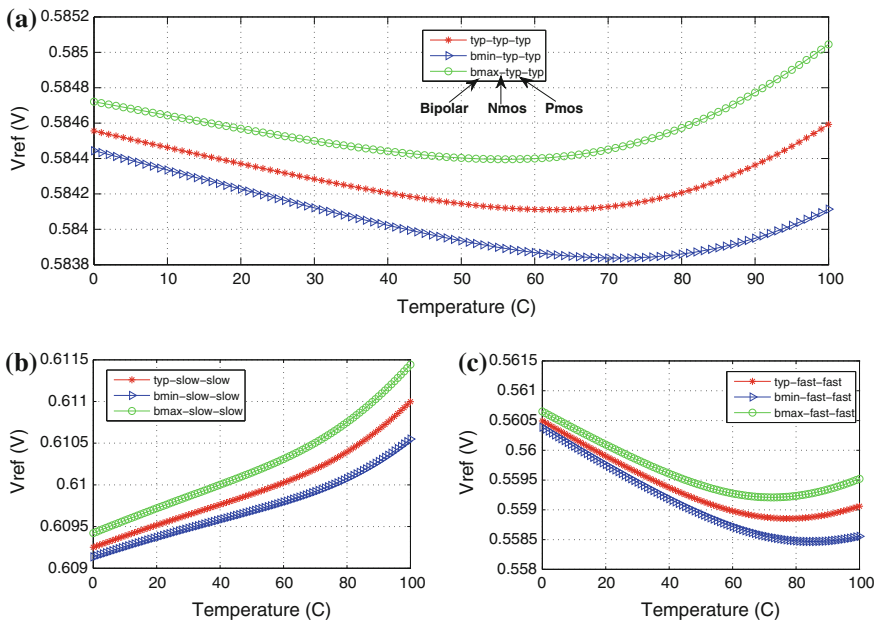
**Table 4.10** Computed width for the bandgap voltage reference circuit ( $P_{out}$ )

Transistor	Width	Transistor	Width	Transistor	Width
$W_{M3}$ ( $\mu\text{m}$ )	0.525	$W_{M6}$ ( $\mu\text{m}$ )	0.525	$W_{M6b}$ ( $\mu\text{m}$ )	1.695
$W_{M7}$ ( $\mu\text{m}$ )	3.235	$W_{M8}$ ( $\mu\text{m}$ )	32.055	$W_{M3b}$ ( $\mu\text{m}$ )	0.230
$W_{M9}$ ( $\mu\text{m}$ )	0.320				

types of components: *N*-type transistor (typical, slow, fast), *P*-type transistor (typical, slow, fast), and bipolar (typical,  $b_{min}$ ,  $b_{max}$ ), respectively. Therefore, we chose the SPICE netlists to simulate the corners of these components as follows: the first netlist for (typical, typical, typical), the second one for (fast, fast,  $b_{max}$ ), and the third one for (slow, slow,  $b_{min}$ ). We use SPICE netlist to load specific corners, in order to optimize the circuit process deviation. Actually, there are 27 ( $3^3$ ) combination of the corner netlist. Here, we keep only three cases, all typical, all slow and all fast.

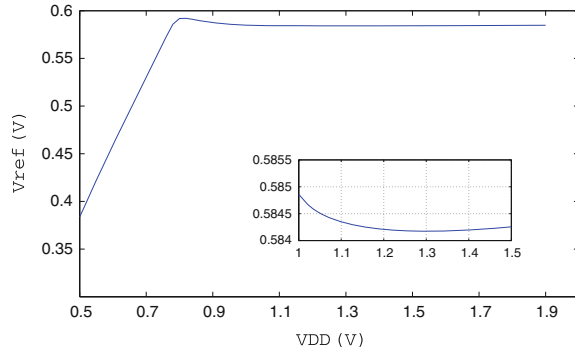
Figure 4.17a–c represents, respectively, a SPICE DC temperature sweep simulation from 0 to 100 °C. Figure 4.17a represents 3 curves corresponding to 3 sets of parameters (typical,  $b_{min}$ ,  $b_{max}$ ) for bipolar, while the *P*-type transistor and *N*-type transistor are set to the typical case. Figure 4.17b represents 3 curves corresponding to 3 sets of parameters (typical,  $b_{min}$ ,  $b_{max}$ ) for bipolar, while the *P*-type transistor and *N*-type transistor are set to the slow case. Figure 4.17c represents 3 curves corresponding to 3 sets of parameters (typical,  $b_{min}$ ,  $b_{max}$ ) for bipolar, while the *P*-type transistor and *N*-type transistor are set to the fast case. This combination is generated to further verify the electrical behavior of the bandgap voltage reference circuit.

The simulated curves of  $V_{ref}$  versus temperature show that in condition of NMOS/PMOS corners are typical, there is a very clear compensation, and the lowest reference voltage point is around 65 °C. In condition of NMOS/PMOS corners are fast, the compensation phenomenon is less obvious. In condition of



**Fig. 4.17** Reference voltage versus temperature: **a** 3 extreme sets of bipolar parameters (typical for NMOS and PMOS). **b** 3 extreme sets of bipolar parameters (slow for NMOS and PMOS). **c** 3 extreme sets of bipolar parameters (fast for NMOS and PMOS)

**Fig. 4.18** The reference voltage dependence over  $V_{DD}$  (Typical, temperature = 37 °C)



NOMS/PMOS corners are slow, there is no compensation phenomenon, but the voltage variation from 0 to 100 °C is still less than 1.5 mv.

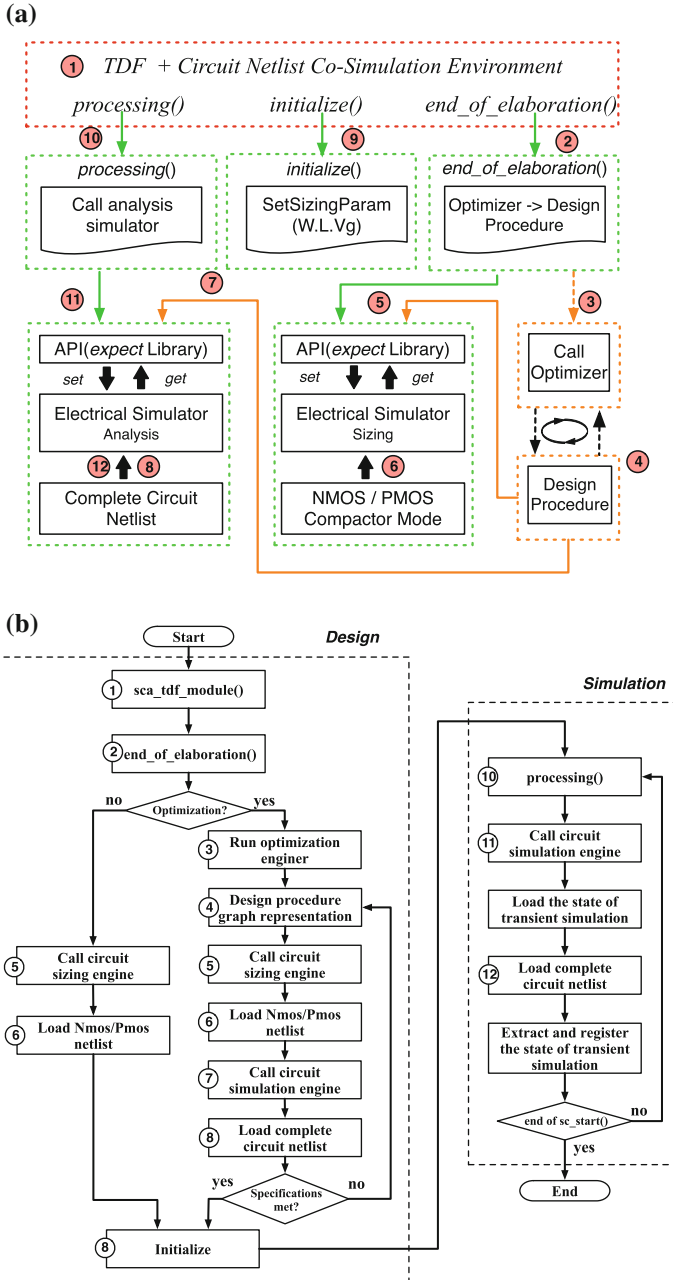
The variation of the reference voltage curve when  $V_{DD}$  changes from 0.5 to 1.9 V at 37 °C is shown in Fig. 4.18. The inserted plot shows the zoom-in view for  $V_{DD}$  between 1 and 1.5 V, which confirmed the circuit can work as low as 1 V.

## 4.5 Simulation Cycle in Co-simulation Environment

In this section, we explain in detail the simulation cycle in SystemC-AMS, Eldo co-simulation environment, which refers to **step 2** in Fig. 4.8. As shown in Fig. 4.19a, the co-simulation interface is related to the TDF module with circuit netlist. It involves three member functions: **end\_of\_elaboration()**, **initialize()**, and **processing()**. The **end\_of\_elaboration()** function calls the optimization engine, which invokes the design procedure at each optimization iteration, and the design procedure computes sizes and biases parameters ( $W$ ,  $V_G$  etc.) from the design parameters such as  $V_{EG}$ ,  $I_D$ , and  $L$ . The **initialize()** function sets these sizes and biases variables to circuit netlist. The signal processing function **processing()**, where the circuit netlist into the SPICE simulator is loaded, performs circuit-level transient simulation.

Note that in the member functions **end\_of\_elaboration()** and **processing()**, call two different simulators, named sizing simulator and analysis simulator. Both of them encapsulate an electrical simulator, mentioned in Sect. 4.2.2.2. The only difference between sizing simulator and analysis simulator is the transistor netlist loaded by the electrical simulator. The sizing simulator contains only two transistors: one PMOS and one NMOS, while the analysis simulator includes the complete circuit netlist.

To further describe the integration methodology, a flowchart is represented in the Fig. 4.19b, which introduces the algorithm to implement the design process and co-simulation in the standard simulation cycle of SystemC-AMS. In this algorithm, each step is defined by a number that corresponds to either a TDF module (①) or a

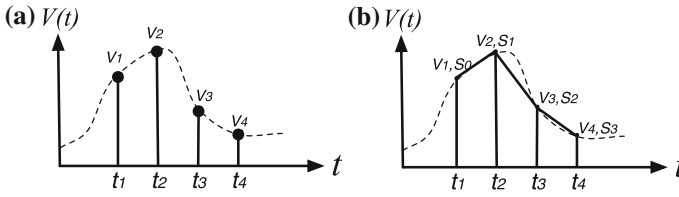


**Fig. 4.19** **a** SystemC-AMS, Eldo co-simulation environment. **b** Algorithm that permits to realize circuit sizing interface from system level to circuit level

function call (②–⑫) shown in Fig. 4.19a. The number of each step is the same for Fig. 4.19a and b.

This algorithm can be divided into two parts, which are *system design* and *system simulation*, respectively:

1. The *system design* part corresponds to the sizing and biasing of the circuit within the complete system-level description, and it is the bottom-up design part in Fig. 4.1.
  - In step ②, it defines all the required parameters used for circuit design procedure, such as the configuration of the optimizer, the specifications of the circuit. The sizing and biasing procedure is executed by using a sizing simulator (© in Fig. 4.1).
  - In case of performing optimization, an optimizer is called in step ③, just before calling the sizing and biasing procedure.
  - In step ④, the optimizer invokes the sizing and biasing procedure, which is presented by a graph as shown in Sect. 4.2.2.1.
  - In step ⑤, the sizing simulator loads the suitable electrical netlist NMOS/PMOS. Both transistors refer to a transistor compact model, entirely sizable and biasable through simulator interactive commands.
  - At each iteration of the optimization, the sizing simulator computes the sizes and bias values based on different design parameters.
  - The optimizer is closed in case the specifications are successfully met.
  - At the end of the optimization loop, the optimized sizes and bias values will be restored and transmitted in step ⑨.
  - This sizing simulator is closed before the starting step ⑩.
2. From step ⑩, until the end of execution, the steps correspond to the system simulation include the circuit-level propagation. It is the top-down simulation path in Fig. 4.1.
  - In steps ⑩, ⑪, and ⑫, at each *time step*, the signal interface passes the input samples and evaluates the simulated output samples. These steps are executed until the last input sample is processed.
  - At the first execution of step ⑩, an analysis simulator (Ⓓ in Fig. 4.1) is opened, it calls the complete circuit netlist at the step ⑫, and it is closed at the end of the system simulation.
  - During the simulation, a loading and registration of the state of the circuit are performed, respectively, before and after step ⑫ at each time step. These two operations refer to the initial condition of the circuit transient simulation (see Sect. 4.5.1 for more details).



**Fig. 4.20** **a** TDF signal with sampled values. **b** Transient simulation with a set of pulsewise linear signals

### 4.5.1 Transient Analysis Method

The TDF model of computation is not conservative, and it considers values that are discrete in time and value. However, we aim at performing conservative nonlinear simulations for the components described in SPICE netlist. To be able to handle such problem, we convert the TDF input signal shown in Fig. 4.20a to the piecewise linear version shown in Fig. 4.20b. This conversion will be considered as the stimuli signal during SPICE simulation.

The pulse width is set to the sampling period, and a transient analysis is performed during each period. At the beginning of the transient analysis, the voltages at nodes 1, 2, 3, 4, and 5 (The five nodes connect to all the small-signal capacitances in the circuit.) marked in Fig. 4.14 are, respectively, set to previous statement. At the end of current simulation, the value of each node is retrieved and used as the initial conditions for the simulation of the next time step.

[Input TDF signal] [Transient simulation]

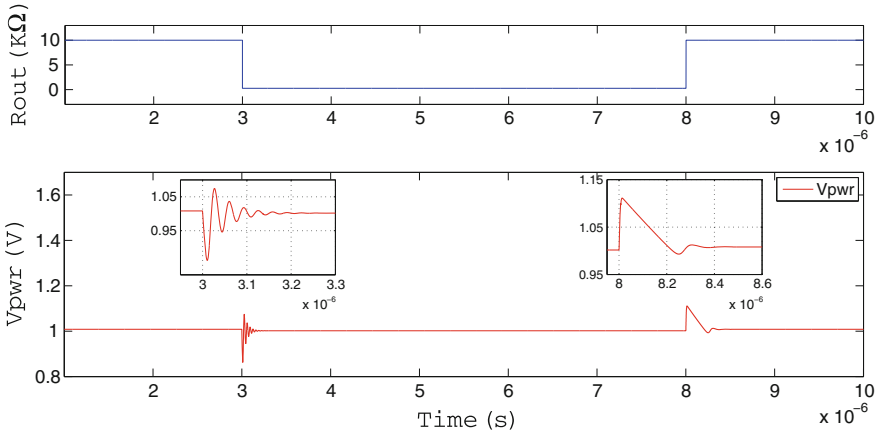
To construct a piecewise linear signal and perform the transient simulation from  $t_n$  to  $t_{n+1}$ , we should firstly know both the sample value  $V_n$  and  $V_{n+1}$ . Then, we consider the previous statement as the initial condition of this period. Finally, with the command `.TRAN tn dt uic` in SPICE netlist, it activates the transient analysis. Note that `dt` is the sampling period, Eldo automatically initializes all the node voltages itself as well as the option `uic` included in a `.TRAN` command.

Using the above approach, the unified platform for mixed-signal system design can mix non-conservative system-level behavior with conservative nonlinear circuit simulation.

## 4.6 Simulation Results

System responses against model responses for two different tests are given in Figs. 4.21 and 4.22. For testing the functionalities of this feedback system, we keep the most two sensitive blocs (PMOS, folded-cascode amplifier) in circuit netlist and model the others modules in SystemC-AMS (bandgap voltage reference circuit, rectifier), as shown in Fig. 4.8.

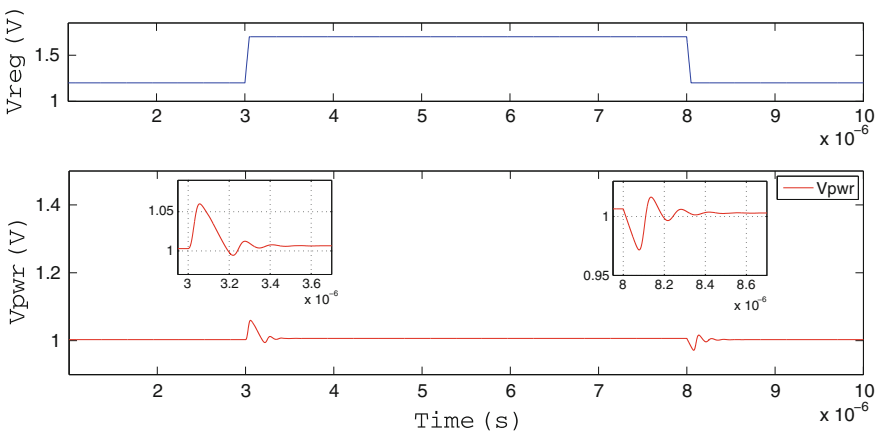




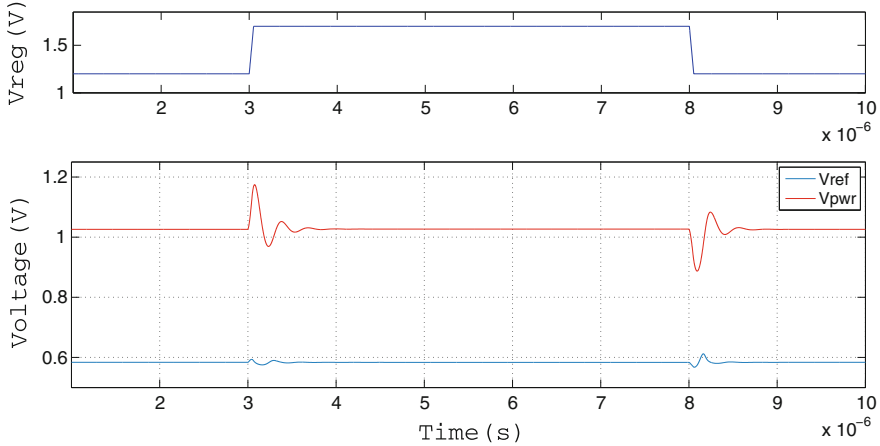
**Fig. 4.21** SystemC-AMS, Eldo co-simulation results, and output voltage waveform ( $V_{pwr}$ ) when the load changes from 10 k $\Omega$  to 250  $\Omega$

Figure 4.21 shows the transient waveform of the regulated voltage when the output load switches between 250  $\Omega$  and 10 k $\Omega$ . The line transient response is measured in condition of the  $V_{reg}$  is equal to 1.5 V. The difference between the voltage levels at the two stable states is equal to 6.2 mV. We notice that when the output load decreases form 10 k $\Omega$  to 250  $\Omega$ , there is an oscillation at the beginning before getting stable. This indicates that there might be stability issues in this configuration.

Figure 4.22 shows the transient waveform of the regulated voltage when the input voltage  $V_{reg}$  switches from 1.2 to 1.7 V within 250 ns. This line transient response is measured for load condition (5 k $\Omega$ , we choose the mean value between



**Fig. 4.22** SystemC-AMS, Eldo co-simulation results, and output voltage waveform ( $V_{pwr}$ ) when the input voltage switches from 1.2 to 1.7 V



**Fig. 4.23** Eldo simulation results, output voltage ( $V_{pwr}$ ), and reference voltage waveform ( $V_{ref}$ ) when the input voltage switches from 1.2 to 1.7 V

10 k $\Omega$  and 250  $\Omega$ , the load capacitance is set to 50 pF). The zoom part of the simulation confirms that it takes about 0.5  $\mu$ s to settle within 1 % of its final value. The difference between the voltage levels at the two stable states is equal to 3.5 mV.

Another simulation is shown in Fig. 4.23. It presents the simulation result of **step 3** in Fig. 4.8, where the whole circuit netlist is simulated only in Eldo. To compare with the co-simulation environment as shown in Fig. 4.22, we applied the same configuration to simulate the whole circuit. We notice that it takes 0.8  $\mu$ s to settle within 1 % of its final value. The difference between the voltage levels at the two stable states is equal to 1 mV. Besides, the transient response of  $V_{ref}$  indicates the optimized bandgap voltage reference circuit generates a very stable reference voltage for the regulator circuit.

All of the results can be seen in two aspects. Firstly, we propagate the circuit non-idealities and performances from circuit level to system level by using our platform. Secondly, the proposed platform works well in a feedback system, where the feedback loop is applied by introducing a delay. These observations demonstrate the effectiveness and reliability of our proposed modeling, design, optimization, and co-simulation methodology.

## 4.7 Conclusion

We present a platform for modeling, design, optimization, and co-simulation of mixed-signal systems. It is based on C/C++ language which can be used with SystemC-AMS. In this platform, an optimization engine is introduced for simulation-based hierarchical sizing and biasing using CHAMS. This optimization

engine meets both linear and nonlinear specifications. It is a fast design exploration of analog firm IP, where global exploration following the PEANO curves and Nelder–Mead simplex optimization is performed to realize local exploration.

The co-simulation principles make it possible to link circuit performances to system models, perform conservative nonlinear transient simulation for TDF model of computation, and enable feedback of non-functional properties in the functions models. The proposed approach is used to design and verify an implantable telemetry system. The simulation results prove the efficiency and correctness of our platform.

We foresee that the environment SystemC-AMS will be a common industry platform for modeling, design, optimization, and verification of mixed-signal systems. Compiling of the different level of abstractions to reach this goal, researchers should focus on the design aspects of the mixed-signal systems in SystemC-AMS.

## References

1. Accellera Systems Initiative, SystemC AMS 2.0 Standard. <http://www.accellera.org/downloads/standards/systemc/ams/> (2013)
2. SystemC AMS PoC Library Beta 2, May 2011, Fraunhofer Institute, Dresden. <http://www.systemc-ams.org/> (2011)
3. IEEE Computer Society. 1666-2005 IEEE Standard SystemC Language Reference Manual, pp. 1666–2005. IEEE
4. Zaidi, Y., Grimm, C., Hasse, J.: On mixed abstraction, languages, and simulation approach to refinement with SystemC AMS. *EURASIP J. Embed. Syst.* (2010). doi:10.1155/2010/489365
5. Formaggio, L., Fummi, F., Pravadelli, G.: A timing-accurate HW/SW co-simulation of an ISS with SystemC. In: *IEEE/ACM/IFIP*, pp. 152–157 (2004)
6. Frank, F., Weigel, R.: Co-simulation of SPICE Netlists and VHDL-AMS models via a simulator interface. In: *Signals, Systems and Electronics*, pp. 75–78 (2007)
7. Kirchner, T., Bannow, N., Grimm, C.: Analogue mixed signal simulation using SPICE and SystemC. In: *Design, Automation Test in Europe Conference Exhibition*, pp. 284–287 (2009)
8. Iskander, R., Louërât, M.-M., Kaiser, A.: Hierarchical sizing and biasing of analog firm intellectual properties. *Integr. VLSI J.* **46**(2), 123–148 (2013). doi:10.1016/j.vlsi.2012.01.001
9. Vachoux, A., Grimm, C., Einwich, K.: Extending SystemC to support mixed discrete-continuous system modeling and simulation. In: *IEEE International Symposium on Circuits and Systems*, pp. 5166–5169 (2005)
10. Mu, Z., Van Leuken, R.: SystemC-AMS model of a dynamic large-scale satellite-based AIS-like network. In: *Forum on Specification and Design Languages*, pp. 1–8 (2011)
11. Cenni, F., Scotti, S., Simeu, E.: Behavioral modeling of a CMOS video sensor platform using SystemC AMS/TLM. In: *Forum on Specification and Design Languages*, pp. 1–6 (2011)
12. Iskander, R., Louërât, M.-M., Kaiser, A.: Automatic DC operating point computation and design plan generation for analog IPs. *Analog Integr. Circ. Sig. Process. J.* **56**, 93–105 (2008). doi:10.1007/s10470-007-9075-3
13. Javid, F., Iskander, R., Louërât, M.-M.: Simulation-based hierarchical sizing and biasing of analog firm IPs. In: *IEEE International Behavioral Modeling and Simulation Conference*, pp. 43–48 (2009)
14. Levi, T., Lewis, N., Tomas, J., Fouillat, P.: IP-based methodology for analog design flow: application on neuromorphic engineering. In: *IEEE International NEWCAS-TAISA Conference*, pp. 343–346 (2008)

15. Saleh, R., Wilton, S., Mirabbasi, S., Hu, A., Greenstreet, M., Lemieux, G., Pande, P.P., Grecu, C., Ivanov, A.: System-on-chip: reuse and integration. In: Proceedings of the IEEE, pp. 1050–1069 (2006)
16. Javid, F., Iskander, R., Louërât, M.-M., Dupuis, D.: Analog circuits sizing using bipartite graphs. In: IEEE International Midwest Symposium on Circuits and Systems, pp. 1–4 (2011)
17. Libes, D.: Exploring Expect: A Tcl-Based Toolkit for Automating Interactive Programs. O’Reilly, Sebastopol (1995)
18. Liu, W.: MOSFET Models for SPICE Simulation: Including BSIM3v3 and BSIM4. Wiley, New York (2001)
19. NXP. MOS Model PSP level 103. [http://www.nxp.com/models/mos/\\_models/psp/](http://www.nxp.com/models/mos/_models/psp/) (2011)
20. Enz, C., Krummenacher, F., Vittoz, E.: An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications. *Analog Integr. Circ. Sig. Process. J.*, 83–114 (1995)
21. Gielen, G.E., Wambacq, P., Sansen, W.: Symbolic analysis for analog circuits: a tutorial overview. In: Proceedings of the IEEE 82, pp. 287–304 (1994)
22. Gielen, G.E., Sansen, W.: Symbolic Analysis for Automated Design of Analog Integrated Circuits. Kluwer Academic Publishers, The Netherlands (1991)
23. Dong, N., Roychowdhury, J.: General-purpose nonlinear model-order reduction using piecewise-polynomial representations. *IEEE Trans. Comput. Aid. Des.* **27**, 249–264 (2008). doi:10.1109/TCAD.2007.907272
24. Bernardinis, F.D., Jordan, M., Sangiovanni-Vincentelli, A.: Support vector machines for analog circuit performance representation. Proceedings of the Design Automation Conference, pp. 964–969 (2003)
25. Malak, A., Li, Y., Iskander, R., Durbin, F., Javid, F., Guebhard, J.-M., Louërât, M.-M., Tissot, A.: Fast multidimensional optimization of analog circuits initiated by monodimensional global peano explorations. *Integr. VLSI J.* **48**, 198–212 (2014). doi:10.1016/j.vlsi.2014.04.002
26. Poivey, C., Durbin, F., Haussy, J.: Méthodes d’optimisation Globale pour la CAO de Circuits Intégrés. Interface avec le simulateur Spice-PAC: Optimisation des performances de circuit non linéaires. Rapport CEA-R-5465 (1988)
27. Rust, J.: Using randomization to break the curse of dimensionality. *Econometrica: J. Econometric Soc.* **65**, 487–516 (1997)
28. Lagarias, J.C., Reeds, J.A., Wright, M.H., Wright, P.: Convergence properties of the Nelder-Mead simplex algorithm in low dimensions. *SIAM J. Optim.* **9**, 112–147 (1998)
29. Nelder, J.A., Mead, R.: A simplex method for function minimization. *Comput. J.* **7**, 308–313 (1965)
30. Spendley, W., Hext, G.R., Himsworth, F.R.: Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics* **4**(4), 441–461 (1962)
31. Sobot, R.: Implantable RF telemetry for cardiac monitoring in the murine heart: a tutorial review. *EURASIP J. Embed. Syst.* (2013). doi:10.1186/1687-3963-2013-1

# Chapter 5

## Framework for Formally Verifying Analog and Mixed-Signal Designs

Mohamed H. Zaki, Osman Hasan, Sofiène Tahar  
and Ghiath Al-Sammame

**Abstract** This chapter proposes a complementary formal-based solution to the verification of analog and mixed-signal (AMS) designs. The authors use symbolic computation to model and verify AMS designs through the application of induction-based model checking. They also propose the use of higher order logic theorem proving to formally verify continuous models of analog circuits. To test and validate the proposed approaches, they developed prototype implementations in Mathematica and HOL and target analog and mixed-signal systems such as delta-sigma modulators.

**Keywords** Formal verification · Computer-aided analysis · Symbolic verification · Theorem proving · Electronic design automation and methodology

### 5.1 Introduction

Analog and mixed-signal (AMS) integrated circuits are cornerstone components used at the interface between an embedded system and its external environment [1]. As such, AMS designs are dedicated for realizing data processing functions over physical signals, such as analog to digital (A/D) and digital to analog (D/A) converters. Computer-aided design (CAD) methods have been proposed and developed

---

M.H. Zaki (✉) · S. Tahar · G. Al-Sammame  
Concordia University, Montreal, Québec, Canada  
e-mail: mzaki@ece.concordia.ca

S. Tahar  
e-mail: tahar@ece.concordia.ca

G. Al-Sammame  
e-mail: sammame@ece.concordia.ca

O. Hasan  
National University of Sciences and Technology, Islamabad, Pakistan  
e-mail: osman.hasan@seecs.nust.edu.pk

to overcome challenges in the design process of AMS designs [2, 3]. Sophisticated CAD tools and concepts are then needed to provide unique insights into the behavior and characteristics of the integrated circuits, to help the designer select best design strategies. The verification of AMS designs is one of the most important issues in their design.

In general, there exists two main approaches for validating an electronic systems with respect to a set of given properties. The first method uses monitoring with simulation to check if a property is valid or not. However, since realistic electronics systems are accepting large numbers of input combinations, it is impossible to cover all the behaviors using simulation. The major research efforts today are centered to find a clever way to cover most operating modes through an intelligent generation of test cases and coverage analysis. The second method, formal verification, is exploring a mathematical model of the system in order to prove the correctness of its properties. The foundations of this method are based on logic, automata, and semantics in which roots originate from computational intelligence. For digital circuits, this is applied using, for example, model checking or satisfiability-based verification. A major obstacle here is that of state explosion as the number of states of the system is exponential in the number of state variables.

However, the situation with analog and mixed-signal designs is radically different. The continuous-time behavior of analog circuits is expressed using models of differential and algebraic equations, while discrete-time behavior is described using a system of recurrence equations (SRE). In fact, closed-form solutions and systematic mathematical analysis methods for these models exist typically only for limited classes of systems. Usually, designers use differential and difference equations models more with engineering and applied mathematics tradition, not related to the careful semantics and methodological concepts developed for modeling digital concurrent systems. However, as computer systems are becoming more complex, the importance of analog components rises as AMS systems become more often integrated. The verification with simulation alone is proven not to be enough, and formal methods are advocated to occupy a complementary currently used design methods for analog systems, as they already do for digital systems.

This chapter suggests changing the strategy by tackling the problem from the point of view of difference equations (DE) used to describe the discrete-time behavior of AMS designs. In fact, a basic understanding of discrete-time behavior is essential in the design of modern AMS designs. For instance, discrete-time signal processing is used in the design and analysis of data converters used in communication and audio systems. Moreover, discrete-time processing techniques based on switched capacitor methods are used extensively in the design of analog filters [4]. We extend the definition of DE in order to represent digital components. The model then is called a generalized SRE. Then, we define the algorithms of bounded model checking (BMC) [5] on the SRE model by means of an algebraic computation theory based on interval arithmetics [6]. We associate the bounded model checking with a powerful and fully decidable equational theorem proving to verify

properties for unbound time using induction. We also propose to use higher order logic theorem proving in order to formally verify continuous models of analog circuits. In order to facilitate the user-guided verification process, we develop a library of higher order logic models for commonly used analog components, such as resistor, inductor, and capacitor, and circuit analysis laws, such as Kirchhoff's voltage and current laws. These foundations along with the formalization of calculus fundamentals can be used to reason about the correctness of any AMS property, which can be expressed in a closed mathematical form, within the sound core of a theorem prover. We illustrate the proposed method on the verification of a variety of designs including  $\Delta\Sigma$  modulator and a voltage-controlled oscillator. The rest of the chapter is organized as follows: We start in Sect. 5.2 by discussing relevant related work. The bounded model checking methodology is presented in Sect. 5.3 followed by a description of the theorem proving verification methodology in Sect. 5.4 before concluding with a discussion in Sect. 5.5.

## 5.2 Related Work

Using formal methods, two types of properties are frequently distinguished in temporal logic: *Safety properties* state that something bad does not happen, while *liveness properties* prescribe that something good eventually happens. In the context of AMS designs, examples of safety properties can be about voltages at specific nodes not exceeding certain values throughout the operation. Such a property is important when designing AMS circuits, as a voltage exceeding a certain specified value can lead to failure of functionality and ultimately to a breakdown of the circuit which can result in undesirable consequences of the whole design. On the other hand, occurrence of oscillation or switching is good example of liveness properties. A bounded liveness property specifies that something good must happen within a given time; for example, switching must happen within  $n$  units of time, from the previous switching occurrence. This section overviews the research activities in the application of formal methods for the verification of AMS systems with respect to safety and liveness properties. A detailed literature overview of AMS formal verification can be found in [7].

Model checking and reachability analysis are proposed for validating AMS designs over a range of parameter values and a set of possible input signals. Common in these methods is the necessity for the explicit computation of the reachable sets corresponding to the continuous dynamics behavior. Such computation is usually approximated due to the difficulty to obtain exact values for the reachable state space (e.g., closed-form solutions for ODEs cannot be obtained in general). Several methods for approximating reachable sets for continuous dynamics have been proposed in the literature. They rely on the discretization of the continuous state space by using over-approximating representation domains such as polyhedra and hypercubes [8, 9]. On-the-fly algorithms have been proposed to

address shortcomings of the previous method [10–13]. The model checking tools d/dt [14], CheckMate [15], and PHaver [16] are adapted and used in the verification of a biquad low-pass filter [14], a tunnel diode oscillator and a  $\Delta\Sigma$  modulator [15, 17], and voltage-controlled oscillators [16].

### 5.3 First Verification Methodology: Bounded Model Checking

Our methodology aims to prove that an AMS description satisfies a set of properties. This is achieved in two phases: modeling and verification, as shown in Fig. 5.1. The AMS description is composed in general of a digital part and an analog part. For the analog part, it could be described using recurrence equations. For the digital part, it could be described using event driven models. The properties are temporal relations between signals of the system. Starting with an AMS description and a set of properties, the symbolic simulator performs a set of transformations by rewriting rules in order to obtain a normal mathematical representation called generalized SRE [18]. These are combined recurrence relations that describe each property blended directly with the behavior of the system. The next step is to prove these properties using an algebraic verification engine that combines bounded model checking over interval arithmetic [6] and induction over the normal structure of the generalized recurrence equations. Interval analysis is used to simulate the set of all input conditions with a given length that drives the

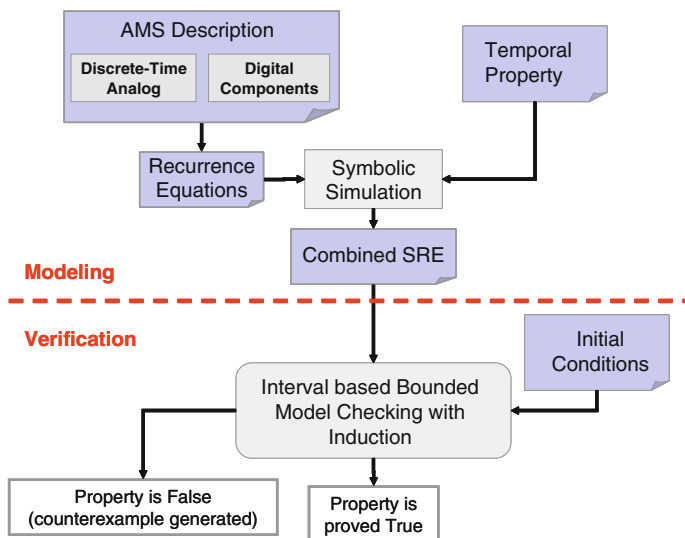


Fig. 5.1 Diagram of the bounded model checking verification



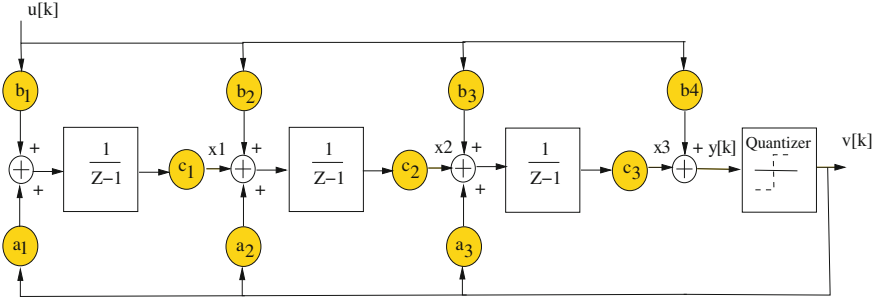


Fig. 5.2 Third-order  $\Delta\Sigma$  modulator

discrete-time system from given initial states to a given set of final states satisfying the property of interest. If for all time steps the property is satisfied, then verification is ensured; otherwise, we provide counterexamples for the non-proved property. Due to the over-approximation associated with interval analysis, divergence may occur, hence preventing the desired verification. To overcome such drawback, unbounded verification can be achieved using the principle of induction over the structure of the recurrence equations. A positive proof by induction ensures that the property of interest is always satisfied; otherwise, a witness can be generated identifying a counterexample.

### 5.3.1 Modeling and Specification

Recurrence equations are functional models used for the definition of relations between consecutive elements of a sequence. In the current work, we argue that, for certain classes of AMS designs, it is more natural to represent their behavior using recurrence equations rather than other conventional models such as hybrid automata. The notion of recurrence equation is extended in [18] to describe digital circuits with control elements, using what is called *generalized If-formula*. Such formalization, we believe, is practical in modeling hybrid systems such as discrete-time AMS design, where discrete components control the dynamics of the circuit, for example, the valuation of an analog signal. In mathematical analysis, we define recurrence equations by:

**Definition 1** (*Recurrence equation*) Let  $\mathbb{K}$  be a numerical domain ( $\mathbb{N}$ ,  $\mathbb{Z}$ ,  $\mathbb{Q}$  or  $\mathbb{R}$ ), a recurrence equation of order  $n_0 \in \mathbb{N}$  is a formula that computes the values of a sequence  $U(n) \in \mathbb{K}$ ,  $\forall n \in \mathbb{N}$ , as a function of last  $n_0$  values:

$$U(n) = f(U(n-1), U(n-2), \dots, U(n-n_0)) \quad (5.1)$$

**Definition 2** (*Generalized If-formula*) In the context of symbolic expressions, the generalized *If-formula* is a class of expressions that extend recurrence equations to describe digital systems. Let  $\mathbb{K}$  be a numerical domain ( $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$  or  $\mathbb{B}$ ), a generalized *If-formula* is one of the following:

- A variable  $x_i(n) \in \mathbf{x}(n)$ , with  $i \in \{1, \dots, d\}$ ,  $n \in \mathbb{N}$ , and  $\mathbf{x}(n) = \{x_1(n), \dots, x_d(n)\}$ .
- A constant  $C \in \mathbb{K}$
- Any arithmetic operation  $\diamond \in \{+, -, \div, \times\}$  between variables  $x_i(n) \in \mathbb{K}$
- A logical formula: any expression constructed using a set of variables  $x_i(n) \in \mathbb{B}$  and logical operators: *not, and, or, xor, nor, \dots*, etc.
- A comparison formula: any expression constructed using a set of  $x_i(n) \in \mathbb{K}$  and a comparison operator  $\alpha \in \{=, \neq, <, \leq, >, \geq\}$ .
- An expression  $IF(X, Y, Z)$ , where  $X$  is a logical formula or a comparison formula and  $Y, Z$  are any generalized *If-formula*. Here,  $IF(x, y, z) : \mathbb{B} \times \mathbb{K} \times \mathbb{K} \rightarrow \mathbb{K}$  satisfies the axioms:
  1.  $IF(True, X, Y) = X$
  2.  $IF(False, X, Y) = Y$

**Definition 3** (*Generalized SRE*) The following describes the transition relation of the system at the end of a simulation time unit  $n$ , by the way of a SRE; one equation for each element  $x$  in the system is:

$$x_i(n) = f_i(x_j(n - \gamma)), (j, \gamma) \in \varepsilon_i, \forall n \in \mathbb{Z} \quad (5.2)$$

where  $f_i(x_j(n - \gamma))$  is a generalized *If-formula*. The set  $\varepsilon_i$  is a finite non-empty subset of  $1, \dots, d \times \mathbb{N}$ , with  $j \in \{1, \dots, d\}$ . The integer  $\gamma$  is called the delay.

*Example 1* Consider the third-order discrete-time  $\Delta\Sigma$  modulator illustrated in Fig. 5.2. Such class of  $\Delta\Sigma$  design can be described using vectors recurrence equations:

$$X(k+1) = CX(k) + Bu(k) + Av(k) \quad (5.3)$$

where  $A, B$ , and  $C$  are matrices providing the parameters of the circuit,  $u(k)$  is the input signal,  $v(k)$  is the digital part of the system and  $b_4 = 1$ . In more detail, the recurrence equations for the analog part of the system are:

$$\begin{aligned} x_1(k+1) &= x_1(k) + b_1u(k) + a_1v(k) \\ x_2(k+1) &= c_1x_1(k) + x_2(k) + b_2u(k) + a_2v(k) \\ x_3(k+1) &= c_2x_2(k) + x_3(k) + b_3u(k) + a_3v(k) \end{aligned} \quad (5.4)$$

Also, the condition of the threshold of the quantizer is computed to be equal to  $c_3x_3(k) + u(k)$ . The digital description of the quantizer is transformed into a recurrence equation using the approach defined in [18]. Thus, the equivalent recurrence equation that describes  $v(k)$  is  $v(k) = IF(c_3x_3(k) + u(k) \geq 0, -a, a)$ , where  $a$  is the maximum output value of the quantizer, typically equals to one.

In order to reason about the functional properties of the design under verification, we need a language that describes the temporal relations between the different signals of the system, including input, output, and internal signals. We adopt the basic subset of linear temporal logic (LTL) [19], as the specification language. Each property  $P(n)$  is composed of two parts: a Boolean formula and a temporal operator. The Boolean formula  $p(n)$  is a recurrence time relation written using a logical formula (see Definition 2) built over the SREs of the system. To describe properties on analog signals such as current and voltages, atomic propositions,  $q(n)$ , are used, which are predicates (inequalities) over reals. The provided propositions are algebraic relations between signals (variables) of the system, such that the Boolean formula is a logical combination of such atomic propositions.

**Definition 4 (Atomic Property)** An atomic property  $q(n)$  is a logical formula defined as follows:  $q(n) = \chi(n) \diamond y$ , where  $\diamond \in \{<, \leq, >, \geq, =, \neq\}$ ,  $\chi(n)$  is an arithmetic formula over the design signals and  $y$  is an arbitrary value ( $y \in \mathbb{R}$ )

The temporal operator can be one of the basic LTL operators: Next (**X**), Eventually (**F**), and Always (**G**). As in traditional BMC, we define temporal operators regarding a bounded time step  $k$ . Thus, the verification of the temporal part is handled by the verification engine during reachability analysis.

*Example 2* Consider the  $\Delta\Sigma$  modulator of Example 1. The modulator is said to be stable if the integrator output remains bounded under a bounded input signal, thus avoiding the overloading of the quantizer in the modulator. This property is of a great importance since the integrator saturation can deteriorate circuit performance, hence leading to instability. If the signal level at the quantizer input exceeds the maximum output level by more than the maximum error value, a quantizer overload occurs. The quantizer in the modulator shown in Fig. 5.2 is a one-bit quantizer with two quantization levels, +1 V and -1 V. Hence, the quantizer input should be always bounded between specific values in order to avoid overloading [15]. The stability property of the  $\Delta\Sigma$  modulator is written as  $P(k) := \mathbf{G}p(k)$ , where

$$p(k) = (x_3(k) > -2 \wedge x_3(k) < 2) \quad (5.5)$$

The symbolic simulation algorithm is based on rewriting by substitution. The computation aims to obtain the SRE defined in the previous section. In the context of functional programming and symbolic expressions, we define the following functions [20].

**Definition 5** (*Substitution*) Let  $u$  and  $t$  be two distinct terms and  $x$  be a variable. We call  $x \rightarrow t$  a substitution rule. We use  $Replace(u, x \rightarrow t)$ , read “replace in  $u$  any occurrence of  $x$  by  $t$ ,” to apply the rule  $x \rightarrow t$  on the expression  $u$ .

The function  $Replace$  can be generalized to include a list of rules.  $ReplaceList$  takes as arguments an expression  $expr$  and a list of substitution rules  $\{R_1, R_2, \dots, R_n\}$ . It applies each rule sequentially on the expression.  $ReplaceRepeated(expr, \mathcal{R})$  applies a set of rules  $\mathcal{R}$  on an expression  $expr$  until a fixpoint is reached, as shown in Definition 6.

**Definition 6** (*Repetitive Substitution*) Repetitive substitution is defined using the following procedure:

```

ReplaceRepeated(expr, \mathcal{R})
  Begin
  Do
    expr_t = ReplaceList(expr, \mathcal{R})
    expr = expr_t
  Until FP(expr_t, \mathcal{R})
  End

```

A substitution fixpoint  $FP(expr, R)$  is obtained, if  $Replace(expr, R) \equiv Replace(Replace(expr, R), R)$ .

Depending on the type of expressions, we distinguish the following kinds of rewriting rules:

*Polynomial Symbolic Expressions*  $R_{Math}$  are rules intended for the simplification of polynomial expressions  $(\mathbb{R}^n[x])$ .

*Logical Symbolic Expressions*  $R_{Logic}$  are rules intended for the simplification of Boolean expressions and to eliminate obvious ones such as  $(and(a, a) \rightarrow a)$  and  $(not(not(a)) \rightarrow a)$ .

*If-formula Expressions*  $R_{IF}$  are rules intended for the simplification of computations over *If-formulas*. The definition and properties of the *IF* rules, such as reduction and distribution, are defined as follows (see [21] for more details):

- IF Reduction:  $IF(x, y, y) \rightarrow y$
- IF Distribution:  $f(A_1, \dots, IF(x, y, z), \dots, A_n) \rightarrow IF(x, f(A_1, \dots, y, \dots, A_n), f(A_1, \dots, z, \dots, A_n))$

*Equation Rules*  $R_{Eq}$  result from converting other equations in the SRE into a set of substitution rules.

*Interval Expressions*  $R_{Int}$  are rules intended for the simplification of interval expressions.

*Interval-Logical Symbolic Expressions*  $R_{Int-Logic}$  are rules intended for the simplification of Boolean expressions over intervals.

Rules  $R_{Int}$  and  $R_{Int-Logic}$  are described in more detail later on. In the case of symbolic expressions over  $\mathbb{R}$ , the normal form is obtained using a Buchberger-based algorithm for the construction of the Gröbner base. The symbolic

computation uses the repetitive substitution  $ReplaceRepeated(Expr, \mathcal{R})$  (defined in Definition 6) over the set of rules defined above as follows:

**Definition 7** (*Symbolic Computation*) A symbolic computation over the SREs is defined as:

$$Symbolic\_Comp(X_i(n)) = ReplaceRepeated(X_i(n), R_{simp})$$

where  $R_{simp}(t) = R_{Math} \cup R_{Logic} \cup R_{IF} \cup R_{Eq} \cup R_{Int} \cup R_{Int-Logic}$ .

The correctness of this algorithm and the proof of termination and confluence of the rewriting system formed by all above rules are discussed in [18].

*Example 3* Applying Definition 6 for the  $\Delta\Sigma$  modulator of Example 1, we obtain the following unified modeling for both the analog and discrete parts.

$$\begin{aligned} x_1(k+1) &= \text{if}(c_3x_3(k) + u(k) \geq 0, x_1(k) + b_1u(k) - a_1a, \\ &\quad x_1(k) + b_1u(k) + a_1a) \\ x_2(k+1) &= \text{if}(c_3x_3(k) + u(k) \geq 0, c_1x_1(k) + x_2(k) + b_2u(k) \\ &\quad - a_2a, c_1x_1(k) + x_2(k) + b_2u(k) + a_2a) \\ x_3(k+1) &= \text{if}(c_3x_3(k) + u \geq 0, c_2x_2(k) + x_3(k) + b_3u(k) \\ &\quad - a_3a, c_2x_2(k) + x_3(k) + b_3u(k) + a_3a) \end{aligned} \quad (5.6)$$

The expression of the property in Example 2 after symbolic simulation is:

$$\begin{aligned} p(k+1) &= \text{if}(c_3x_3(k) + u(k) \geq 0, \\ &\quad -2 < c_2x_2(k) + x_3(k) + b_3u(k) - a_3a, \\ &\quad c_2x_2(k) + x_3(k) + b_3u(k) + a_3a < 2) \end{aligned}$$

### 5.3.2 The Automated Verification Algorithm

The proposed verification algorithm is based on combining induction and bounded model checking to generate correctness proof for the system. This method is an algebraic version of the induction-based bounded model checking developed recently for the verification of digital designs [22]. We start with an initial set of states encoded as intervals as shown in Fig. 5.3. Then, iteratively the possible reachable successors states from the previous states are evaluated using interval analysis-based computation rules over the SREs, i.e., the output of this step is a reduced *If-formula* where all variables are substituted by intervals. If there exists a path that evaluates the property to be false, then we search for a concrete counterexample. Otherwise, if all paths give true, then we transform the set of current

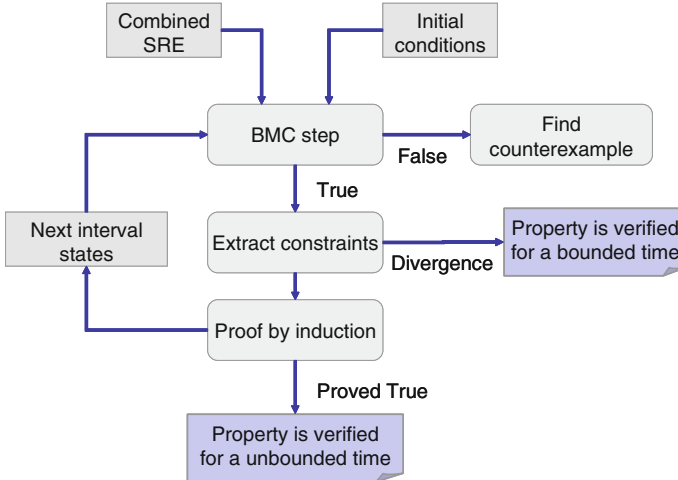


Fig. 5.3 Overview of the verification algorithm

states to constraints and we try to prove by induction that the property holds for all future states. If a proof is obtained, then the property is verified. Otherwise, if the proof fails, then the BMC step is incremented; we compute the next set of interval states and the operations are re-executed.

In summary, the verification loop terminates in one of the following situations:

- Complete Verification:
  - The property is proved by induction for all future states.
  - The property is false and a concrete counterexample is found.
- Bounded Verification:
  - The resource limits have been attained (memory or CPU) as the verification is growing exponentially with increasing number of reachability analysis steps.
  - The constraints extracted from the interval states are divergent with respect to some pre-specified criteria (e.g., width of computed interval states).

### 5.3.2.1 Background

**Bounded Model Checking:** Given a state transition system  $(S, I, \mathcal{T})$ , where  $S$  is the set of states,  $I \subseteq S$  is the set of initial states, and  $\mathcal{T} \subseteq S \times S$ , the general bounded model checking problem can be encoded as follows:

$$\text{BMC}(P, k) \triangleq I(s_0) \wedge \bigwedge_{i=0}^{k-1} \mathcal{T}(s_i, s_{i+1}) \rightarrow P(s_k) \quad (5.7)$$

where  $I(s_0)$  is the initial valuation for the state variables,  $\mathcal{T}$  defines the transition between two states, and  $P(s_k)$  is the property valuation at step  $k$ . For instance,

$$P(s_k) \triangleq \mathbf{GP}(s_k) = \bigwedge_{i=0}^k p(s_i) \quad \text{or} \quad P(s_k) \triangleq \mathbf{FP}(s_k) = \bigvee_{i=0}^k p(s_i)$$

In practice, the inverse of the property ( $\neg P$ ) under verification is used in the BMC algorithm [22], which we refer to as  $\overline{\text{BMC}}$ . When a satisfying valuation is returned by the solver, it is interpreted as a counterexample of length  $k$  and the property  $P$  is proved unsatisfied ( $\neg P$  is satisfied). However, if the problem is determined to be unsatisfiable, the solver produces a proof (of unsatisfiability) of the fact that there are no counterexamples of length  $k$ .

**Interval Arithmetics:** Interval domains give the possibility to extend the notion of real numbers by introducing a sound computation framework [6]. The basic interval arithmetics are defined as follows:

Let  $I_1 = [a, b]$  and  $I_2 = [a', b']$  be two real intervals (bounded and closed), the basic arithmetic operations on intervals are defined by:

$$I_1 \Phi I_2 \triangleq \{r_1 \Phi r_2 \mid r_1 \in I_1 \wedge r_2 \in I_2\}$$

with  $\Phi \in \{+, -, \times, /\}$  except that  $I_1/I_2$  is not defined if  $0 \in I_2$  [6]. In addition, other elementary functions can be included as basic interval arithmetic operators. For example, the exponential function  $\exp$  may be defined as  $\exp([a, b]) = [\exp(a), \exp(b)]$ . The guarantee that the real solutions for a given function are enclosed by the interval representation is formalized by the following property.

**Definition 8 (Inclusion Function)** [6] Let  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  be a continuous function, then  $F : \mathbb{I}^d \rightarrow \mathbb{I}$  is an interval extension (inclusion function) of  $f$  if

$$\{f(x_1, \dots, x_d) \mid x_1 \in X_1, \dots, x_d \in X_d\} \subseteq F(X_1, \dots, X_d) \quad (5.8)$$

where  $\mathbb{I}$  is the interval domain and  $X_i \in \mathbb{I}$ ,  $i \in \{1, \dots, d\}$ .

Inclusion functions have the property to be inclusion monotonic (i.e.,  $X_{\mathbb{I}} \subseteq Y_{\mathbb{I}} \rightarrow F(X_{\mathbb{I}}) \subseteq F(Y_{\mathbb{I}})$ ), hence allowing the checking of inclusion fixpoints [6].

**d-induction:** In formal verification, induction has been used to prove a property  $\mathbf{GP}(n)$  in a transition system by showing that  $P$  holds in the initial states of the system and that  $P$  is maintained by the transition relation of the system. As such, the induction hypotheses are typically much simpler than a full reachable state description. Besides being a complete proof technique, when it succeeds, induction is able to handle larger models than bounded model checking, since the induction step has to consider only paths of length 1, whereas bounded model checking needs

to check sufficiently long paths to get a reasonable confidence. Hence, simple induction is not powerful enough to verify many properties.

*d*-induction [22] is a modified induction technique, where one attempts to prove that a property holds in the current state, assuming that it holds in the previous *d* consecutive states. Essentially, induction with depth corresponds to strengthening the induction hypothesis by imposing the original induction hypothesis on *d* consecutive time frames. Given a state transition system  $(S, I, \mathcal{T})$ , where  $S$  is the set of states and  $I \subseteq S$  is the set of initial states,  $\mathcal{T} \subseteq S \times S$ , the *d*-induction proof is defined as  $d\text{-Ind}_{\text{proof}} \triangleq \psi_{d\text{-base}} \wedge \psi_{d\text{-induc}}$ , where  $\psi_{d\text{-base}}$  is the induction base and  $\psi_{d\text{-induc}}$  is the induction step defined as follows:

$$\psi_{d\text{-base}} \triangleq I(s_0) \wedge \bigwedge_{i=0}^{d-1} \mathcal{T}(s_i, s_{i+1}) \Rightarrow \bigwedge_{i=0}^d p(s_i)$$

and

$$\psi_{d\text{-induc}} \triangleq \bigwedge_{i=k}^{k+d} \mathcal{T}(s_i, s_{i+1}) \wedge \bigwedge_{i=k}^{k+d} p(s_i) \Rightarrow p(s_{k+d+1})$$

It is worth noting that when  $d = 1$ , we have exactly the basic induction steps defined in classical induction. Similar to the general induction methods, (un)satisfiability-based induction  $d\text{-Ind}_{\text{sat}}$  is the dual of the induction proof;  $\text{Ind}_{\text{sat}} = \neg d\text{-Ind}_{\text{proof}}$  with  $d\text{-Ind}_{\text{sat}} \triangleq \phi_{d\text{-base}} \vee \phi_{d\text{-induc}}$ , where the formulas  $\phi_{d\text{-base}}$  (the base step) and  $\phi_{d\text{-induc}}$  (the induction step) are defined as follows:

$$\phi_{d\text{-base}} \triangleq I(s_0) \wedge \bigwedge_{i=0}^{d-1} \mathcal{T}(s_i, s_{i+1}) \wedge \bigvee_{i=0}^d \neg p(s_i)$$

and

$$\phi_{d\text{-induc}} \triangleq \bigwedge_{i=k}^{k+d} \mathcal{T}(s_i, s_{i+1}) \wedge \bigwedge_{i=k}^{k+d} p(s_i) \wedge \neg p(s_{k+d+1})$$

The advantage of *d*-induction over classical induction is that it provides the user with ways of strengthening the induction hypothesis by lengthening the time steps *d* computed. Practically speaking,  $\phi_{d\text{-base}}$  is bounded model checking ( $\overline{\text{BMC}}$ ) as defined earlier in this section. For the case of systems with variables interpreted over real domains such as AMS designs, the satisfiability of the formulas with a given set of initial conditions requires algorithms to produce bounded envelopes for all reachable states at the discrete-time points. In the following, we demonstrate how to achieve BMC using interval arithmetics.

### 5.3.2.2 BMC Realization

The bounded forward reachability algorithm starts at the initial states and at each step computes the image, which is the set of reachable interval states. This procedure is continued until either the property is falsified in some state or no new



states are encountered. We evaluate the reachable states over interval domains, at arbitrary time steps. The verification steps for safety properties are shown in Algorithm 5.1. The AMS model, described as a set of recurrence equations, is provided along with the (negated) property  $\neg P(n)$  under verification. Initial and environment constraints  $Env\_Const$  are also defined prior to the verification procedure described in lines (1–12) as a loop for  $N_{\max}$  time steps. At each step  $n$ , we check whether the property is satisfied or not (line 2). If  $\neg P(n)$  is satisfied, then a counterexample is generated (line 9); if not, then we check if fixpoint inclusion is reached (line 3); otherwise, we update the reachable states (line 11) and go to the next time step of verification. The functions  $Prop\_Check$ ,  $Find\_Counterexample$ , and  $Update\_Reach$  are described below.

---

**Algorithm 5.1** Safety BMC
 

---

**Require:**  $x[n]$   
**Require:**  $\neg P(x[n])$   
**Require:**  $\mathcal{R}^0 = S_0$   
**Require:**  $Env\_Const$

```

1: for  $n = 1$  to  $N_{\max}$  do
2:   if  $Prop\_Check(\neg P[n], x[n]) == False$  then
3:     if  $Reach[T_{\sigma, x[n]}] \subseteq \mathcal{R}^{n-1}$  then
4:       return fixpoint reached
5:     else
6:        $Inc\_Step(n)$ 
7:        $\mathcal{R}^{n-1} = Update\_Reach(\mathcal{R}^{n-2}, Reach[x[n-1]])$ 
8:     end if
9:   else
10:     $Find\_Counterexample(\neg P[n], x[n], Env\_Const)$ 
11:   end if
12: end for

```

---

**Prop\_Check:** Given the property  $\neg P$ , apply algebraic decision procedures to check for satisfiability. The safety verification at a given step  $n$  can be defined with the following formula:

$$Prop\_Check \triangleq \mathbf{x}[n] = f(\mathbf{x}[n-1]) \wedge \neg P(\mathbf{x}[n]) \wedge \mathbf{x}[n-1] \in \mathbb{I}^d \quad (5.11)$$

**Update\_Reach**( $R_1, R_2$ ): This function returns the union of the states in the sets  $R_1$  and  $R_2$ .

**Reach**[ $x[n]:$ ]: This evaluates the reachable states over interval domains, at an arbitrary time step.

**Find\_Counterexample**( $\neg P(n), x[n], Env\_Const$ ): This function returns a counterexample, indicating a violation of the property, within the environment constraints.

Setting bounds on the maximum number of iterations ensures that the algorithm will eventually terminate in one of the following possibilities. If at a given time step  $n \leq N_{\max}$ , no new interval states are explored, then fixpoint inclusion guarantees

that the property will be always verified; otherwise, if a property is proved to be incorrect, then a counterexample is generated. If we reach the maximum number of steps  $n = N_{\max}$ , and no counterexample is generated, then the property is verified up to bounded step  $N_{\max}$ .

*Example 4* Given the design in Example 1 and the safety property in Example 2, we apply Algorithm 5.1. For instance, the correctness of the property  $P(k + 1)$  (see Example 3) depends on the parameter vectors  $A, B$ , and  $C$ , the values of variables  $x_1(k)$ ,  $x_2(k)$ , and  $x_3(k)$ , the time  $k$ , and the input signal  $u(k)$  (see Table 5.1). We verify the  $\Delta\Sigma$  modulator for the following set of parameters inspired from the analysis in [15]:

$$\begin{cases} a = 1 & a_1 = 0.044 & a_2 = 0.2881 \\ a_3 = 0.7997 & b_1 = 0.07333 & b_2 = 0.2881 \\ b_3 = 0.7997 & c_1 = c_2 = c_3 = 1 \end{cases}$$

The initial constraints define the set of test cases over which interval-based simulation is applied. If the property is *false*, as in the first and third cases in Table 5.1, then the verification is completed and a counterexample is generated from the simulated intervals. On the contrary, when the property is *true*, we have a partial verification result as it is bounded in terms of simulation steps. The second case in Table 5.1 illustrates such limitation.

Unfortunately, we note that in some cases (as case 4 in Table 5.1), divergence happens quickly, so we cannot deduce useful information on the property. We

**Table 5.1** Verification results for  $\Delta\Sigma$  modulator in Example 4

Initial constraints	Property evaluation for $n = 0$ to $N_{\max}$ cycles	Counterexample	CPU time used (s)
$0.028 \leq x_1(0) \leq 0.03$ $-0.03 \leq x_2(0) \leq -0.02$ $0.8 \leq x_3(0) \leq 0.82, u := 0.8$	$N_{\max} = 40$ $n = 0$ to $15$ true $n > 15$ false	$x_1[16] \mapsto 0.263$ $x_2[16] \mapsto 1.256, x_3[16] \mapsto 2.42$	1.5
$0.012 \leq x_1(0) \leq 0.013$ $0.01 \leq x_2(0) \leq 0.02$ $0.8 \leq x_3(0) \leq 0.82, u := 0.54$	$N_{\max} = 38$ true		31
$0.163 \leq x_1(0) \leq 0.164$ $-0.022 \leq x_2(0) \leq -0.021$ $0.8 \leq x_3(0) \leq 0.82, u := 0.6$	$N_{\max} = 40$ $n = 0$ to $17$ true $n > 17$ false	$x_1[19] \mapsto 0.163$ $x_2[19] \mapsto 0.886, x_3[19] \mapsto 2.47$	0.8
$0.012 \leq x_1(0) \leq 0.013$ $0.01 \leq x_2(0) \leq 0.02$ $0.8 \leq x_3(0) \leq 0.82, 0.58 \leq u \leq 0.6$	Divergent at time step 4		0.5

tackle such problem by extending the bounded model checking with an induction engine as proposed in the verification methodology.

### 5.3.2.3 Constrained Induction-Based Verification

In the following, we define an induction engine over the SREs for the safety property verification of AMS designs. The inductive proof, which is a special case of the  $d$ -induction described earlier in this chapter, for verifying a safety property  $P(n) = \mathbf{G}p(n)$ , can be derived by checking the formula  $\text{Ind}_{\text{proof}} \triangleq \psi_{\text{base}} \wedge \psi_{\text{induc}}$ , where  $\psi_{\text{base}}$  is the induction base and  $\psi_{\text{induc}}$  is the induction step defined as follows:

$$\psi_{\text{base}} \triangleq \forall s \in S_0 : I(s_0) \Rightarrow p(s_0) \quad (5.12)$$

and

$$\psi_{\text{induc}} \triangleq \forall s_k, s_{k+1} \in S : \mathcal{F}(s_k, s_{k+1}) \wedge p(s_k) \Rightarrow p(s_{k+1})$$

The core of the induction engine is a decision procedure function that checks satisfiability of algebraic formulas under certain constraints on quantified state variables.

**Definition 9** (*The Prove Function*)

$$\begin{aligned} \text{Prove}(\text{quant}(X, \text{cond}, \text{expr})) = \\ \text{If}(\text{Prop\_Verify}(\text{quant}(X, \text{cond}, \text{expr}))) = \text{True}, \\ \text{True}, \\ \text{Find\_Counterexample}(\text{cond} \wedge \neg \text{expr}) \end{aligned}$$

The decision procedure function *Prove* tries to prove a property of the form  $\text{quant}(X, \text{cond}, \text{expr})$ , using *Prop\_Verify*; otherwise, it gives a counterexample using *Find\_Counterexample*, where  $\text{quant} \in \{\forall, \exists\}$  define quantifiers over a set of state variables  $x$ ,  $\text{cond}$  is a logical combination of comparison formulas constructed over the variables  $x$  describing initial and environment constraints, and  $\text{expr}$  is an *If-formula* expression representing the property of interest, obtained after applying the symbolic rule outlined earlier. Similar to *Prop\_Check*, *Prop\_Verify* applies algebraic decision procedures to check for satisfiability, but for all time steps  $n$ . The safety verification can be defined with the following formula:

$$\text{Prop\_Verify} \triangleq \forall n \cdot (\mathbf{x}[n] = \text{SRE}(x[n])) \wedge P(\mathbf{x}[n]) \quad (5.13)$$

The *Prove* function generates a counterexample if the property of interest cannot be proved to hold using *Find\_Counterexample*( $\text{cond} \wedge \neg \text{expr}$ ). If a proof cannot be obtained, then we may need to find a particular combination of inputs and local signal values for which the property is not satisfied. The properties verification

using *Prove* starts by checking the validity at time  $t = 1$  and then at time  $t = n$  assuming that the properties are satisfied at time  $t = n - 1$ . Case splitting divides the property into subproperties for which validation results are conjoined to check the validation of the original property.

Let  $P$  be a property of the form  $quant(X, cond, expr)$ . We define the function *SplitProve* that depending on the *If-formula* structure of  $expr$ , applies the function *Prove*, or splits the verification. *SplitProve* is defined recursively as follows:

**Definition 10** (*The SplitProve Function*) According to the nature of  $expr$ , *SplitProve* can be one of the following:

- $expr$  is a comparison formula  $C$ ,  $SplitProve(quant(X, cond, C)) = Prove(quant(X, cond, C))$
- $expr$  is a logical formula of the form  $a \diamond b$ , with  $\diamond \in \{\neg, \wedge, \vee, \oplus, \dots\}$  and  $a, b$  are *If-formulas* that take values in  $\mathbb{B}$ .  
 $SplitProve(P) \simeq SplitProve(quant(X, cond, a)) \diamond SplitProve(quant(X, cond, b))$
- $expr$  is an expression of the form  $IF(q, l, r)$   $SplitProve(P) = SplitProve(quant(X, cond \wedge q, l)) \vee SplitProve(quant(X, cond \wedge \neg q, r))$

Let  $P(n)$  be the recurrence equation of the property  $P$  written as an *If-formula*,  $cond_{n_0}$  the initial condition at time  $n_0$ ,  $cond_n$  the constraints that are true for all  $n > n_0$ , and  $X$  the set of dependency variables of  $P(n)$ , and the proof by induction over  $n$  is defined as follows:

**Definition 11** (*Proof by Induction*)

$$SplitProve(ForAll(X_{n_0}, cond_{n_0}, P(n_0))) \\ \wedge \\ SplitProve(ForAll(n > n_0 \wedge X_n, n \in \mathbb{N} \wedge cond_n \wedge P(n), P(n+1)))$$

*Example 5* We verify the  $\Delta\Sigma$  modulator of Example 1 for two sets of parameters inspired from the analysis in [15]:

$$Param_1: \begin{cases} a = 1 & a_1 = 0.044 & a_2 = 0.2881 \\ a_3 = 0.7997 & b_1 = 0.044 & b_2 = 0.2881 \\ b_3 = 0.7997 & c_1 = c_2 = c_3 = 1 \end{cases}$$

$$Param_2: \begin{cases} a = 1 & a_1 = 0.044 & a_2 = 0.2881 \\ a_3 = 0.7997 & b_1 = 0.07333 & b_2 = 0.2881 \\ b_3 = 0.7997 & c_1 = c_2 = c_3 = 1 \end{cases}$$

We apply induction in order to verify the  $\Delta\Sigma$  modulator stability for the above sets of parameters and for two cases of conditions (state space constraints). Table 5.2 summarizes the verification results. The property is *True* if it is proved

**Table 5.2** Verification results for  $\Delta\Sigma$  modulator in Example 5

State space constraints	Property with Parameter <sub>1</sub>	Property with Parameter <sub>2</sub>			
Case 1	Values at $t = 0$	True	True		
	$0 \leq x_1(0) \leq 0.01$ $-0.01 \leq x_2(0) \leq 0$ $0.8 \leq x_3(0) \leq 0.82, u := 0.6$				
	Values at $t = n$				
	$-0.1 \leq x_1(n) \leq 0.1$ $-0.5 \leq x_2(n) \leq 0.5$ $0.5 \leq x_3(n) \leq 1.5, u := 0.6$				
Case 2	Values at $t = 0$	False	False		
	$0 \leq x_1(0) \leq 0.02$ $-0.03 \leq x_2(0) \leq -0.01$ $1 \leq x_3(0) \leq 1.4, u := 0.8$				
	Values at $t = n$			$x_2[k] \mapsto 0.4237$ $x_3[k] \mapsto 1.8378$	$x_2[k] \mapsto 0.2103$ $x_3[k] \mapsto 2$
	$-0.1 \leq x_1(n) \leq 0.1$ $-1 \leq x_2(n) \leq 0.5$ $-1 \leq x_3(n) \leq 2.5, u := 0.8$				

under the set of conditions and the set of parameters for all  $k > 0$ . If there is no  $k$  for which the property is valid, then it is *False*, and a counterexample is provided. When the property is valid for some values of  $k$  and not for other values, we say that the property is not proved and counterexamples are provided for both cases.

### 5.3.2.4 Combining $d$ -Induction and Interval-Based BMC

The  $d$ -induction-based verification algorithm is an incremental algorithm, where depth is incremented at each step and induction is applied on the new formulas until a  $d$ -length counterexample is generated or the property is proved. The verification steps are given in Algorithm 5.2.

The AMS model, described as a set of recurrence equations, is provided along with the (negated) property  $\neg P(n)$  under verification. Initial and environment constraints are also defined prior to the verification procedure described in lines (1–18) as a loop of depth  $N_{\max}$  steps. For each depth  $d < N_{\max}$ , we first check the initial  $d$ -induction step by verifying whether the property is verified for all steps up to this depth  $d$  (line 3). If the property is false, we generate a counterexample (lines 4). Before checking the induction step (line 10), we verify whether an inclusion fix-point is reached. If so, the verification ends as it will be trivial to check for the induction step as no new verification information can be implied. When we apply the induction step, where either the property is verified for unbounded time (line 11), otherwise, we conclude that the current depth is not enough to verify the property and the depth is incremented (line 14).

---

**Algorithm 5.2** d-induction based BMC
 

---

**Require:**  $x[n] := SRE(\mathcal{A})$   
**Require:**  $\neg P(x[n])$   
**Require:**  $\mathcal{R}^0 = S_0$   
**Require:**  $Env.Const$   
 1: initialize  $d = 1$   
 2: **for**  $d = 1$  to  $N_{max}$  **do**  
 3:   **if**  $Prop\_Check(\neg \bigwedge_{i=0}^d P(i), x[n]) == True$  **then**  
 4:      $Find\_Counterexample(\neg P(n), x[n], Env.Const)$   
 5:   **else**  
 6:     **if**  $Prop\_Check(\neg P(d), x[d]) == False$  **then**  
 7:       **if**  $Reach[x[d]] \subseteq \mathcal{R}^{d-1}$  **then**  
 8:          **return** fixpoint reached  
 9:       **else**  
 10:          **if**  $Prop\_Verify(\neg \bigwedge_{i=n}^{d+n} P(i), \bigwedge_{i=n}^{d+n} x[i]) == False$  **then**  
 11:            **return** verified  
 12:          **end if**  
 13:       **end if**  
 14:        $Inc\_Step(d)$   
 15:        $\mathcal{R}^{n-1} = Update\_Reach(\mathcal{R}^{n-2}, Reach[x[n-1]])$   
 16:     **end if**  
 17:   **end if**  
 18: **end for**

---

It is worth noting that constraints used in the induction steps are extracted from the previous reachable states. Hence, we strengthen the induction hypothesis by lengthening the time steps  $d$  computed. In case a counterexample needs to be generated, the extracted constraints allow for finding a partial path violating the property. Setting bounds on the maximum number of iterations ensures that Algorithm 5.2 will eventually terminate in one of the following possibilities. If the initial induction step fails, a counterexample is generated; otherwise, if at a given time step  $n \leq N_{max}$ , no new interval states are explored, and then, fixpoint inclusion guarantees that the property will be always verified. In case the induction step is verified true, then the algorithm terminates; otherwise, we increase the induction depth and restart the verification. If we reach the maximum number of steps  $n = N_{max}$ , and no counterexample is generated, then the property is verified up to bounded step  $N_{max}$ .

### 5.3.3 Applications

We have implemented a prototype for the presented verification algorithms using symbolic algebraic manipulation and real number theorem proving developed inside the computer algebra tool *Mathematica* [23].

### 5.3.3.1 Third-Order $\Delta\Sigma$ Modulator

We extended the verification results outlined throughout the chapter and summarized in Tables 5.1 and 5.2 by applying the  $d$ -induction algorithm to verify the stability of the third-order  $\Delta\Sigma$  modulator for different combinations of design parameters, inputs, and initial conditions. We are able to prove properties using the inductive BMC method, which we were unable to verify previously using the conventional BMC method (rows 2 and 4 in Table 5.1). In row 2 (Table 5.1), we are able only to verify the property for a bounded time step, with the  $d$ -induction BMC method; however, we are able to prove that the property will always hold (second row with  $\text{param}_2$  in Table 5.3). On the other hand, in row 4 (Table 5.1), the divergence occurs quickly; however, the property is proven *True* as shown in Table 5.3, row 4 with  $\text{param}_2$ .

### 5.3.3.2 Voltage-Controlled Oscillator

Recurrence equations have been proposed as a simplified operational modeling framework for certain AMS designs, in which precise continuous-time modeling poses challenging requirements to achieve simulation. As an instance, precise PLL verification necessitates the accounting for different time constants which render the simulation hard to achieve. Accordingly, at the early steps of the design, a discrete-time model is constructed representing the main functional aspects of the design. This can be later translated to a more refined model at subsequent design stages.

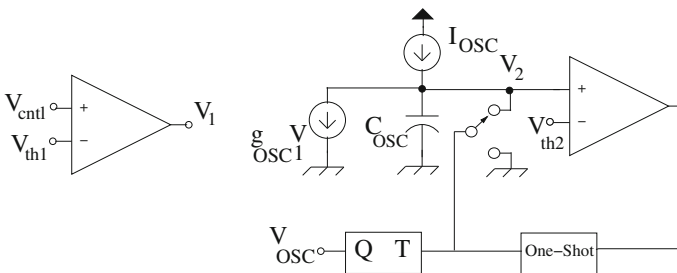
**Table 5.3**  $d$ -induction BMC verification results for the third-order  $\Delta\Sigma$  modulator

	State space constraints	Verification results	Verification details
Param <sub>1</sub>	$0 \leq x_1(0) \leq 0.01$ $-0.01 \leq x_2(0) \leq 0$ $0.8 \leq x_3(0) \leq 0.82, u := 0.6$	Proved true by $d$ -induction	$k$ -step = 3
	$0 \leq x_1(0) \leq 0.02$ $-0.03 \leq x_2(0) \leq -0.01$ $1 \leq x_3(0) \leq 1.4, u := 0.8$	Proved true by BMC then divergent	$k$ -step = 14
Param <sub>2</sub>	$0 \leq x_1(0) \leq 0.01$ $-0.01 \leq x_2(0) \leq 0$ $0.8 \leq x_3(0) \leq 0.82, u := 0.6$	Proved true by $d$ -induction	$k$ -step = 3
	$0.012 \leq x_1(0) \leq 0.013$ $0.01 \leq x_2(0) \leq 0.02$ $0.8 \leq x_3(0) \leq 0.82, u := 0.54$	Proved true by $d$ -induction	$k$ -step = 3
	$0 \leq x_1(0) \leq 0.02$ $-0.03 \leq x_2(0) \leq -0.01$ $1 \leq x_3(0) \leq 1.4, u := 0.8$	Proved false by counterexample	$k$ -step = 16
	$0.012 \leq x_1(0) \leq 0.013$ $0.01 \leq x_2(0) \leq 0.02$ $0.8 \leq x_3(0) \leq 0.82, 0.58 \leq u \leq 0.6$	Proved true by $d$ -induction	$k$ -step = 3

In the following, we apply the induction-based verification for the voltage-controlled oscillator (VCO) block of a charge pump PLL. A VCO is an oscillator, in which output frequency is controlled and varied by the applied input voltage. The recurrence equation modeling of the VCO is based on the circuit shown in Fig. 5.4 that describes a relaxation oscillator, in which output is a digital signal [4]. In the shown design, the input voltage is used to derive the VCO which according to some switching conditions triggers the one-shot timer, which in turn acts by controlling the discharging switch  $S_{\text{osc}}$  and the input to the toggle circuit. For instance, assume that the capacitor  $C_{\text{osc}}$  is initially discharged, it will be slowly charged by the current  $I_{\text{osc}}$  with the voltage  $V_2$  at each analysis step. Once the voltage  $V_{\text{th2}}$  across the capacitor  $C_{\text{osc}}$  exceeds the threshold voltage  $V_{\text{th2}}$ , then the output of the comparator goes to high (if it is not) and the one-shot timer is activated. The details about the functionality modeling of this VCO can be found in [4].

For correct operation of the VCO within the PLL design, it is required that the output will toggle from time to time (frequency of toggling is depending on the input voltage to the VCO). Such property has a flavor of liveness characteristics, which cannot be checked directly through induction. However, we use induction to check whether the input voltage variations will not lead to improper functionality. The verified property can be stated as follows: For a given set of input voltage variations,  $V_{\text{osc}}$  will always remain unchanged ( $\mathbf{G}V_{\text{osc}}[n] - V_{\text{osc}}[n - 1] = 0$ ). If this property is verified true, then we deduce that our choice of input signal range and/or parameters values is inappropriate for a correct behavior for the design.

We verified the property over several input signal  $V_1$  ranges, for different values of the transcendence of the VCO  $G_{\text{osc}}$  and the capacitor  $C_{\text{osc}}$ . The results in the experiments are obtained using the parameters proposed in [4]. First, we choose the range of input voltage as the interval  $[0, 2]$  volts. The property in this case is verified true. However, when we increase the input range to  $[0, 2.3]$ , the property becomes false. From those two results, we deduce that a possible correct functionality would require at least a larger swing for the input signal to the VCO. In another experiment, we preserve the first input voltage range while perturbing the set of parameter values and the property is verified again to false. Another interesting property we checked is the following safety criteria: For all possible input



**Fig. 5.4** Voltage-controlled oscillator



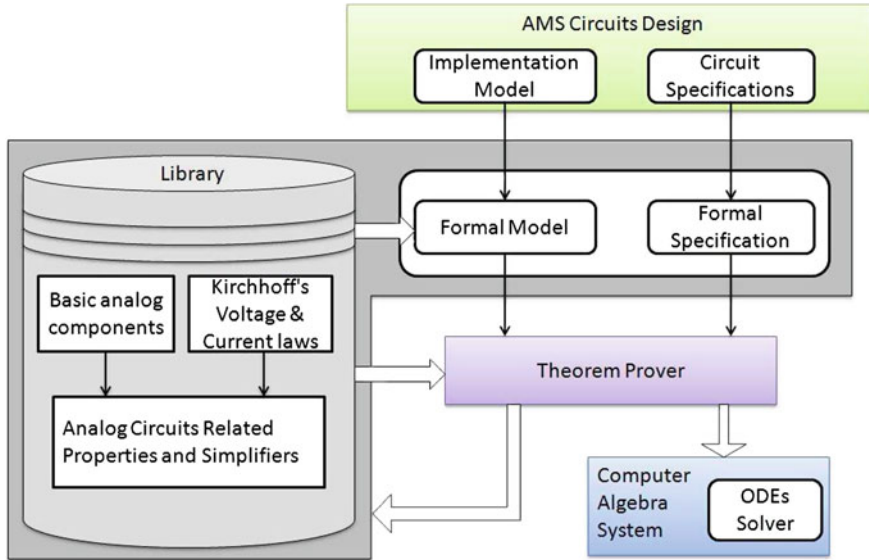
voltage ranges (i.e.,  $V_1 \in [-2.5, 2.5]$ ), the comparator input voltage  $V_2$  will never exceed certain bounds (i.e.,  $V_2 \in [-2.51, 2.51]$ ). This property is verified true. In fact, this verification is very beneficial as it provides us with the knowledge of the upper and lower bounds of the reachable state space. It is important to note that the correct functionality of the VCO requires the analysis over different voltage changes and notice the output. This would demand a dynamic verification method such as reachability or simulation, rather than a static method such as induction-based verification. Nevertheless, this latter technique allows the designer to have a better knowledge about the design limitations and to avoid and prune out undesirable constraints and parameters values when integrating the design with other components.

## 5.4 Second Verification Methodology: Theorem Proving

Most of the existing formal verification approaches work with abstracted discretized models of analog circuits (e.g., [24, 25]). This is mainly because of the inability to model and analyze continuous systems by the widely used formal verification techniques, such as model checking or automated theorem proving. Thus, despite the inherent soundness of formal verification methods, such analysis cannot be termed as absolutely accurate. Higher order logic theorem proving can be used to overcome these limitations due to the high expressiveness of the underlying logic. However, most of the existing higher order logic theorem proving-based analog circuit verification works (e.g., [26–28]) use discrete models of the given analog circuits by abstracting the continuous details. Thus, neither real numbers nor differential equations are used to represent the analog circuit behaviors in these analyses, which makes them prone to round-off and approximation errors.

We argue that the high expressibility of higher order logic can be leveraged upon to formalize the continuous models of analog circuit implementations and their desired specifications. Their equivalence can then be verified within the sound core of a theorem prover. Due to the high expressibility of higher order logic, the proposed approach is very flexible in terms of analyzing a variety of analog circuits and reasoning about their generic properties.

There are two main challenges in the proposed approach. Firstly, due to the undecidable nature of higher order logic, the proofs have to be done interactively, which may become very tedious due to the involvement of continuous elements and transcendental functions. Secondly, no closed-form solutions exist for a large number of analog circuits, and thus for these kinds of circuits, we cannot formally reason about approximate solutions in a theorem prover. We overcome both of these challenges in the proposed methodology [29], depicted in Fig. 5.5, by developing a library of analog circuit analysis definitions, theorems, and automatic simplifiers to minimize the user effort in the formal reasoning process and by using the support of computer algebra system for solving differential equations for which no closed-form solutions exist.



**Fig. 5.5** Proposed methodology for the formal verification of AMS circuits

The first step in the proposed methodology is to obtain an implementation model of the given analog circuit by using the behavior of its individual components and its overall structure. To facilitate this formalization, we developed a database of formal definitions of commonly used analog components, such as resistors, capacitors, and inductors, and circuit analysis laws such as Kirchhoff's voltage and current laws. The second step in the proposed methodology is to develop a formal model of the specification of the circuit, which is usually expressed as a differential equation. For this purpose, we choose the HOL4 theorem prover to implement the proposed methodology since it provides formalized libraries of real numbers and calculus foundations [30]. The third step is to verify the equivalence or implication relationships between the formalized implementations and specifications. To minimize the user interaction, required in this step, we formally verified most of the frequently used properties and developed some simplifying tactics with access to these results so that the users can verify most of the proof goals associated with analog circuit verification with minimal interaction. The main contribution in this regard is the formal verification of properties related to solutions of differential equation. Finally, if the differential equation corresponding to the given analog circuit does not have a closed mathematical solution, then it can be fed to a computer algebra system, such as Mathematica, to obtain its approximate solution. It is important to note here that the soundness of the analysis is not compromised at all by the computer algebra system link since it would only be invoked for the cases where a closed-form precise solution cannot be attained.

The main strengths of the proposed approach include its generic nature and accuracy. Any kind of analog circuit can be modeled, and its corresponding linear, nonlinear, homogenous, or non-homogenous differential equation can be formally expressed in higher order logic. If a closed-form solution for this equation exists, then it can be formally verified within the sound core of a theorem prover. In this case, modeling or analysis does not involve computer arithmetics or any discretization and thus, actual continuous models are formally verified. On the other hand, if a closed form does not exist, then the analysis is done using computer algebra systems, which is definitely the most accurate method in this scenario.

In the rest of this section, we first provide a formalization of the solutions of the second-order homogeneous linear differential equation to be able to reason about the solutions of differential equations for which a closed-form solution exist. Many interesting analog circuits lead to these kinds of equations. The formalization of circuit analysis fundamentals, i.e., KVL, KCL, and basic circuit components, is provided next. A couple of illustrative examples are then presented in the end.

### 5.4.1 Second-Order Homogeneous Linear Differential Equations

Second-order homogeneous linear differential equations are widely used to model analog circuits, and differential equations of higher order are seldom required in this domain. They can be mathematically expressed as follows:

$$p_2(x) \frac{d^2y(x)}{dx} + p_1(x) \frac{dy(x)}{dx} + p_0(x)y(x) = 0 \quad (5.14)$$

where terms  $p_i$  represent the coefficients of the differential equation defined over a function  $y$ . The equation is linear because (i) the function  $y$  and its derivatives appear only in their first power and (ii) the products of  $y$  with its derivatives are also not present in the equation. By finding the solution of the above equation, we mean to find functions that can be used to replace the function  $y$  in Eq. (5.14) and satisfy it.

We proceed to formally represent Eq. (5.14) by first formalizing an  $n$ th-order derivative function as follows [31]:

**Definition 12** (*Nth-order Derivative of a Function*)

$$\begin{aligned} \vdash & (\forall f \ x. \text{n\_order\_deriv } 0 \ f \ x = f \ x) \wedge \\ & (\forall f \ x \ n. \text{n\_order\_deriv } (n+1) \ f \ x = \\ & \quad \text{n\_order\_deriv } n \ (\text{deriv } f \ x) \ x) \end{aligned}$$

The function `n_order_deriv` accepts an integer  $n$  that represents the order of the derivative, the function  $f$  that represents the function that needs to be

differentiated, and the variable  $x$  that is the variable with respect to which we want to differentiate the function  $f$ . The function `deriv` accepts two parameters  $f$  and  $x$  and returns the derivative of the function  $f$  at point  $x$ . Thus, the function `n_order_deriv` returns the  $n$ th-order derivative of  $f$  with respect to  $x$ . Now, based on this definition, we can formalize the left-hand side (LHS) of an  $n$ th-order differential equation in HOL4 as the following definition [31].

**Definition 13** (*LHS of a Nth-order Differential Equation*)

$$\vdash \forall P \ y \ x. \text{diff\_eq\_lhs } P \ y \ x = \\ \text{sum}(0, \text{LENGTH } P) (\lambda n. (\text{EL } n \ P \ x) * \\ (\text{n\_order\_deriv } n \ y \ x))$$

The function `diff_eq_lhs` accepts a list  $P$  of coefficient functions corresponding to the  $p_i$ 's of Eq. (5.14), the differentiable function  $y$  and the differentiation variable  $x$ . It utilizes the functions `sum (0, m) f` and `EL m L`, which correspond to the summation  $(\sum_{i=0}^{m-1} f_i)$  and the  $m$ th element of a list  $L_m$ , respectively. It generates the LHS of a differential equation of order equal to the number of elements in the coefficient list  $P$  using the length of the list function `LENGTH`.

If the coefficients  $p_i$ 's of Eq. (5.14) are constants, then using the fact that the derivative of the exponential function  $y = e^{rx}$  (with a constant  $r$ ) is a constant multiple of itself  $dy/dx = re^{rx}$ , we can obtain the following solution of Eq. (5.14):

$$Y(x) = c_1 e^{r_1 x} + c_2 e^{r_2 x} \quad (5.15)$$

where  $c_1$  and  $c_2$  are arbitrary constants and  $r_1$  and  $r_2$  are the roots of the auxiliary equation  $p_2 r^2 + p_1 r^1 + p_0 = 0$ . In this chapter, we formally verify this result which plays a key role in formal reasoning about the solutions of second-order homogeneous linear differential equations [31].

**Theorem 1** *Differential Equation with distinct roots*

$$\vdash \forall a \ b \ c \ c1 \ c2 \ r1 \ r2 \ x. \\ (c + (b * r1) + (a * r1^2) = 0) \wedge \\ (c + (b * r2) + (a * r2^2) = 0) \Rightarrow \\ (\text{diff\_eq\_lhs } (\text{const\_list } [c; b; a]) \\ (\lambda x. c1 * (\text{exp } (r1 * x)) + \\ c2 * (\text{exp } (r2 * x)))) \ x = 0)$$

where  $[c; b; a]$  represents the list of constants corresponding to the coefficients  $p_0$ ,  $p_1$ , and  $p_2$  of Eq. (5.14);  $r_1$  and  $r_2$  represent the roots of the corresponding auxiliary equation as given in the assumptions;  $c_1$  and  $c_2$  are the arbitrary constants; and  $x$  is the variable of differentiation. The function `const_fn_list` transforms a list of

real numbers to the corresponding list of constant functions recursively, i.e., functions with data type  $\text{real} \rightarrow \text{real}$  that return a constant value for all values of arguments [31]. The formal reasoning about Theorem 1 is primarily based on the linearity property of higher order derivatives

### 5.4.2 Kirchhoff's Voltage and Current Laws

Kirchhoff's voltage law (KVL) and Kirchhoff's current law (KCL) form the most foundational circuit analysis laws. The KVL and KCL state that the directed sum of all the voltage drops around any closed network (loop) of an electrical circuit and the directed sum of all the branch currents leaving an electrical node is zero, respectively. Mathematically,

$$\sum_{k=1}^n V_k = 0, \sum_{k=1}^n I_k = 0 \quad (5.16)$$

where  $V_k$  and  $I_k$  represent the voltage drops across the  $k$ th component in a loop and the current leaving the  $k$ th branch in a node, respectively. The formalization is as follows [29]:

**Definition 14** (*Kirchhoff's Voltage and Current Law*)

$$\begin{aligned} &\vdash \forall V \ t. \text{kv1 } V \ t = \\ &\quad (\forall x. 0 < x \wedge x < t \Rightarrow \\ &\quad \quad (\text{sum } (0, \text{LENGTH } V) (\lambda n. \text{EL } n \ V \ x) = 0)) \\ &\vdash \forall I \ t. \text{kcl } I \ t = \\ &\quad (\forall x. 0 < x \wedge x < t \Rightarrow \\ &\quad \quad (\text{sum } (0, \text{LENGTH } I) (\lambda n. \text{EL } n \ I \ x) = 0)) \end{aligned}$$

The function `kv1` accepts a list  $V$  of functions of type  $(\text{real} \rightarrow \text{real})$ , which represents the behavior of time-dependant voltages in the given circuit and a time variable  $t$  as a *real* number. It return the predicate that guarantees that the sum of all the voltages in the loop is zero for all time instants in the interval  $(0, t)$ . Similarly, the function `kcl` accepts a list  $I$ , which represents the behavior of time-dependant currents and a time variable  $t$  and returns the predicate that guarantees that the sum of all the currents leaving the node is zero for all time instants in the interval  $(0, t)$ .

We now present some of the foundational formalization that is required to formally model analog circuits. The  $V$ - $I$  characteristics of fundamental analog components such as resistors, inductors, capacitors, and op-amps can be formalized as [29]:

**Definition 15** (*Resistor, Inductor Capacitor, and Op-amp*)

$$\begin{aligned}
&\vdash \forall R \ i. \text{resistor\_voltage } R \ i = (\lambda t. i \ t * R) \\
&\vdash \forall R \ v. \text{resistor\_current } R \ v = (\lambda t. v \ t / R) \\
&\vdash \forall L \ i. \text{inductor\_voltage } L \ i = \\
&\quad (\lambda t. L * \text{deriv } i \ t) \\
&\vdash \forall L \ v \ I_0. \text{inductor\_current } = \\
&\quad (\lambda t. I_0 + 1/L * \text{integral } (0, t) \ v) \\
&\vdash \forall C \ i \ V_0. \text{capacitor\_voltage } C \ i \ V_0 = \\
&\quad (\lambda t. V_0 + 1/C * \text{integral } (0, t) \ i) \\
&\vdash \forall C \ v. \text{capacitor\_current } = \\
&\quad (\lambda t. C * \text{deriv } v \ t) \\
&\vdash \forall V_{\text{pos}} \ V_{\text{neg}} \ A. \text{op\_amp\_voltage } V_{\text{pos}} \ V_{\text{neg}} \ A = \\
&\quad (\lambda t. A * (V_{\text{pos}} \ t - V_{\text{neg}} \ t))
\end{aligned}$$

The variables  $i$  and  $v$  represent the time-dependant current and voltage variables, respectively, in the above function definitions. While the variables  $R$ ,  $L$ , and  $C$  represent the constant resistance, inductance, and the capacitance of their respective components, respectively. The variables  $I_0$  and  $V_0$  are used in the definitions of inductance and capacitance to model the initial current in the inductor and the initial voltage across the capacitor, respectively. The parameters  $V_{\text{pos}}$ ,  $V_{\text{neg}}$ , and  $A$  represent non-inverting input, inverting input, and gain of an op-amp, respectively. The function `deriv` accepts two parameters  $f$  and  $x$  and returns the derivative of the function  $f$  at point  $x$ . Likewise, the function `integral` takes three parameters  $f$ ,  $a$ , and  $b$  and returns the integrated result of  $f$  in the interval  $(a, b)$ . All these functions return a  $(\text{real} \rightarrow \text{real})$  type function that models the corresponding time-dependant voltage or current.

### 5.4.3 Applications

#### 5.4.3.1 RLC Series Circuit

Serially connected resistor (R), inductor (L), and capacitor (C), or the RLC, circuit is one of the classical examples of an AMS circuit. It is also widely used in modeling parasitics in the metal interconnect of submicrometer ICs. We utilize the foundational formalization for analyzing AMS circuits, described in the last two subsections, to formally verify the electrical current flow relationship in the RLC circuit, shown in Fig. 5.6, with the intent to demonstrate the proposed methodology for formally analyzing AMS circuits.

The first step in the proposed methodology is to model the behavior of the given circuit in higher order logic. The behavior of the given circuit can be captured using the KVL as follows [31]:

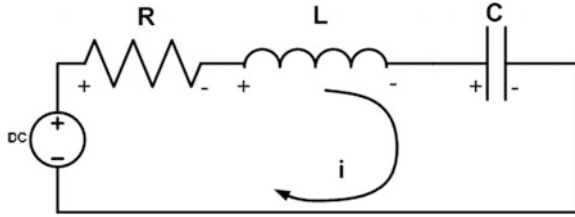


Fig. 5.6 RLC series circuit with constant voltage

**Definition 16** (*RLC Series Circuit Model*)

$$\vdash \forall R L C V V_0 i t. \text{rlc\_ckt } R L C V V_0 i t = \\ \text{kvl } [\text{resistor.voltage } R i; \\ \text{inductor.voltage } L i; \\ \text{capacitor.voltage } C i V_0; (\lambda t. -V)] t$$

The list input of the function `kvl` is composed of all the elements of the circuit that have a voltage drop. The `dc` voltage source `V` is modeled in this list as a time-independent constant. The next step in the proposed methodology is to obtain a differential equation representation of the given AMS circuit. We formally verified this relationship as follows [31].

**Theorem 2** *Differential Equation for the RLC Circuit*

$$\vdash \forall R L C V V_0 i t y. \\ (0 < y) \wedge (y < t) \wedge \\ (\forall x. 0 \leq x \wedge x \leq t \Rightarrow i \text{ differentiable } x) \wedge \\ (\forall x. 0 \leq x \wedge x \leq t \Rightarrow ((\lambda t. \text{deriv } i t)) \\ \text{differentiable } x) \wedge \\ (\text{rlc\_ckt } R L C V V_0 i t) \Rightarrow \\ (\text{diff\_eq.lhs}(\text{const\_list}[1/C;R;L]) i y = 0)$$

The conclusion of Theorem 2 describes the second-order differential equation corresponding to the RLC circuit given in the assumption using the function `rlc_ckt`. The theorem is verified under the assumptions that both the current function `i` and its first derivative are differentiable. It is also important to note that the theorem is valid for all time `y` in the interval  $(0, t)$ , where `t` represents the upper bound of the time for which the behavior of the function `rlc_ckt` is valid. Theorem 2 has been primarily verified using Theorem 1, some real analysis-based reasoning.

### 5.4.3.2 Delta-Sigma Modulator

In order to illustrate the proposed methodology, we present the formal verification of the first-order delta-sigma modulator, shown in Fig. 5.7, which is the widely used benchmark in formal verification of analog circuits.

The implementation model of this circuit can be obtained by applying KCL function at the input node of the op-amp:

**Definition 17** (*Implementation Model of Delta-Sigma Modulator*)

$$\begin{aligned} &\vdash \forall R C V_{in} V_{out} V_c V_{eq} y. \\ &\text{delta\_sigma\_imp } R C V_{in} V_{out} V_c V_{eq} y = \\ &\quad (\text{kcl } [\text{resistor\_current } R V_{in}; \\ &\quad \quad \text{resistor\_current } R V_{out}; \\ &\quad \quad \text{capacitor\_current } C (\lambda x. -V_c x)] t) \wedge \\ &\quad (V_{out} = (\lambda t. V_{eq} t - V_c t)) \end{aligned}$$

The next step is to formalize its specification:

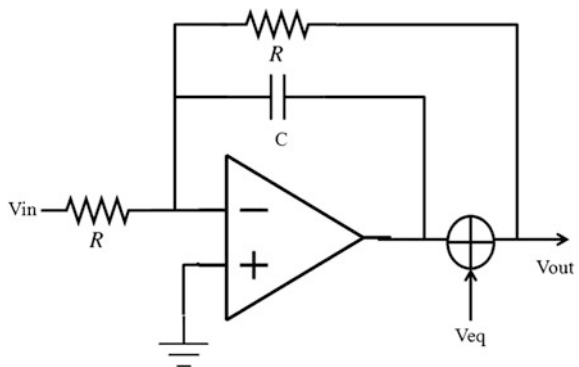
**Definition 18** (*Behavioral Model of Delta-Sigma Modulator*)

$$\begin{aligned} &\vdash \forall R C V_{in} V_{out} V_{eq} y. \\ &\text{delta\_sigma\_behav } R C V_{in} V_{out} V_{eq} y = \\ &\quad (\text{diff\_eq } [1; R * C] V_{out} y = \\ &\quad \quad -V_{in} y + \text{diff\_eq } [0; R * C] V_{eq} y) \end{aligned}$$

The function `diff_eq` accepts the list of coefficients of a differential equation, the differentiable function, and the differentiation variable and returns the corresponding differential equation.

Next, we formally verified the following implication between the implementation and specification of the given first-order delta-sigma modulator.

**Fig. 5.7** First-order delta-sigma modulator





**Theorem 3** *Implementation implies Specification*

$$\begin{aligned} \vdash \forall R C V_{in} V_c V_{out} V_{eq} t . \\ \text{delta\_sigma\_imp } R C V_{in} V_c V_{out} V_{eq} t \Rightarrow \\ \text{delta\_sigma\_behav } R C V_{in} V_{out} V_{eq} t \end{aligned}$$

The proof was very straightforward due to the available formally verified properties and simplifiers for real analysis-related reasoning in HOL. The differential equation of Definition 18 does not have a closed-form mathematical solution, and thus, we feed it to a computer algebra system to obtain its solution and thus other interesting characteristics of the delta-sigma modulator.

The proof scripts for both of the application theorems are composed of just 300 lines approximately. This is far less than the proof script for the formalization, presented in the previous two subsections, which is more than 3500 lines of HOL code. This fact clearly indicates the usefulness of our foundational formalization associated with the proposed methodology. Just like the case studies, presented in this section, our formalization results can be utilized to automatically verify interesting properties of a wide variety of analog circuits in a straightforward manner and the results would be guaranteed to be correct due to the inherent soundness of theorem proving.

## 5.5 Summary

Early uncovering of design flows is a daunting procedure during the integration of digital and AMS components. The heterogeneous verification of AMS designs poses great challenges for the development of System-on-Chip because of the infinite state space composed of continuous and discrete states. In this chapter, we have presented two complementary formal verification methodologies that address this obstacle. The rigorous characteristics of the methodology strengthen the verification and provide a support for simulation through state space exploration and corner cases identification. Experimental results have proven the feasibility of the approach. The symbolic-based method can find application along the design flow of complex AMS designs. Formal verification can be applied to check conformance of reduced order models. We are currently expanding the application of formal verification as a guidance during circuit sizing. In addition, our formally verified exact solutions of differential equations can also be used to formally verify error bounds for the numerical method-based solutions for the analog circuits for which the differential equations do not have closed-form mathematical solutions. To broaden the scope of analog circuit verification, we also plan to extend the library of analog circuit components with diodes and transistors, etc. We are also working on developing reasoning support for non-homogeneous linear differential equations.

Finally, the calculus theories available in HOL-Light [32] are based on multivariate real numbers and thus can model complex numbers. Moreover, this work has been recently extended to formalize some Laplace transform theory [33]. Our formalization can be ported in a very straightforward manner to HOL-Light to be able to benefit from these mathematical foundations, which would enable handling the formal analysis of analog circuits in the complex plane.

## References

1. Gielen, G.G., Rutenbar, R.A.: Computer-aided design of analog and mixed-signal integrated circuits. *Proc. IEEE* **88**(12), 1825–1852 (2000)
2. Shi, G., Tan, S.X.-D., Tlelo-Cuautle, E.: *Advanced Symbolic Analysis for VLSI Systems*. Springer, Berlin (2014) (ISBN 978-1-4939-1102-8)
3. Fakhfakh, M., Tlelo-Cuautle, E., Fernandez, F.V.: *Design of Analog Circuits through Symbolic Analysis*, 491 pages. Bentham Sciences Publishers Ltd., Sharjah (2012) (ISBN: 978-1-60805-425-1)
4. Johns, D., Martin, K.: *Analog Integrated Circuit Design*. Wiley, New York City (1997)
5. Biere, A., Cimatti, A., Clarke, E., Strichman, O., Zhu, Y.: Bounded model checking. *Adv. Comput.* **58**, 118–149 (2003)
6. Moore, R.E.: *Methods and Applications of Interval Analysis*. Society for Industrial and Applied Mathematics, Philadelphia (1979)
7. Zaki, M.H., Tahar, S., Bois, G.: Formal verification of analog and mixed signal designs: a survey. *Microelectron. J.* **39**(12):1395–1404 (2008)
8. Kurshan, R.P., McMillan, K.L.: Analysis of digital circuits through symbolic reduction. *IEEE Trans. Comput. Aided Des.* **10**(11), 1356–1371 (1991)
9. Hartong, W., Klausen, R., Hedrich, L.: Formal verification for nonlinear analog systems: approaches to model and equivalence checking. In: *Advanced Formal Verification*, pp. 205–245. Kluwer, The Netherlands (2004)
10. Greenstreet, M.R., Mitchell, I.: Reachability analysis using polygonal projections. In: *Hybrid Systems: Computation and Control*, LNCS, vol. 1569, pp. 103–116, Springer, Berlin (1999)
11. Yan, C., Greenstreet, M.R.: Verifying an arbiter circuit. *IEEE Form. Methods Comput. Aided Des.* 1–9 (2008)
12. Zaki, M.H., Al Sammane, G., Tahar, S., Bois, G.: Combining symbolic simulation and interval arithmetic for the verification of AMS designs. *IEEE Form. Methods Comput. Aided Des.* 207–215 (2007)
13. Walter, D., Little, S., Myers, C.: Bounded model checking of analog and mixed-signal circuits using an SMT solver. In: *Automated Technology for Verification and Analysis*, LNCS, vol. 4762, pp. 66–81. Springer, Berlin (2007)
14. Dang, T., Donze, A., Maler, O.: Verification of analog and mixed-signal circuits using hybrid system techniques. In: *Formal Methods in Computer-Aided Design*, LNCS, vol. 3312, pp. 14–17. Springer, Berlin (2004)
15. Gupta, S., Krogh, B.H., Rutenbar, R.A.: Towards formal verification of analog designs. In: *IEEE/ACM International Conference on Computer Aided Design*, pp. 210–217 (2004)
16. G. Frehse, B.H. Krogh, R.A. Rutenbar. Verifying analog oscillator circuits using forward/backward abstraction refinement. *IEEE/ACM Des. Autom. Test Eur.* pp. 257–262 (2006)
17. Lata, K., Roy, S.K.: Formal verification of analog and mixed signal designs using SPICE circuit simulation traces. *J. Electron. Test.* **29**(5), 715–740 (2013)

18. Al-Sammame, G.: *Simulation Symbolique des Circuits Decrits au Niveau Algorithmique*. PhD thesis, Université Joseph Fourier, Grenoble, France (2005)
19. Clarke, E.M., Grumberg, O., Peled, D.A.: *Model Checking*. MIT Press, Cambridge (2000)
20. Al Sammane, G., Zaki, M., Tahar, S.: A symbolic methodology for the verification of analog and mixed signal designs. *IEEE/ACM Des. Autom. Test Eur.* 249–254 (2007)
21. Moore, J.S.: Introduction to the OBDD algorithm for the ATP community. *J. Autom. Reason.* 12(1):33–45 (1994)
22. Amla, N., Du, X., Kuehlmann, A., Kurshan, R.P., McMillan, K.L.: An analysis of SAT-based model checking techniques in an industrial environment. In: *Correct Hardware Design and Verification Methods*, LNCS, vol. 3725, pp. 254–268. Springer, Berlin (2005)
23. Wolfram, S.: *Mathematica: A System for Doing Mathematics by Computer*. Addison Wesley Longman Publishing, USA (1991)
24. Frehse, G., Le Guernic, C., Donzé, A., Cotton, S., Ray, R., Lebeltel, O., Ripado, R., Girard, A., Dang, T., Maler, O.: Spaceex: scalable verification of hybrid systems. In: *Computer Aided Verification*, LNCS, vol. 6806, pp. 379–395. Springer, Berlin (2011)
25. Denman, W., Akbarpour, B., Tahar, S., Zaki, M., Paulson, L.C.P.: Formal verification of analog designs using MetiTarski. In: *Formal Methods in Computer Aided Design*, pp. 93–100. IEEE, New York (2009)
26. Hanna, K.: Reasoning about Real Circuits, vol. 859, pp. 235–253. Springer, Berlin (1994)
27. Ghosh, A., Vemuri, R.: Formal verification of synthesized analog circuits. In: *ACM/IEEE International Conference on Computer Design*, vol. 31, pp. 40–45 (1999)
28. Hanna, K.: Reasoning about analog level implementation of digital systems. *Form. Methods Syst. Des.* 16(2), 127–158 (2000)
29. Taqdees, S.H., Hasan, O.: Formal verification of continuous models of analog circuits. In: *Frontiers in Analog CAD*, Poster Paper (2013)
30. Harrison, J.: *Theorem Proving with the Real Numbers*. Springer, Berlin (1998)
31. Usman, M., Hasan, O.: Formal verification of cyber-physical systems: coping with continuous elements. In: *Computational Science and Its Applications*, LNCS-Part 1, vol. 7971, pp. 358–371. Springer, Berlin (2013)
32. Harrison, J.: A HOL theory of Euclidean space. In: *Theorem Proving in Higher Order Logics*, LNCS, vol. 3603, pp. 114–129. Springer, Berlin (2005)
33. Taqdees, S.H., Hasan, O.: Formalization of Laplace transform using the multivariate calculus theory of HOL-light. In: *Logic for Programing Artificial Intelligence and Reasoning*, LNCS, vol. 8312, pp. 744–758. Springer, Berlin (2013)

# Chapter 6

## Automatic Layout Optimizations for Integrated MOSFET Power Stages

David Guilherme, Jorge Guilherme and Nuno Horta

**Abstract** This chapter presents a design automation approach that generates automatically error-free area and parasitic optimized layout views of output power stages consisting of multiple power MOSFETs. The tool combines a multitude of constraints associated with DRC, DFM, ESD rules, current density limits, heat distribution, and placement. It uses several optimization steps based on evolutionary computation techniques that precede a bottom-up layout construction of each power MOSFET, its optimization for area and parasitic minimization, and its optimal placement within the output stage power topology network.

### 6.1 Introduction

In integrated audio power stages or power management units (PMG), it is necessary to design the layout of power transistors, but due to several technology design constraints and lack of investment in dedicated tools, this task has been mainly manual. Multiple constraints had hampered approaches based on parametric cells (*pcells*), respectively:

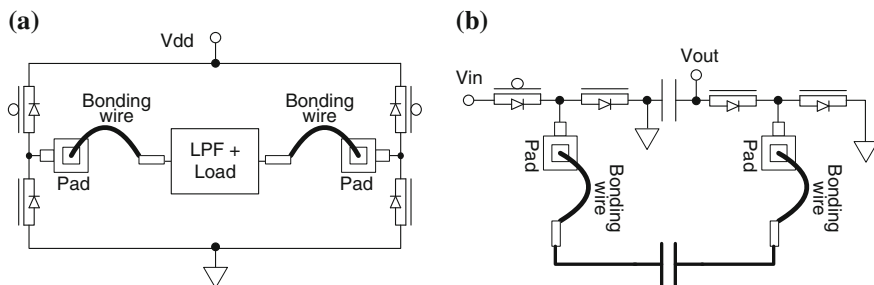
- Design kits do not supply transistor *pcells* meeting ESD rules and guidelines;
- Electromigration constraints of maximum current densities on metal tracks, vias, and contacts [1, 2];
- Design and manufacturing rules (DFM) specifically related with metal stress relief and etching effects [3].

---

D. Guilherme · N. Horta  
Instituto de Telecomunicações, Lisbon, Portugal  
e-mail: davidfcguilherme@gmail.com

N. Horta  
e-mail: nuno.horta@lx.it.pt

J. Guilherme (✉)  
Instituto Politécnico de Tomar, Tomar, Portugal  
e-mail: jorge.guilherme@ipt.pt



**Fig. 6.1** **a** H-bridge Class-D audio amplifier output stage. **b** Negative charge-pump power converter topology

Other approaches such as silicon compilers have been used for complex structures, such as memories and I/O cells [4, 5], and require the manual design of a large library of basic building cell layout variants. This implies a large effort setting up the compiler library for each retarget node, but the parametric flexibility of the block design remains limited to a small amount of layout combinations/permutations.

Therefore, designers have been forced to manually design the layout of power stages with the same basic topology over and over again, for different form factors and dimensions on several technologies and process nodes. This time-consuming process is also prone to bugs, but could easily be automated, saving design effort, increasing productivity, and speeding up the development phase.

For illustration purposes, Fig. 6.1 depicts two common CMOS power stage circuits, each having 4 power devices; Fig. 6.1a is an audio Class-D loudspeaker driver, and Fig. 6.1b is a voltage inverter charge pump.

This chapter presents an automatic layout design tool capable of delivering area and power optimized transistors, complying with DRC, DFM, and ESD rules and guidelines. Given channel dimensions, maximum current rating, and expected operating temperature, power transistor is automatically folded and partitioned, and the layout is generated to comply all requirements, including current density limits for long-term reliability. Reusability for fast process migration is achieved while maintaining vast design flexibility.

The tool assures the layout is clean by design, automatically launching physical verifications, and extracts the parasitic netlist of each device in an industry accepted sign-off tool (e.g., Calibre®). Finally, the tool also launches electrically simulations to rate the design and reports results.

Besides the automatic tool for device layout creation and verification reporting, this chapter also considers a framework to automatically floorplan the power stage devices, optimizing for circuit area, wire length at power nets, and heat transfer profiling.

## 6.2 Power MOSFET Especial Requirements

Additionally to generic design rules, the layout of each power MOSFET must follow additional rules and guidelines, especially electrostatic discharge (ESD) rules since are intended to connect to external pins.

During manufacturing, it is inevitable that the IC will suffer various kinds of ESD events. Different environments, wafer processing, packaging, testing, and human handling will generate different kinds of ESD [6]. To prevent permanent damage to the IC, which would reduce fabrication yield, foundries require several protection schemes and deliver verification decks to check ESD mandatory rules.

One of the most effective protection schemes, which every foundry requires, is the self-protection scheme associated to any externally connected device: No matter NMOS or PMOS, they all exhibit snapback phenomena [7], and to prevent a destructive effect by the positive feedback mechanism of the snapback, silicide-blocking (SAB) is required on the drain side of the MOSFET. This will prevent silicide formation on the terminal region connected to the PAD, which has the effect of increasing the series resistance exponentially at the discharging path. The resistance value depends on the instantaneous current passing at the corresponding finger junction. Then, the discharging current cannot be increased above a certain limit (e.g., 20 mA), and subsequently, current will be forced to flow to other non-saturated regions, thus distributing along finger junctions uniformly.

SAB can also be used on both junction terminals of the MOSFET, drain and source, which increases the device robustness to extreme ESD events, such as charged-device-model (CDM) class of events [6]. However, foundries seldom require self-protection on both terminals, at the drain side seems to be sufficient on most cases, since the source is usually connected to a large and strongly protected power rail.

Besides ESD rules, a power MOSFET must be carefully designed to comply with electromigration (EM) rules or current density limits [1, 2]. Those rules belong to the class design-for-manufacture (DFM) rules and intend to determine the reliability and lifetime of metal lines, vias, and contacts under heavy current for long periods of time. When a stress current is applied, metal ions move along the current pathway and their vacancy at the origin will cause resistance increase to the point tracks begin to fuse and the chip stops functioning.

The material lifetime prediction model can be calculated with the Black's equation [1]:

$$\text{TTF} = A \cdot J^{-n} \cdot e^{E_a/kT} \quad (6.1)$$

where TTF is the mean time to failure,  $A$  is a process constant,  $n$  is the exponent of current density ( $n = 1$ ),  $J$  is the current density flowing in metal,  $E_a$  is the activation energy ( $E_a = 1.0$  eV),  $k$  is the Boltzman constant, and  $T$  is the device temperature. It is easy to understand that an increase in the device temperature can cause a significant reduction in the TTF, and we can estimate the device expected lifetime due to chip-level self-heating effects.

To facilitate EM verification, foundries usually deliver tables of current density limits for several temperatures (e.g., 80, 100, 125 °C) and several metal track widths.

Additional DFM rules and requirements are usually triggered from the fact that power MOSFET layouts cover large silicon areas and favor large metal track widths and large arrays of vias and contacts. Rules related to metal stress relief (metal slot rules) and etching effects (density rules) must be taken to compliance. More advanced DFM rules related with lithography friendly patterns are not usually required since power MOSFETs are typically thick oxide devices using large mask patterns, and designed for power management ICs implemented in mature processes.

Regarding circuit-level layout considerations, the most important objectives are area minimization and power loss minimization. By placing devices in very close proximity helps reduce parasitic parameters and achieve high efficiency and performance. However, high-density power stages also tend to exhibit higher temperatures, hot spots and large heat gradients, which are detrimental to performance of neighboring circuits, reliability, and lifetime.

### 6.3 A Tool for Automatic Power MOS Layout Creation

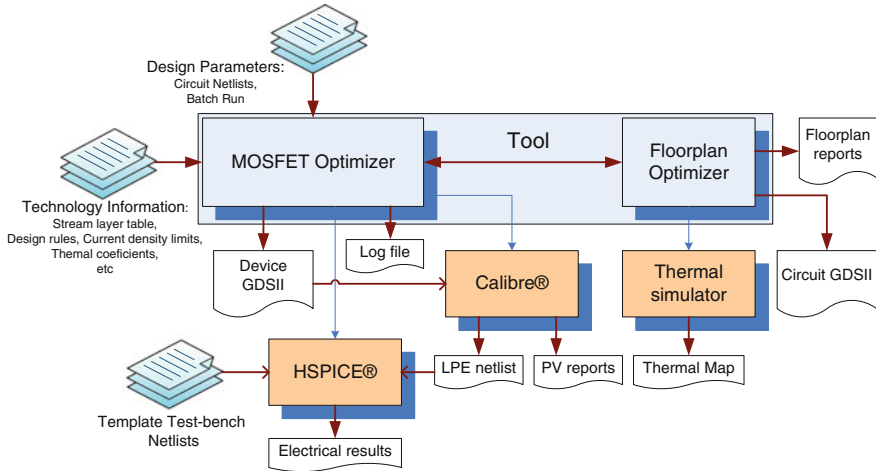
A framework of a tool for automatic layout of integrated MOSFET power stages comprises two operating phases:

1. The generation of optimal power device layouts;
2. The floorplanning of the power stages.

The tool architecture is depicted in Fig. 6.2. It must start by loading and parsing into internal dictionaries at least 2 files; one containing the stream layer table of the fabrication process and another with all mandatory design rules.

The tool must also be feed with several sets of design parameters and form factors for each target power MOSFET, which are extracted from an input SPICE compatible circuit netlist. The total number of parameters is dependent on the number of device transistors to create ( $k \in N$ ):

$$\left\{ \begin{array}{ll} \text{Channel width} & W = \{w_1, \dots, w_k\} \\ \text{Channel length} & L = \{l_1, \dots, l_k\} \\ \text{Number of fingers} & \text{NF} = \{\text{nf}_1, \dots, \text{nf}_k\} \\ \text{SAB Drain extension} & \text{De} = \{\text{de}_1, \dots, \text{de}_k\} \\ \text{SAB Source extension} & \text{Se} = \{\text{se}_1, \dots, \text{se}_k\} \\ \text{Upper metal plate} & M = \{m_1, \dots, m_k\} \\ \text{Max channel current} & I_{\max} = \{I_1, \dots, I_k\} \\ \text{Max working temperature} & T = \{t_1, \dots, t_k\} \\ \text{Max/Min form factor} & \text{FF}_{\max/\min} = \{\text{ff}_1, \dots, \text{ff}_k\} \end{array} \right.$$



**Fig. 6.2** Tool architecture

Based on the inputs, the tool first phase will be the optimization of isolated devices, using the *MOSFET Optimizer* block in Fig. 6.2. It will try to partition the MOS transistor dimensions in several ways, adjusting the number of fingers, finger dimensions, the number of contacts/vias, and the terminal metal paths, to comply with the given maximum channel current  $I_i$  at temperature  $t_i$ .

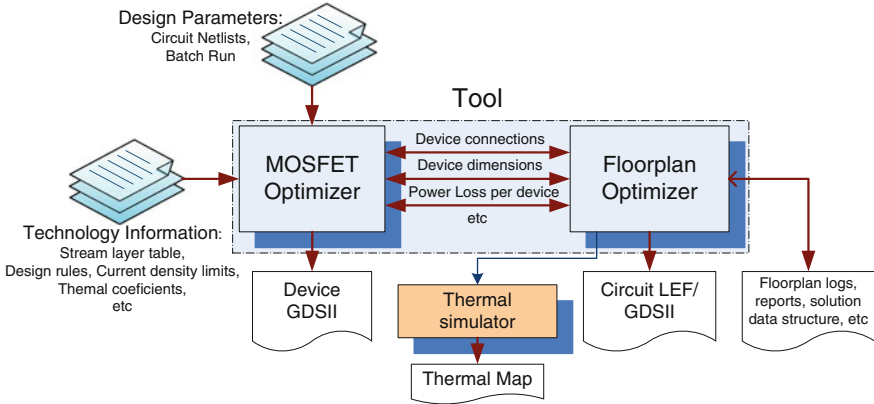
Using an internal optimization process, the tool discovers the best set of MOS parameters that minimizes device area while complying with all design rules and constraints and creates a hierarchical layout that exports in GDSII format.

After this layout creation step, the tool starts a GDSII verification phase, using Calibre® and HSPICE®. It first starts physical DRC/DFM verifications (PV) and then conducts a LPE netlist extraction for simulation purposes. Assuming there are no errors, the tool triggers two electrical simulations based on two template test benches: a DC sweep and a transient simulation. The first simulation measures the total resistive pathway from source to drain, including MOS  $R_{ON}$  and parasitic resistances. The second simulation measures dynamic power consumption when the transistor is switched ON and OFF at 2 MHz. Therefore, on the last step of this first phase, the designer has a clean layout of each device and two strong quality measures of the extracted layout to access figures of merit.

If the results are not good enough, the designer can restart this tool again, with different MOS dimensions. This method enables a fast and automated design cycle that can be used to create a large set of possible devices for a given problem and explore thereafter specification trade-offs.

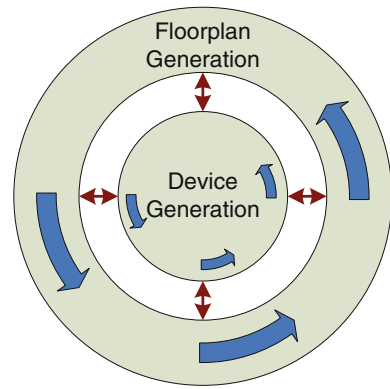
After the power devices of a given circuit topology have being sized, they must be laid out in a given space subject to optimal placement goals, such as optimization for overall circuit area and wire length of power nets. This second phase is conducted by the *Floorplan Optimizer* block as shown in Fig. 6.2.





**Fig. 6.3** Tool architecture—details for floorplan optimization phase

**Fig. 6.4** Concentric optimization problems



Figures 6.3 and 6.4 illustrate the general principle: During the first phase, all power devices are optimized and their layout is generated. The tool’s internal objects characterizing those devices are stored and reused again at the second phase, including explicit device boundary dimensions and pin positions. Additional information is also added: device connection information from the netlist and power loss estimations from the LPE simulations. This later information is required to optimize the floorplan of the power stage (optimization method described later), weighing several factors in a cost function: circuit area, wire length, and expected temperature spread over implantation area.

The optimized floorplan is streamed out to a second GDSII file, which instantiates abstract views of the previously designed devices in an optimal layout. The floorplan optimizer also creates several useful reports, including power density maps and thermal maps of optimal solutions. Thermal maps are created with the help of an external open-source FEM simulator.

The initial automatic layout problem could be solved globally: optimizing devices and circuit at the same time, but the division into 2 phases allows optimizations of simpler problems, containing lower number of design variables, and taking less time to reach a feasible solution. However, this concentric optimization framework cannot assure that the most optimal and global solution is reached at the end of the second phase. It is possible to reuse the archived solution as a first draft on a second pass into the tool, regenerating devices and the floorplan again. It is the responsibility of the user to decide whether design is good enough or needs a respin in the tool.

A different framework could consider the power devices equivalent to soft macros from the start and shape them to match the best floorplan, but such an approach could easily generate multiple design violations as ESD and DFM constraints would be very difficult to meet.

## 6.4 Floorplanning/Placement

The basic approaches to layout floorplan representations are as follows: the absolute representation and the topological representation. The first manipulates the absolute coordinates of the cells independently and was originally used on device-level analog layout computer-aided design (CAD) tools, allowing the exploration of a huge search space and placement of irregular shaped cells. The second—topological representation—was early adopted by digital flow placement tools and encodes the position relations between pair of cells, which are usually much more regular.

The absolute representation also allows illegal overlaps during the optimization operations since there is no relation between coordinates of neighbor cells [8]. Therefore, the optimization algorithm has to severally penalize all cell overlaps and at a given point of the optimization cycle, the number of feasible placements can be quite low. As a result, the optimization run-time can be very high and the quality of the final global placement is poor.

Topological representations reduce significantly the search space by encoding relative positions between cells and eliminate the cell overlap issue, but require more computational power to build feasible layouts. Topological floorplan representations can be classified into two categories: (1) slicing floorplans or (2) non-slicing floorplans.

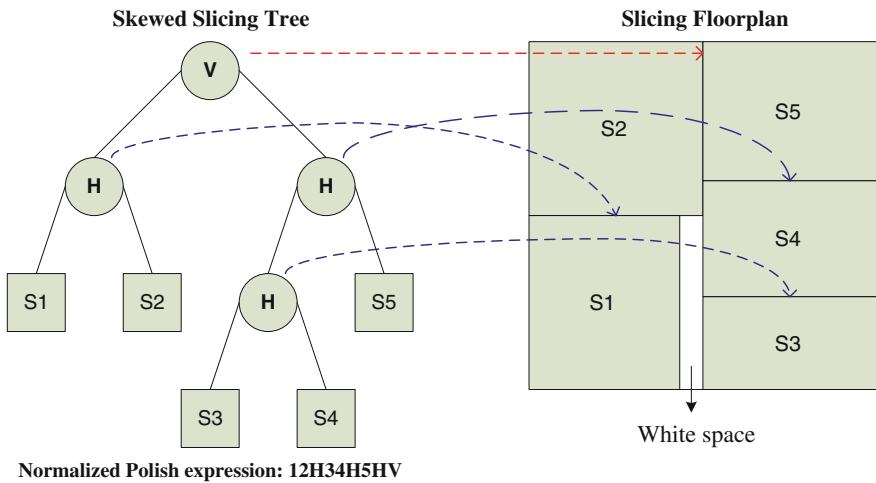
### 6.4.1 Slicing Floorplans

A slicing floorplan can be obtained by repetitively cutting the layout area horizontally or vertically (slicing), and the cells are organized in sets of slices as the result of the recursive bisection of the layout. The slicing sequence and the relative position of cells and slices can be recorded in a binary tree [9].

A slicing tree is a binary tree with cells at the leaves and cut types at the internal nodes. There are two types of cuts: V—the vertical cut where the left (right) branch represents a left (right) sub-floorplan or cell; and H—the horizontal cut where the left (right) branch represents the bottom (top) sub-floorplan or cell. However, the resulting floorplan can be constructed following slightly different cutting orders, and this means more than one slicing tree correspond to every floorplan. The non-uniqueness of a layout representation can enlarge significantly the solution domain space and reduce the optimization performance. Therefore, it is desirable to suppress redundant representations using a characteristic that turns them unique—if the tree does not contain a node of the same cut type as its right branch, then it is a skewed slicing tree. Skewed slicing trees are univocal representations of floorplans. Equivalent sequence representations of these binary trees are called normalized Polish expressions.

During floorplan optimization, the searching algorithm does not move cells explicitly, but alters the relative positions by modifying the slicing tree or the normalized Polish expression [9]. Figure 6.5 represents a slicing structure, obtained by recursively bisecting the floorplan area rectangles into smaller ones. The respective skewed slicing tree and normalized Polish expression are also depicted.

The set of all normalized Polish expressions form the solution space and its size can become quite large after a dozen cells. However, not all layout topologies have a slicing structure and using this representation will increase the occupied area by increasing white space utilization, thus making it less efficient. This problem becomes more severe for sets of cells with very different form factors and dimensions—this can be the case for several power stage topologies.



**Fig. 6.5** Example of a slicing floorplan, corresponding binary tree representation, and normalized polish expression

### 6.4.2 *Non-slicing Floorplans*

There are several non-slicing floorplan representations, which in turn can be divided into classes. Some deserve mentioning are as follows:

- **Class of mosaic representations**—This class commonly divides the floorplan implantation space in rectangular dissections (rooms) that form a mosaic, and each room is assigned exactly one cell. Some recent mosaic representations are as follows:
  - Corner block list (CBL);
  - Quarter-state sequence (Q-sequence);
  - Twin binary trees (TBT).
- **Class of compacted representations**—This class shares a special packing structure, for which the cells are compacted in relation to some corner of the floorplan region, e.g., the bottom-left corner, and no cell can be shifted down or left. The best known compacted representations are as follows:
  - Ordered tree (O-tree);
  - Upgraded binary tree (B\*-tree);
  - Corner sequence (CS).
- **General class representations**—This class gathers the most general and flexible floorplan representations:
  - Sequence pair (SP);
  - Bounded-sliceline grid (BSG);
  - Transitive closure graph (TCG);
  - Transitive closure graph with a sequence (TCG-S);
  - Adjacent constraints graph (ACG).

The last (general) class is especially suited for analog layout placement, since most can handle major topological constraints, including device symmetry, proximity, and matching. Performance metrics and comparison tables between representations are easily available in literature [9]; Table 6.1 lists some of the above.

Within the above-listed non-slicing representations, one of the most popular is the SP, which encodes “left–right” and “up–down” relations between cells [9]. The solution space can be explored by a general random search algorithm. The most common are the simulated annealing (SA) and the genetic algorithms (GA) [8, 10].

### 6.4.3 *Competing Floorplan Representations*

For this particular problem, the authors choose to use three competing floorplan representations, the normalized Polish expression, the B\*-tree, and the SP representations. These representations are used in conjunction with the SA algorithm to

**Table 6.1** Floorplan representations

Representation	Solution space	Packing time	Flexibility
Polish expression	$O(n! 2^{2.6n}/n^{1.5})$	$O(n)$	Slicing
CBL	$O(n! 2^{3n-3})$	$O(n)$	Mosaic
Q-sequence	$O(n! 2^{3n-1})$	$O(n)$	Mosaic
O-tree	$O(n! 2^{2n}/n^{1.5})$	$O(n)$	Compacted
B*-tree	$O(n! 2^{2n}/n^{1.5})$	$O(n)$	Compacted
CS	$O((n!)^2)$	$O(n)$	Compacted
SP	$(n!)^2$	$O(n^2)$ or $O(n \lg \lg n)$	General
BSG	$O(n! C(n^2, n))$	$O(n^2)$	General
TCG-S	$(n!)^2$	$O(n \lg n)$	General
ACG	$O((n!)^2)$	$O(n^2)$	General

solve the general power stage floorplan problem. The tool runs simultaneously three optimizations, each using a different representation. After the optimization process, all three solutions are stored and used to generate alternate floorplans, but the tool also automatically compares them and selects the best based on the final score value.

While the Polish representation is very fast to compute and will quickly converge to a draft solution, it is less efficient than the other two in terms of area optimization. The SP is the most general and flexible of the three, potentially returning the best floorplans, but it is also the heaviest in term of CPU utilization as it requires the calculation of the longest path or the longest common subsequence algorithms. The B\*-tree compact representation stays in the middle as a reasonable balance between computation time and solution quality.

Each of the three representations requires a convenient data structure which can be univocally translated to a floorplan—Table 6.2.

All three representations have a *device orientation* list that codes both device rotations ( $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$ ) and horizontal mirroring (flip). Device orientation is critical to total wire-length and parasitic parameter reduction.

While the Polish representation uses one single expression list to create a viable floorplan, as discussed previously at the *slicing floorplan* section, the other representations require two lists.

The B\*-tree representation has a first *B\*-tree list* to hold device IDs that will be used sequentially to build the tree accordingly to the order in the list. The second *B\*-tree device index* list establishes the relationship between the different device IDs. In other words, given any new leaf/branch to be inserted in the tree, the second list unequivocally defines where to add this new element.

The SP representation uses a pair of sequences stored in a *horizontal list* and a *vertical list* to encode “left–right” and “up–down” relations between devices. If two devices have the same sequence in the horizontal list and vertical list, this means that the first device is on the left of the second. If the two devices have a different sequence in the two lists, it means that the first device is on top of the second.

**Table 6.2** Floorplan representation data set

Polish expression	B*-tree	Sequence pair (SP)
Polish expression	B*-tree list	SP horizontal list
Device orientation	B*-tree device index	SP vertical list
	Device orientation	Device orientation

**Table 6.3** Floorplan representation moves

Polish expression	B*-tree	Sequence pair (SP)
Swaps 2 adjacent nodes	Swaps 2 nodes in the main list	Swaps 2 nodes in the horizontal list
Flips a bisection type	Swaps 2 nodes in the index list	Swaps 2 nodes in the vertical list
Change device orientation	Swaps 2 nodes in both lists	Swaps 2 nodes in both lists
	Change device orientation	Change device orientation

The allowable moves in each representation are listed in Table 6.3, and the contents are self-explanatory.

The move function must not restrict the design space. At the beginning of the move function, one random operation from the list of allowed operations is chosen. One common operation used on all representations is node swapping. For these kinds of moves, two randomly chosen nodes are swapped, changing their current relationship within the representation which can imply a change from left–right to a bottom–up relationship, or can just imply a change from or branch/leaf to another.

Every time a move is applied in the algorithm, it generates a unique slightly different floorplan representation, enabling the exploration of the solution space.

## 6.5 Thermal Evaluation

An important aspect of power MOS placement is the expected steady-state thermal profile of a set of power devices operating in very close proximity, as this is the case for integrated power stages. Hot-spot temperatures and large temperature spreads over the implantation area of power stages are detrimental to circuit performance and accelerate aging effects, and will also cause physical stress that further reduces reliability [11, 12].

The governing equation of the physics of heat transfer is derived from Fourier’s law and the conservation of energy:

$$k \cdot \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) + g - \rho \cdot C \cdot \frac{\partial T}{\partial t} = 0 \quad (6.2)$$

where  $T = T(x, y, z, t)$  is temperature as function of space and time,  $k$  is the thermal conductivity,  $\rho$  the mass density,  $C$  the specific heat capacity, and  $g$  is the rate of heat generation per unit volume. The heat equation is a typical example of a parabolic partial differential equation, and although analytical solutions can be found for simple cases, in integrated circuit heat transfer problems, numerical solutions are preferable due to the multiplicity of heat generating sources and the nature of the medium being non-homogeneous.

Numerical solutions use a mesh- or grid-like structure to perform a simulation that can be based on one of several methods, but the most suitable and accurate method is the FEM, since it is easy applicable to mediums possessing a multitude of boundary conditions and nonlinearities of thermal properties [13, 14].

The FEM discretize Eq. (6.2) into a matrix equation, which must be solved iteratively. To solve this equation faster, a method called model order reduction (MOR) can be employed to find an approximation of lower order [13]. Lower order matrix equations can be calculated much faster, and it is also possible to rewrite those equations in a state-space format suitable to transform simplified heat-transfer models into equivalent electrical R or RC networks [8]. Given that equivalent parametric RC networks playing as thermal models make electrothermal simulations possible.

The discretized version of the heat diffusion Eq. (6.2) is a good approximation for calculating the steady-state spatial temperature, although considering the medium homogeneous and completely linear thermal boundary characteristics. It could be used with FEM within an optimization cycle for very simple floorplans. However, it is prohibitively time-consuming to calculate the steady-state spatial temperature of each floorplan candidate solution when the number of cells is higher than a couple instances, since it requires the construction of a new thermal grid matrix for each evaluation.

An approximate measure which can represent the temperature of each power device and therefore could quickly calculate the power stage thermal profile is highly desirable. Some simplification methods can be found in literature [15–19]. One of the simplest approaches considers absolute temperature not as important as heat diffusion. Let us elaborate, starting from heat diffusion for each placed cells.

Besides internal heat generation, a very important factor in device temperature is heat diffusion between adjacent cells. The heat diffusion HD between two adjacent cells ( $S_1, S_2$ ) is proportional to their temperature difference  $T_{S_1} - T_{S_2}$  and the length of the shared cell boundary between them:

$$\text{HD}_{(S_1, S_2)} \approx (T_{S_1} - T_{S_2}) \cdot \text{boundary}_{(S_1, S_2)} \quad (6.3)$$

The problem with this approach is that we also do not know the exact temperatures of the cells for each floorplan candidate solution, and it is impossible to calculate the heat diffusion directly. The obvious alternative is to replace cell temperatures by yet another approximation, for example, considering each cell in isolation:

$$T_{Si} \approx P_{Si} \cdot R_{Si} \approx IC_{\tau}/k \cdot pd_{Si} \quad (6.4)$$

where  $T_{Si}$  should be the steady-state temperature,  $P_{Si}$  the power dissipation,  $R_{Si}$  the thermal resistance,  $IC_{\tau}$  the thickness of the chip from the device channel to the most convective chip boundary (usually the bulk adhesive in the chip package),  $k$  the average thermal conductivity of the material, and  $pd_{Si}$  the power density of the device implantation (power dissipated/layout area). Equations (6.3) and (6.4) could be incorporated into a matrix system and be solved numerically; however, this succession of approximations will be accumulating errors and the final temperature profile will suffer from severe deviations.

However, from a floorplan perspective, the absolute temperature is not as important as the temperature spread over the placement area, then using the cell power density on Eq. (6.3) allows the calculation of a total heat diffusion factor for each cell  $HD_{Si}$ , over all its neighbors:

$$HD_{Si} \propto \sum_j (pd_{Si} - pd_{Sj}) \cdot \text{boundary}_{(Si,Sj)} \quad (6.5)$$

After calculating (6.5) for every cell, a floorplan temperature spread index (total thermal diffusion) can be calculated by summing all heat diffusions:

$$D_T \propto \sum_i HD_{Si} \quad (6.6)$$

## 6.6 Device-Level Structural Design

MOSFET design is organized hierarchically, from a given set of parameters and an abstract top-level transistor, is decomposed into several layout levels, and ends in basic shapes—Fig. 6.6.

From top-to-bottom, the set of initial MOS parameters are rearranged in a way it minimizes several fitting goals, and the top-level transistor can be divided in an aggregated matrix of unit-level MOS transistors having a small number of fingers and viable (violations free) dimensions, both isolated or inserted into a matrix. In turn, each unit-MOS instance can be decomposed into a set of aggregated shapes, designed sequentially.

On top of the unit-MOS matrix, it is placed columns of stacked metal tracks, filled with vias, to short-circuit drain and source terminals in the  $YY$  direction—Fig. 6.7. Those terminal metal stacks are automatically generated from Metal-2 through a given top-plane  $m_i$ . The  $XX$  direction is dedicated to bulk and gate connections, and only Metal-1 is used for this purpose.

The algorithm flowchart is depicted in Fig. 6.8. The core function of the tool is the *lay\_pwrmos* method, responsible for designing aggregated power MOSFETs. This method starts by loading layer and rule files (context dictionaries) in order to



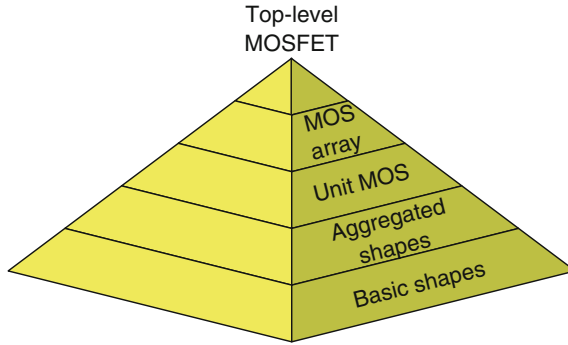


Fig. 6.6 Top-to-bottom design approach

do some calculations related with EM compliance limits and expected active area. Those calculations are not based on any generated layout, but pre-layout estimations to be used as rating equations on the next step, an optimization method by random search.

The optimization method uses the GA to select the best hierarchical partition and optimal unit-MOS dimensions that minimize area and parasitics. After the optimization is done, a recalculation of parameters is executed to take in account terminal pitch and guardring parameters. Thereafter, the unit-MOS can be effectively created and aggregated in the top-level layout matrix, which will also incorporate several unit terminal connection cells instantiated around the aggregation matrix.

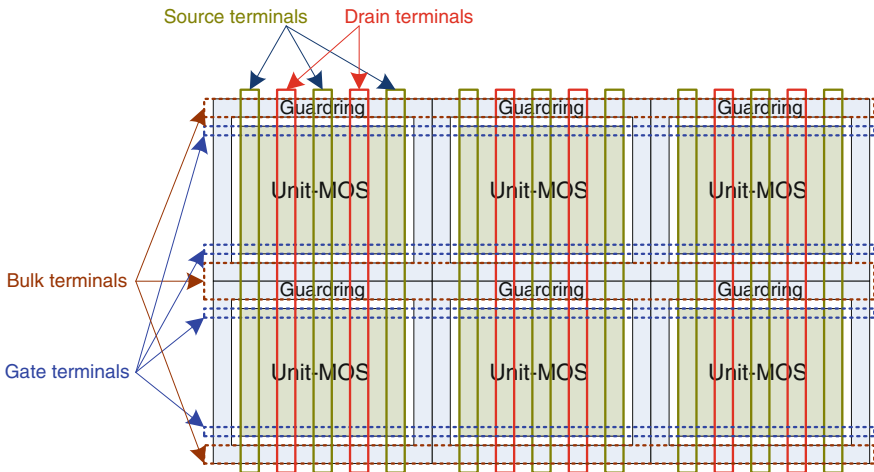


Fig. 6.7 Aggregated MOS layout is an array of unit-MOS

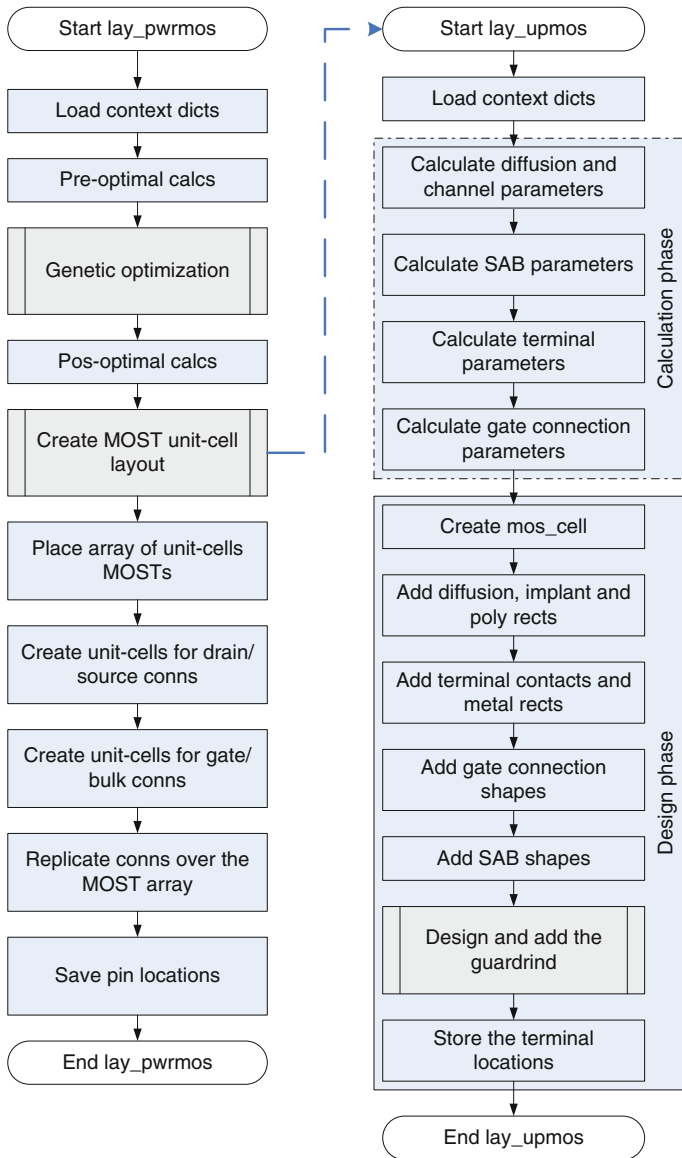
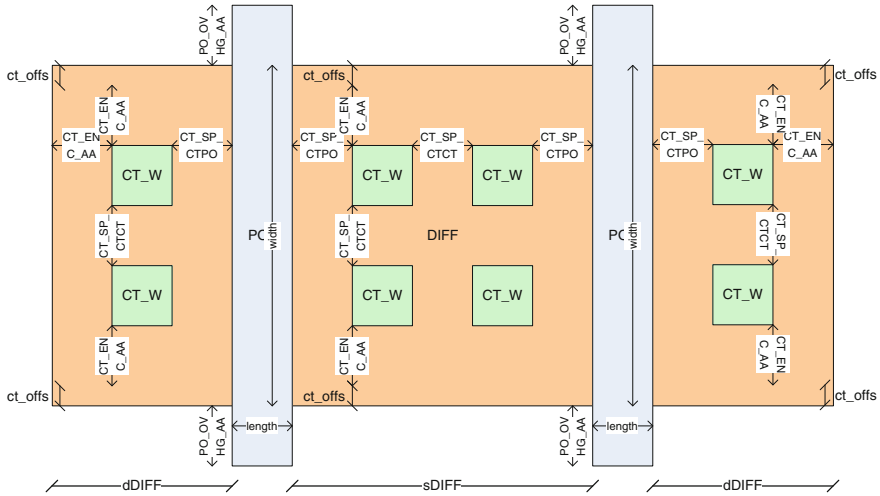


Fig. 6.8 Aggregated MOS design flow

The design of a unit-MOS cell (*lay\_upmos*) is sequential, all necessary rules are loaded, and thereafter, starting from a reference point, a bunch of calculations are made to produce the coordinates of aggregated and basic shapes. The construction process repeats itself for every other layer, starts with a calculation phase and ends



**Fig. 6.9** Example of a unit-MOS internal layout

with a design phase. This simple process is similar to methods employed in *pcell* generation.

To illustrate an example, Fig. 6.9 shows a basic unit transistor with two fingers, one contact column at source regions and two parallel contact columns at the drain region. In order to calculate the necessary parameters of the diffusion shape (a rectangle), one must load several contact-related rules (depicted in the same figure) and then make the following calculations:

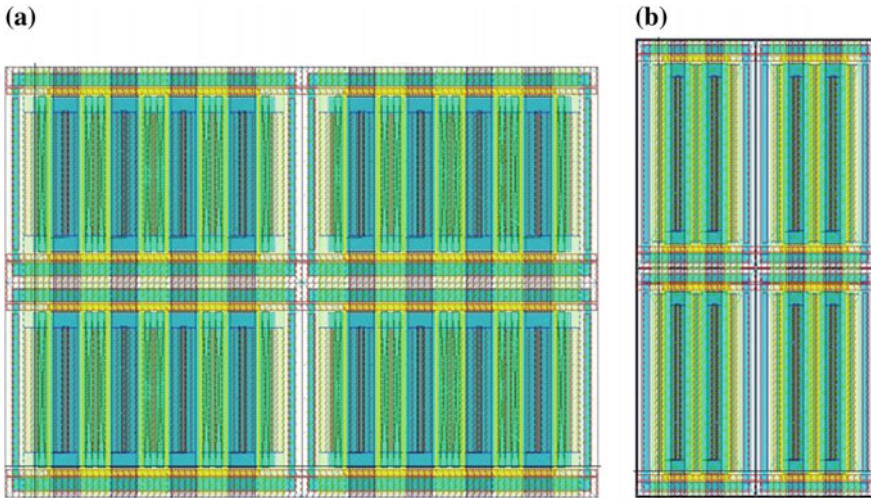
- Drain diffusion width:  $Ddiff\_w = CT\_ENC\_AA + CT\_W + CT\_SP\_CTPO$
- Source diffusion width:  $Sdiff\_w = 2 \cdot CT\_W + CT\_SP\_CTCT + 2 \cdot CT\_SP\_CTPO$
- Total diffusion width:  $Tdiff\_w = 2 \cdot Sdiff\_w + Ddiff\_w + 2 \cdot length$
- Total diffusion length:  $Tdiff\_l = width$

If the coordinates of the bottom-left corner are  $(orig\_xx, orig\_yy)$ , then the rectangle object that defines the diffusion shape is built by the following 2 vertices:

$$(orig\_xx, orig\_yy) \rightarrow (orig\_xx + Tdiff\_w, orig\_yy + Tdiff\_l)$$

Further, if we consider SAB extensions, the necessary rule list increases and the equations also suffer a proportional increase in complexity.

Figure 6.10 depicts two small aggregated MOSFETs created by the tool, having terminal extensions with SAB, one N-channel and another P-channel. One of them, the NMOS (Fig. 6.10a), have SAB regions on both junctions.



**Fig. 6.10** **a** Aggregated NMOS having terminals with SAB. **b** Aggregated PMOS having only drain with SAB

## 6.7 Optimization Procedures

This design framework uses two different random search optimization algorithms: the SA and the GA. The first is very simple to code and operate and has been the optimization method of choice on the vast majority of floorplanning and geometry placement problems. The proposed framework uses the SA for circuit-level floorplanning. The GA is more complex to code and tune, but is becoming accepted and used on a large scale of different problems and engineering areas, because of its flexibility and superior capability of finding solutions in hard problems. The authors choose it for optimizing single device layouts.

### 6.7.1 Optimization for Circuit Floorplanning—Simulated Annealing

The SA mimics a phenomenon in nature—the thermal annealing of solids—and applies the same basic principle to optimize a given system. Annealing refers to heating a solid and then cooling it slowly, causing atoms to assume a nearly globally minimum energy state. The SA simulates a small random displacement of an atom that results in a change in energy. If the change in energy is negative, the energy state of the new configuration is lower and the new configuration is accepted. If the change in energy is positive, the new configuration has a higher

energy state; however, it may still be accepted according to the Boltzmann probability factor:

$$P = e^{(-\Delta E/k_b T)} \quad (6.7)$$

where  $\Delta E$  is the difference between energy values,  $k_b$  is the Boltzmann constant, and  $T$  the current temperature. The probability of acceptance is proportional to temperature, but as the solid cools, the probability gets smaller and inversely proportional to the difference of energy values.

For the floorplanning problem (or any other optimization problem), an analogy is made between SA energy and a cost function value. The design is started at a high “temperature,” where it potentially has a high cost value we want to minimize. Random perturbations are then made to the design, and the cost function is evaluated. If the cost value becomes lower, the design solution is updated by the current design, but if it is higher, it may still be accepted according the probability given by the Boltzmann factor (6.7). This allows the algorithm to escape local minima and continue to explore the design space for a minimum global cost value.

The cost function is a linear combination of the total circuit area, total wire length, and temperature spread across the layout area:

$$\text{cost} = P_A \cdot \sum_j A_j + P_L \cdot \sum_j L_j + P_T \cdot D_T \quad (6.8)$$

$P_A$ ,  $P_L$ , and  $P_T$  are the weighing coefficients for total area, wire length, and temperature diffusion, respectively.

The wire-length metric is an index used to estimate the total wiring required to route the power nets within the floorplan. There are several possible indexes, the half perimeter wire-length (HPWL), the Steiner-tree wire-length (STWL), the minimum spanning tree wire-length (MSTWL), the complete graph wire-length (CGWL), the minimum chain wire-length (MCWL), etc.

This work uses the HPWL metric as this is one of the simpler, easily to code and faster. The HPWL estimates a net wire-length by creating a bounding box around all the pins to the routed and then taking the half perimeter of this box as the estimation value.

### 6.7.2 Optimization for MOS Device Layout—Genetic Algorithm

The GA belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. The GA uses a stochastic search optimization method, which enables simultaneous local and global search

capability, either for continuous or discrete kind of problems, and do not require continuous, convex, or differentiable objective functions. This kind of optimization strategy also avoids problem formalization (of classic methods) and is easier to use and modify. The GA can also model multiobjective, multiconstraints, and multimodal nonlinear problems, making it one of best choices, even when used only on a simple single-objective problem as this one.

In a GA, a population of variables (called chromosomes or the genotype of the genome) encodes candidate solutions (called individuals, creatures, or phenotypes) in an optimization problem and evolves them toward better solutions.

The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated and multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and occasionally randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

The GA algorithm flowchart is depicted in Fig. 6.11.

The GA requires a genetic representation of the solution domain; in this case, it can be a segment of the object parameters, specifically:

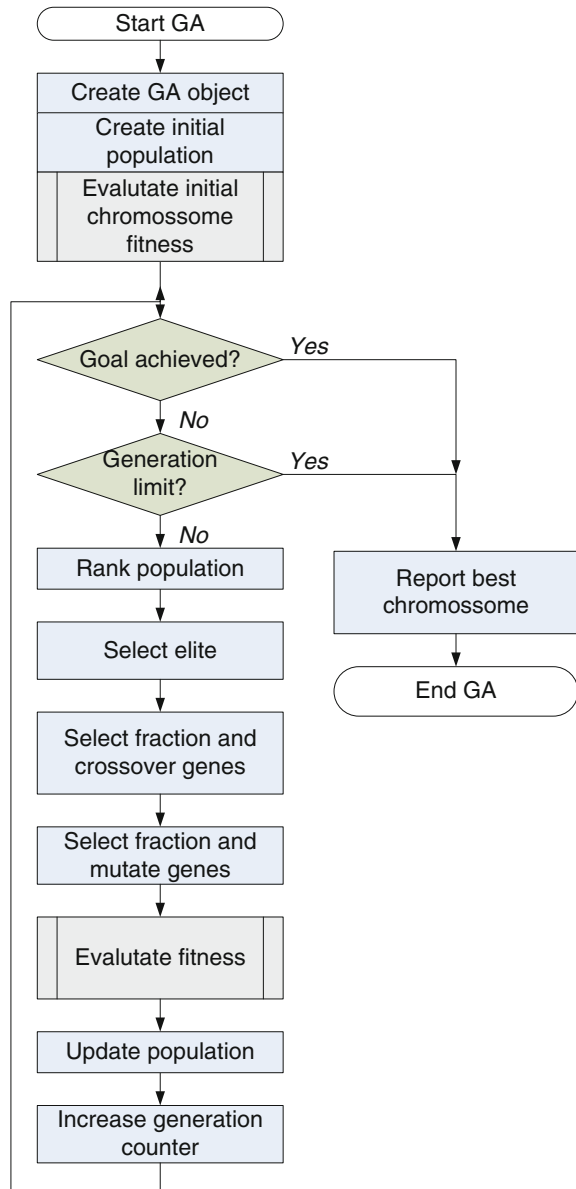
- Number of unit-cell fingers;
- Unit-cell multiplier in the *XX* direction;
- Unit-cell multiplier in the *YY* direction;
- Number of parallel column contacts at the drain and source.

The GA also requires a fitness function to score possible solutions accordingly to results from several tests. Key layout characteristics are calculated for each individual of the population of possible solutions and therefore tested against:

- ESD minimum/maximum finger constraints;
- Current density limit at metal fingers;
- DRC maximum active area separation to pick up constraint;
- Form factor constraints;

Additionally to the constraint tests, the fitness function also calculates the projected MOS area for each individual of the population. This value is used to rank all the possible solutions within the population; however, if any constraint tests is violated, the respective individual is severely penalized and potentially discarded from the selection poll.

**Fig. 6.11** Genetic optimization flow



## 6.8 Results

### 6.8.1 Half-Bridge Class-D Output Stage

The tool is demonstrated in a design of a half-bridge power stage for a Class-D amplifier and compared with a reference manual design at an 180 nm process. The target MOS characteristics are stated in Table 6.4.

The transistors include SAB extension only at the drain, by the minimum required by ESD rules. The gates are connected on both sides by a minimum of 2 parallel line contacts. Each transistor has drain and source terminal connections from Metal-1 to Metal-5 and must pass EM current density limits for a maximum of 1 A at an expected maximum temperature of 100 °C, and warrant safe operation during 10 years. The reference design took 1 min week to layout on an 180-nm CMOS process and has the metrics stated in Table 6.5.

Using these results as reference, 2 automatic designs were generated with slight different form factors on the same 180-nm process and registered as designs #1 and #2. Yet another design was generated in a similar 130 nm process and registered as design #3. Automatic designs took around 2 min to be generated and characterized, and all use the same target characteristics. Metrics are compared with the manual reference in Table 6.6, by Eq. 6.9:

$$\text{Metric comparison} = \left( \frac{\text{Design Metric}}{\text{Manual Design Reference}} - 1 \right) \times 100 \% \quad (6.9)$$

Negative values represent a reduction of the respective metric, while positive values represent an increase. The literature reference [20] designed in a 250 nm process is also compared.

**Table 6.4** Target MOS characteristics

Characteristic	PMOS	NMOS	Unit
$R_{on}$	0.30	0.30	ohm
$I_{max}$	1	1	A
$T_{max}$	100	100	°C
$W$	18,900	7560	μm
$L$	0.35	0.35	μm
Metal stack	1–5	1–5	–

**Table 6.5** Reference layout metrics

Metric	PMOS	NMOS	Unit
Dynamic loss	1.04	1.02	mW
$R_{on}$	0.42	0.33	ohm
Area	84,420	42,076	μm <sup>2</sup>



**Table 6.6** MOS layout comparison

	Design	PMOS (%)	NMOS (%)
Dynamic loss @2 MHz	Design #1 at 180 nm	-1.06	-13.23
	Design #2 at 180 nm	-1.16	-13.80
	Design #3 at 130 nm	-3.19	-16.30
	Reference [19] at 250 nm	<i>810.39<sup>a</sup></i>	<i>428.06<sup>a</sup></i>
$R_{on}$	Design #1 at 180 nm	-23.47	-10.21
	Design #2 at 180 nm	-24.65	-10.63
	Design #3 at 130 nm	-19.04	-12.79
	Reference [19] at 250 nm	2.33	-32.82
Area	Design #1 at 180 nm	12.13	-0.13
	Design #2 at 180 nm	-0.16	-6.45
	Design #3 at 130 nm	-24.63	-23.11
	Reference [19] at 250 nm	<i>329.46<sup>b</sup></i>	<i>356.15<sup>b</sup></i>

<sup>a</sup>Values from gate charge graph equated in same conditions as reference design

<sup>b</sup>Values from micrograph and scale

Italic value shows a worst case

## 6.8.2 H-Bridge Class-D Output Stage

The tool is also demonstrated for a full H-bridge Class-D output stage containing 4 power transistors and 2 gate driver circuits. Each transistor is divided in 2 equal slices, which creates a problem with 8 power devices and 2 additional shapes to be placed in the implantation area. The target characteristics are listed in Table 6.7.

The tool was deployed in a bulk CMOS 180 nm process. At the first phase, the slices of the power devices were created and electrically characterized for  $R_{on}$  and dynamic power losses. The bounding boxes of those resulting transistors and pin positions were then used on the second phase. The extracted parasites were also reused to over-estimate power losses in normal operation and allow thermal evaluations. Each power device was set to dissipate 10 times the value measured at the dynamic loss simulation, and the gate driving circuits were set to dissipate 4 mW in a worst-case scenario. The transistors include SAB extension at the drain side and

**Table 6.7** Target characteristics

Characteristic	PMOS	NMOS	Unit
$R_{on}$	0.30	0.30	ohm
$I_{max}$	0.75	0.75	A
$T_{max}$	125	125	°C
$W$	18,900	7560	$\mu\text{m}$
$L$	0.35	0.35	$\mu\text{m}$
Device slices	2	2	-
Metal stack	1-5	1-5	-
Driver circuit area	1400	1400	$\mu\text{m}^2$
Power loss at drivers	4	4	mW

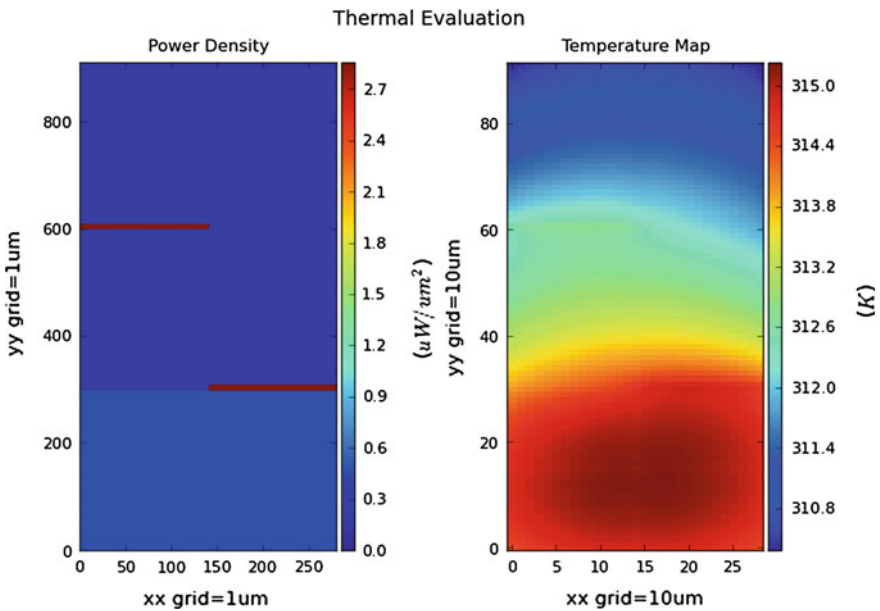
**Table 6.8** Power stage layout characteristics

Characteristic	PMOS	NMOS	Unit
Dynamic loss	0.9425	0.747	mW
$R_{on}$	0.4049	0.393	ohm
Device area	82,942	39,678	$\mu\text{m}^2$
Total area	248,040		$\mu\text{m}^2$
Area overhead	0		%
Maximum junction temperature	315.2		K
Temperature spread	4.9		K

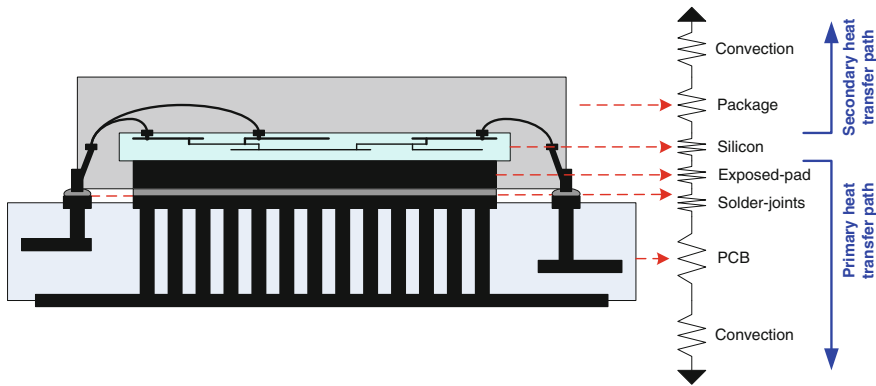
must pass EM current density limits for a maximum of 0.75 A at a junction temperature of 125 °C.

The main results are listed in Table 6.8. The best floorplan solution was obtained by the Polish method, achieving zero area overhead (floorplan area/summation of all element areas – 100 %) and zero white space.

Using an external FEM thermal simulator, it was possible to calculate the maximum hot-spot temperature, temperature spread over the implantation area, and the temperature map—Fig. 6.12. The magnitude of these outputs will depend greatly on heat conduction coefficients of the process and heat transfer coefficients of package and heat sink.



**Fig. 6.12** Thermal evaluation of the H-bridge—power density and temperature maps



**Fig. 6.13** Cross-sectional view of a QFN package soldered to a PCB, and simplified thermal model

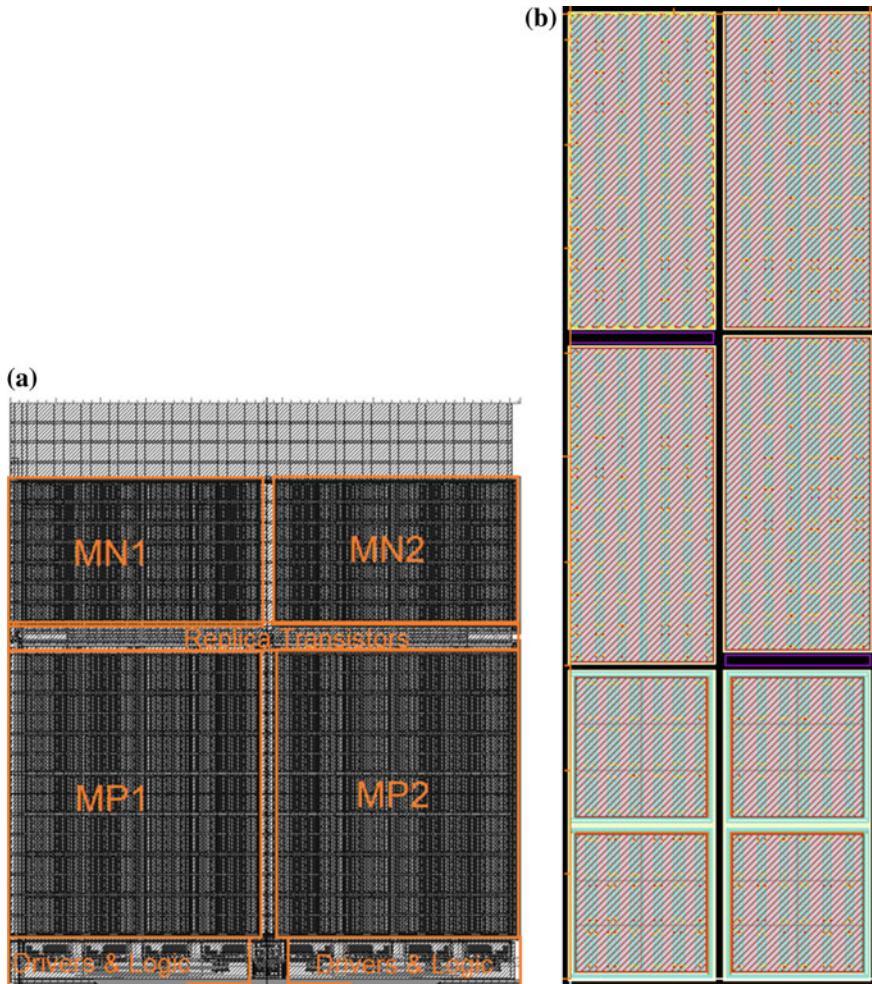
Figure 6.13 illustrates a chip model with a traditional QFN package, soldered to a PCB, and a simplified thermal resistance model network. This model allows the designer to find absolute values of temperature at the chip junction level. In the example, the primary heat transfer is the substrate to PCB path via exposed pad, and the main external parameters that affect temperature value are thermal conductivity of solder-joints and copper traces, and copper area connected to the exposed pad. Board-level thermal loading (dissipation of other PCB components) and air velocity on PCB and components are also major parameters.

The maximum junction temperature is expected to be 315 K and the spread 4.9 K, using effective heat transfer coefficients of  $4 \times 10^5$  and  $1 \times 10^5$  on the primary and secondary paths, respectively.

Figure 6.14b shows the resulting circuit-level floorplan. The total implantation area is  $0.248 \text{ mm}^2$ . The same specifications were pursued at a manual design in the same process (Fig. 6.14a) and required  $0.344 \text{ mm}^2$  of area. The manual design took 1.5 min weeks to design and validate, while the automatic one took 15 min. The automatic solution achieves savings of 28 % on area and speeds up the design flow by orders of magnitude.

### 6.8.3 Floorplan Benchmarks

The floorplanning capability of this tool was further verified on a set of hard MCNC benchmark problems. The listed problems have a limited number of blocks, below 12 elements, but the number of nets can be higher than a hundred. Table 6.9 states area overhead (circuit area/sum of device areas - 100 %) and average convergence time in seconds, for the three concurrent floorplan representation algorithms.



**Fig. 6.14** Layout of H-bridge. **a** Manual reference design. **b** Automated design

It is impossible to recognize a preferable algorithm, while SP returns better solutions for the *apte*, it is clearly the worst on the *xerox* problem. The postfix representation takes advantage of being the fastest one and can iterate much more often than the others, which compensates from being inherently less efficient in terms of area utilization.

An arbitrary problem with 12 devices and 14 nets, called *default*, was also floorplanned, this time enabling and disabling the thermal evaluation. This problem systematically returned the best balanced results with the B\*-tree representation. The results are listed in Table 6.10. When disabling the thermal evaluation, the tool speeds up by 36.7 % and improves area utilization from 11.34 to 9.52 % area

**Table 6.9** Floorplan benchmarks

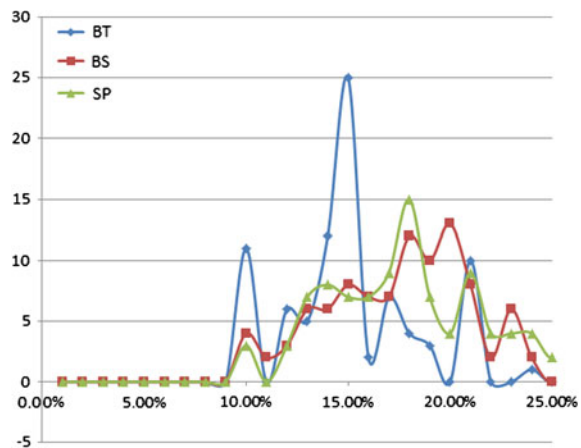
Benchmark	Metric	B-tree	B*-tree	SP
apte	Area overhead	2.15 %	2.08 %	0.78 %
	Avg. time (s)	3.57	4.68	3.53
hp	Area overhead	5.15 %	6.04 %	5.87 %
	Avg. time (s)	4.99	6.44	6.19
xerox	Area overhead	5.35 %	5.35 %	13.48 %
	Avg. time (s)	3.87	5.64	4.37

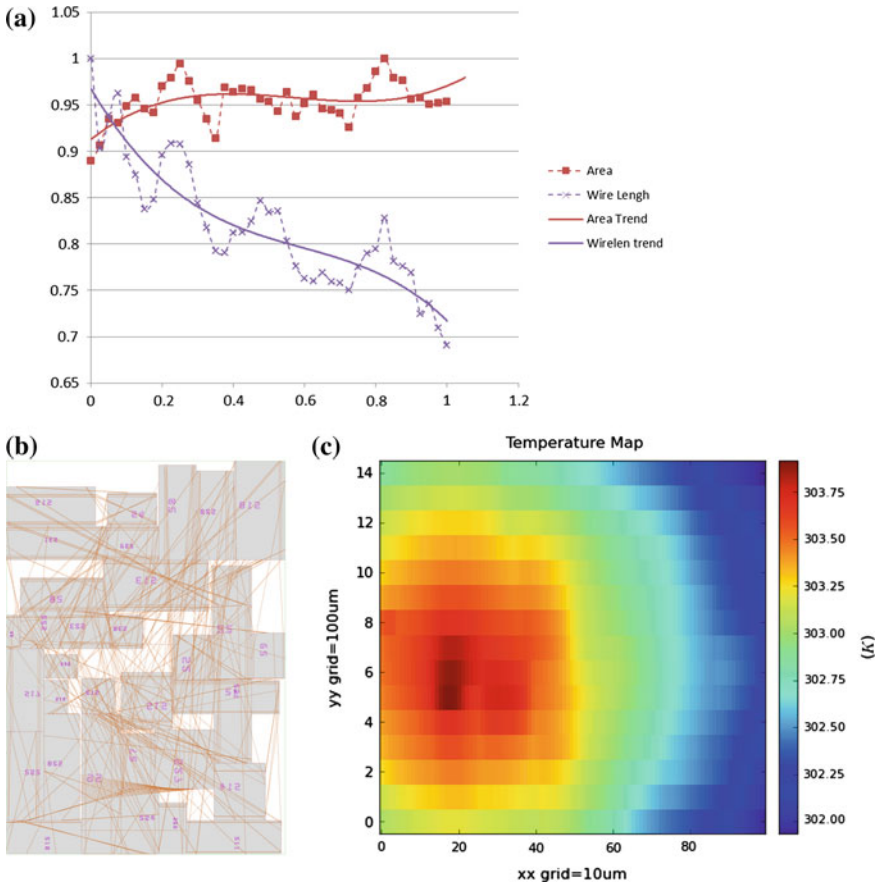
**Table 6.10** Default floorplan example

Metric	B-tree	B*-tree
	Including thermal cost	Excluding thermal cost
Area overhead (%)	9.52 %	11.34 %
Wire-length ( $\mu\text{m}$ )	305.00	224.00
Max. temp. (K)	340.40	342.30
Temp. spread (K)	12.80	14.80
Avg. time per cycle ( $\mu\text{s}$ )	870	550

overhead. However, the maximum temperature and spread increases by  $2^\circ$  over the implantation area.

Conducting a systematic run of the same *default* problem over 100 times with fixed number of iterations, and creating the histogram illustrated in Fig. 6.15a, is possible to see that while the B\*-tree (BS) and SP representations have a more continuous cost function than the Polish representation (BT). The Polish representation exhibits tall hills and deep valleys, having higher frequencies at specific cost

**Fig. 6.15** Histogram of area overhead cost from 100 solutions for the *default* test problem



**Fig. 6.16** The *ami33* benchmark problem. **a** Wire coefficient sweep results. **b** Floorplan layout, including air-wires. **c** Thermal map

values and zero frequency at other cost intervals. This behavior is a consequence of the smaller and more discretized design space of the Polish representation.

Yet another systematic run was conducted for a larger problem, the *ami33* of the MCNC benchmarks. This time the wire-length coefficient  $P_L$  of Eq. (6.8) was swiped from zero to a normalizing value where the wire cost is equal to 30 % of the area cost. The results are depicted in Fig. 6.16, and a downward trend on the wire-length cost is clearly seen as the coefficient approaches one. The upward trend in area cost is not as explicit, but exists and is to be expected since some area must be sacrificed in order to reduce wire-length.

One of the possible solutions of Fig. 6.16a is depicted in Fig. 6.16b and the corresponding thermal map in Fig. 6.16c. It is clear that the three conflicting goals are creating a considerable white space in the floorplan.

Besides increasing the number of iterations, to improve the quality of solutions, it would be advisable to change form factors and pin relative positions. Floorplan optimizations with soft macros are beyond the scope of this work, because here macros are power devices. Power transistors cannot have independent form factors which can easily generate multiple design violations as ESD and DFM constraints can become difficult to meet.

## 6.9 Conclusions

An automatic tool for layout generation of integrated MOSFET power stages in bulk CMOS was demonstrated. The tool starts by generating optimal power device layouts in isolation and at a second phase floorplans those devices in power stages.

The generated devices were compared with reference manual designs, and the results obtained are superior: lower resistance and dynamic power losses, while attaining or saving silicon area. Generated power devices are automatically compliant with DRC, DFM, and ESD rule sets, and technology independent. Several designs were generated in 2 process nodes: 180 and 130 nm CMOS nodes.

The tool generates optimized floorplans of power stages for area and total wire-length minimization. Circuit-level floorplans can be thermal-aware solutions to smooth temperature distribution profiles and increase reliability and performance. An example was demonstrated using total thermal diffusion as an evaluation index of junction temperature spread. Parasitic metrics, power dissipation, and thermal maps are automatically generated to completely characterize generated solutions.

All created designs were directly exported into GDSII format, which allows complete independence from any IC design platform and permits that the exported. *gds* files can be automatically or semi-automatic validated by industry accepted sign-off tools. Automatic physical validation, parasitic extraction, and post-layout electrical characterization were performed on all power stage examples.

The layout design and verification flow was speed up by several orders of magnitude.

## References

1. Liew, B.K., Cheung, N.W., Hu, C.: Effects of self-heating on integrated circuit metallization lifetimes. In: IEDM Technical Digest., Washington, pp. 323–326 (1989)
2. Semenov, O., Vassighi, A., Sachdev, M.: Impact of Self-Heating Effect on Long-Term Reliability and Performance Degradation in CMOS Circuits. *IEEE Trans. Device Mater. Reliab.* **6**(1), 17–27 (2006)
3. Tam, W.C., Blanton, S.: To DFM or not to DFM? In: IEEE Proceedings of the 48th Design Automation Conference, pp. 65–70, June 2011

4. Tien, L.C., Tang, J.J., Chang, M.C.: An automatic layout generator for I/O cells. In: Proceedings of the 5th International Workshop on System-on-Chip for Real-Time Applications, pp. 295–300, July 2005
5. Ming, C., Na, B.: An efficient and flexible embedded memory IP compiler. In: Proceedings of International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 268–273, Oct 2012
6. Kelly, M., Servais, G., Diep, T., Lin, D., Twerefour, S., Shah, G.: A comparison of electrostatic discharge models and failure signatures for CMOS integrated circuit devices. In: Proceedings of the Electrical Overstress/Electrostatic Discharge Symposium, pp. 175–185, Sept 1995
7. Franell, E., Druen, S., Gossner, H., Schmitt-Landsiedel, D.: ESD full chip simulation: HBM and CDM requirements and simulation approach. *Adv. Radio Sci.* **6**(10), 245–251 (2008)
8. Suman, B., Kumar, P.: A survey of simulated annealing as a tool for single and multiobjective optimization. *J. Oper. Res. Soc.* **57**, 1143–1160 (2006)
9. Alpert, C.J., Mehta, D.P., Sapatnekar, S.S. (eds.): Handbook of algorithms for physical automation. CRC Press, Boca Raton. ISBN:10: 0849372429, ISBN:13: 978-0849372421 (2009)
10. Martins, R., Lourenco, N., Horta, N.: LAYGEN II—automatic layout generation of analog integrated circuits. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **32**(11), 1641–1654 (2013)
11. Lall, P.: Tutorial: temperature as an input to microelectronics—reliability models. *IEEE Trans. Reliab.* **45**(1), 3–9 (1996)
12. Pedram, M., Nazarian, S.: Thermal modeling, analysis, and management in VLSI circuits: principles and methods. *Proc. IEEE* **94**(8), 1487–1501 (2006)
13. Bechtold, T., Rudnyi, E., Korvink, J.: Dynamic electro-thermal simulation of microsystems—a review. *J. Micromech. Microeng.* **15**(11), R17–R31 (2005)
14. Batty, W., Christoffersen, C., Panks, A., David, S., Snowden, C., Steer, M.: Electrothermal CAD of power devices and circuits with fully physical time-dependent compact thermal modeling of complex nonlinear 3-d Systems. *IEEE Trans. Compon. Packag. Technol.* **24**(4), 566–590 (2001)
15. Han, Y., Koren, I.: Simulated annealing based temperature aware floorplanning. *J Low Power Electron.* **3**(2), 1–15 (2007)
16. Ardestani, E., Ziabari, A., Shakouri, A., Renau, J.: Enabling power density and thermal-aware floorplanning. In: Proceeding of Semiconductor Thermal Measurement and Management Symposium, pp. 302–307, Mar 2012
17. Song, T., Sturcken, N., Athikulwongse, K., Shepard, K., Lim, S.K.: Thermal analysis and optimization of 2.5-D integrated voltage regulator. In: IEEE 21st Conference on Electrical Performance of Electronic Packaging and Systems, pp. 25–28 (2012)
18. Ning, P., Wang, F., Ngo, K.D.T.: Automatic layout design for power module. *IEEE Trans. Power Electron.* 481–487 (2013)
19. Logan, S., Guthaus, M.R.: Fast thermal-aware floorplanning using white-space optimization. 17th IFIP International Conference on Very Large Scale Integration, pp. 65–70 (2009)
20. Ng, W.T., Chang, M., Yoo, A., Langer, J., Hedquist, T., Schweiss, H.: High speed CMOS output stage for integrated DC-DC converters. In: Proceedings of 9th International Conference on Solid-State and Integrated-Circuit Technology, pp. 1909–1912, Oct 2008



# Chapter 7

## Optimizing Model Precision in High Temperatures for Efficient Analog and Mixed-Signal Circuit Design Using Modern Behavioral Modeling Technique: An Industrial Case Study

Sahbi Baccar, Timothée Levi, Dominique Dallet  
and François Barbara

**Abstract** This chapter deals with the description of a modeling methodology dedicated to simulation of AMS circuits in high temperatures (HT). A behavioral model of an op-amp is developed using VHDL-AMS in order to remedy the inaccuracy of the SPICE model. The precision of the model simulation in HT was improved thanks to the VHDL-AMS model. Almost all known op-amp parameters were inserted into the model, which was developed manually. Future work can automate the generation of such a behavioral model to describe the interdependency between different parameters. This is possible by using modern computational intelligence techniques, such as genetic algorithms, or other techniques such as Petri nets or model order reduction.

---

S. Baccar (✉)  
IRSEEM Laboratory-ESIGELEC, Rouen, France  
e-mail: sahabi.baccar.fr@ieee.org; sahib.baccar@gmail.com

T. Levi  
IMS-Laboratory, University of Bordeaux, Bordeaux, France  
e-mail: timothee.levi@ims-bordeaux.fr

D. Dallet  
IMS-Laboratory, Bordeaux INP, Bordeaux, France  
e-mail: dominique.dallet@ims-bordeaux.fr

F. Barbara  
Schlumberger Etudes et Productions, Schlumberger, France  
e-mail: barbara@clamart.oilfield.slb.com

## 7.1 Introduction

### 7.1.1 High-Temperature Electronics (HTE) Circuits

In the last decades, there was a considerable growth of many industries requiring electronic circuits operating in high temperature such as oil, aerospace, and automotive. In such applications, circuits are integrated in industrial systems that measure either physical parameters (temperature, pressure, speed, position) or specific parameters of the application (geological parameters, mechanical parameters, etc.) [1, 2]. The value of 125 °C was chosen as the reference temperature from which every circuit operating up to it is considered as belonging to the family of high-temperature electronics (HTE). Conventional electronics are defined as circuits which operate in a temperature ranging from  $-40$  to 125 °C (Fig. 7.1).

Figure 7.1 illustrates and compares HTE circuits' family to some other circuits' families with their defined temperature ranges. The operating temperature of some HT applications can reach some values that are much greater than 125 °C. The operating temperature depends on the industrial application and its specificities [3–6]. Appearing of new extreme conditions not only has defined a new family of circuits but has mainly motivated industrial and scientific researches in this area [7]. As a result, the arrival of first circuits dedicated to HT was announced in the end of the 1990s. In spite of the availability of such circuits, many industrials believe that maturity age of dedicated HT circuits is not yet reached. Moreover, integrating of such high technology circuits with novel materials that resist to HT effects is not immediately possible. The major reason is the fact that dedicated HT circuits are still expensive and their usage in great number in an industrial chain raises the production cost. Actually, there are still few companies that manufacture such circuits. Besides, there is a lack of mastering the new technology of these circuits.

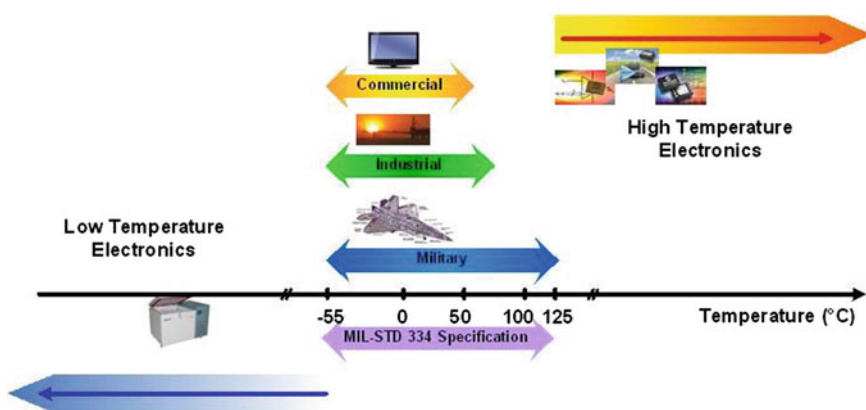


Fig. 7.1 High-temperature electronics family compared to other circuits families

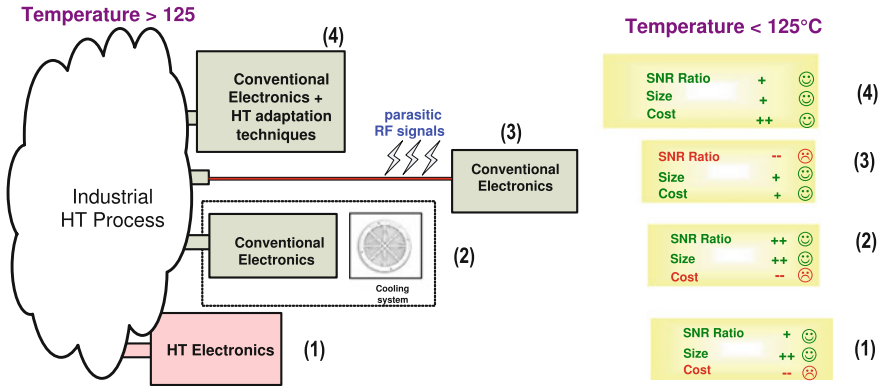


Fig. 7.2 Different possibilities for interfacing HT circuits to an industrial HT process

Finally, HT circuits cover often only a limited number of electronic functionalities [8]. Consequently, designers prefer until today to use conventional electronics. They add to them appropriate adaptation circuits to compensate the undesirable effects of the increasing of temperature. Another solution consists in adding a cooling system in order to obtain an operating temperature that belongs to typical temperature ranges of conventional electronics. In Fig. 7.2, we have summarized different possibilities of using HT circuits near an industrial process.

It is clear that scenario (1) leads to a better value of SNR ratio compared to the other possibilities by putting directly dedicated HT circuits near the industrial process. In addition, the size of the circuitry is smaller than that of other alternatives. However, the major drawback of the alternative (1) is the huge cost. In alternative (3), conventional electronics are put in a temperature which is lower than 125 °C. The industrial process and the circuitry are linked with wire, and this is the main disadvantage of such solution. Actually, this wire is exposed to many severe sources of interference. Consequently, parasitic signals will contribute to decrease the SNR value. It is possible to overcome this problem by moving the circuitry close to the industrial process as shown in alternative 2. However, a cooling system must be added, and this will increase the cost and the size. That is why using conventional electronics with appropriate adaptation circuits (solution 4) that eliminate or at least reduce HT effect seems to be an attractive solution. It is a good trade-off between the three criterions: the cost, the size, and the fact of acquiring properly the signal. However, this alternative is not an ideal as there is really an accuracy problem when using industrial circuits' models for HT simulation. More details about this issue are given in the second part of this section.

### 7.1.2 *The Issue of Using HTE Models*

In order to conceive precise instrumentation and measurement systems, the performance and behaviors of the circuitry part should be predicted with efficient simulation tools and should be tested in a second step. It is well known that a simulation is based on a set of precise models that describe the behavior and the performance of the studied circuits in specific conditions. Besides the models of circuits, simulation of electronic circuits relies on a specific program that was developed by exploring some known formula in electronics (mainly Kirchhoff laws).

SPICE program is especially considered as the most famous program that has allowed the circuits' design to be automated. Since the development of its first version in Berkley University, it was used by academics, industrials, researchers, etc. Many versions of the program were developed by many EDA companies. In parallel, the first models of analog circuits and then mixed-signal circuits have also started to be published. Many transistor families with different technologies were modeled according to the SPICE format. Using these models has facilitated the job of analog and mixed-signal (AMS) designers. Using SPICE program and SPICE models becomes so common that publishing SPICE models of manufactured circuits becomes almost a requirement for semiconductors companies.

In our case, there is a really a problem when using SPICE model if alternative (4) in Fig. 7.2 is adopted. Actually, if this alternative will avoid the using of expensive dedicated HT circuits with a good quality of the signal and with a reasonable size, the accuracy of the used models is not guaranteed. The conventional electronics and the adaptation circuits are commercial circuits having specific SPICE models that are offered by their constructors. Models are supposed to predict the performances of the circuit in a temperature that does not exceed 125 °C. As we are interested here in making the circuits operating in a greater temperature than 125 °C, we have to test first the validity of the used SPICE models in HT.

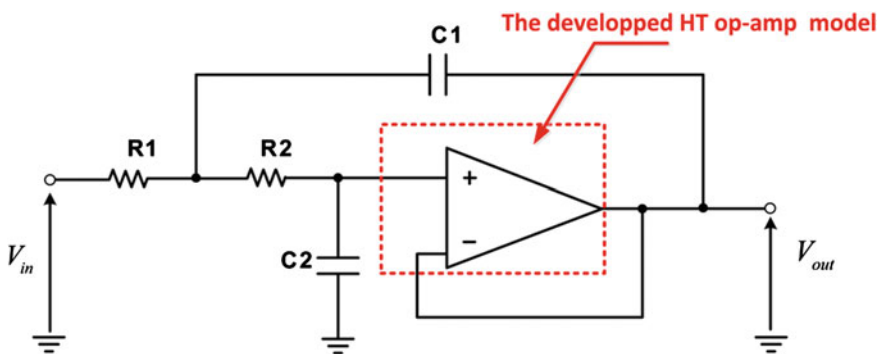
If the industrial SPICE model will be inaccurate in HT, we will be constrained to choose among two possibilities: either extending the SPICE model validity in HT or developing a new model with another modeling methodology that will fit better the industrial application context. This will be the main goal of this work. In order to reach it, we will study the feasibility of each choice. We will select the possible modeling approaches with a comparative study of the known modeling approaches of AMS circuits. Once the modeling approach is chosen, we select the modeling language, and we test the model and validate it in HT. The chapter is organized as follows. Section 7.2 will be devoted to detailing the development of the modeling methodology in HT. We will study a specific device that is considered as a key component of the most of the instrumentation circuits: the op-amp. In Sect. 7.2.1, we will test first the accuracy of a SPICE model of a commercial op-amp in HT. Secondly, we will present the behavioral modeling approach that was chosen to remedy the inaccuracy of SPICE model in HT. The modeling methodology will be detailed, and the advantages of the behavioral modeling will be argued.

The simulation results that will validate the model in HT will be presented in Sect. 7.3. The improvement of the model precision will be evaluated. Finally, we conclude this work in Sect. 7.4 and give some prospects.

## 7.2 Modeling Methodology for Accurate HT Simulation

The goal of this part consists in developing a modeling methodology that will optimize the precision of the simulation results in HT. This study deals with AMS circuits on an instrumentation AFE. However, the study was limited to an op-amp as it is omnipresent in AMS circuits.

Op-amp is a really a key component of AMS circuits as it is present in almost all AMS circuits dedicated to instrumentation [9, 10]. Instead of developing each AMS circuit model separately in HT, it will be more practical to develop their model from an accurate op-amp model in HT by using an architectural description as illustrated in Fig. 7.3. The section starts by studying an industrial SPICE op-amp model in HT. We expect that such model will be inaccurate. We will develop an accurate one with an appropriate methodology that takes into account industrial constraints. Measurement and theoretical equations will validate such model. We will assess the inaccuracy of the considered SPICE model by comparing simulations results to measurements. A brief interpretation and analysis of this inaccuracy will be presented. At the end of the section, we will make a comparison between all different modeling approaches in order to choose the appropriate one for industrial constraints. We will show that behavioral modeling approach seems to be the most appropriate choice for such constraints.



**Fig. 7.3** Using a HT op-amp model for developing a HT analog filter model by using architectural description

### 7.2.1 Op-Amp Parameters Test in HT

The studied op-amp is a commercial reference. For confidentiality reasons, this reference will not be given. However, some values of the op-amp parameters were summarized in Table 7.1.

The values of these parameters are given for temperatures ranging from  $-40$  to  $85$  °C, a common-mode voltage that equals zero and a power supply voltage that equals the nominal value ( $V_s = \pm 15$  V). A campaign of measurements of each parameter was lead to enable to test accuracy of SPICE simulations and technical specification of the constructor.

Among mentioned parameters in Table 7.1, the offset voltage  $V_{os}$  was chosen to be simulated with SPICE model. This parameter is defined as the algebraic deviation of the linear zone of the op-amp characteristic. On other words, it is equal to the input voltage of the op-amp that will give an output voltage that equals zero. The offset voltage is caused by the dissymmetry and imperfection in the transistors of the op-amp input stage. The typical values of offset voltage vary from some microvolts to some millivolts [11]. They depend on the operating conditions such as humidity, circuit age, and mainly temperature. Tolerated values of offset voltage depend on the application; it can be some microvolts or even some decades of microvolts for a specific application, whereas it should not exceed 1 microvolt in other applications [12]. Offset voltage represents often a random variation aspect. For this reason, it is often presented by histograms in technical specification documents. It is considered by industrials as the most sensitive parameter to the temperature variation. Its sensitivity is evaluated by the coefficient “voltage Temperature Coefficient” (VTC):  $VTC = \Delta V_{os}/\Delta T$ , where  $\Delta V_{os}$  is the voltage offset variation, and  $\Delta T$  is temperature variation. The used unity of this coefficient in technical documents is  $\mu V/^\circ C$ . Thereby, the formula of VTC definition assumes a linear evolution of  $V_{os}$ . Measurement will show that this assumption is valid only until a specific temperature.

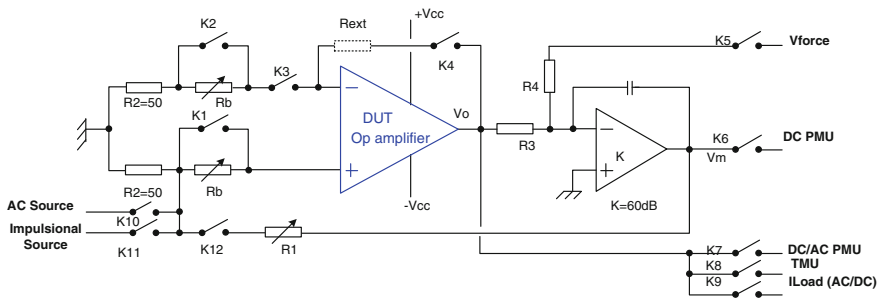
**Table 7.1** Characteristics of the studied op-amp as given by the constructor

Parameter	Symbol	Values			Unity
		Minimal	Typical	Maximal	
Offset voltage	$V_{os}$		180	650	$\mu V$
$V_{os}$ deviation	$\Delta V_{os}/\Delta T$		1	3	$\mu V/^\circ C$
Offset current	$I_{os}$		10	30	nA
Polarization current	$I_b$		55	150	nA
Common mode-rejection ratio	CMRR	82	101		dB
Power supply rejection ratio	PSRR	90	108		dB
Open-loop gain	$A_{ol}$	1000	3500		V/mV
Slew rate	SR	2	3		V/ $\mu S$
Gain bandwidth product	GBWP	3	5		MHz



**Fig. 7.4** Experimental apparatus for measurement and test of op-amp performances

The measurement apparatus is illustrated by Fig. 7.4. We can see the part with which a thermal flow is applied in order to heat the circuit so that its temperature can reach high values. The process of measurement is automated and commanded by a computer program. By following a step of 10 °C, measurements were achieved for temperature going from 20 to 220 °C. In order to avoid uncertainty errors, each measurement is repeated 10 times. To test the effect of the power supply and common-mode voltages on op-amp parameters, measurements are taken each time for a specific value of each of these two voltages. Almost all op-amp parameters were measured automatically. It was possible thanks to a specific test-bench circuit that is depicted in Fig. 7.5. This structure comprises two op-amps whose one is the



**Fig. 7.5** Generic experimental test-bench circuit

DUT. The second op-amp is ideal and is called “nulling op-amp” [13]. It is fed in its negative input by a loop that controls the tested op-amp. Depending on the tested parameter, some switches are in the “on” position and others in the “off” one. Furthermore, the suitable signal is generated depending on the measured parameter. It can have different types: an AC source, DC source, impulse signal, etc.

### 7.2.2 Simulation of SPICE Op-Amp Model in HT

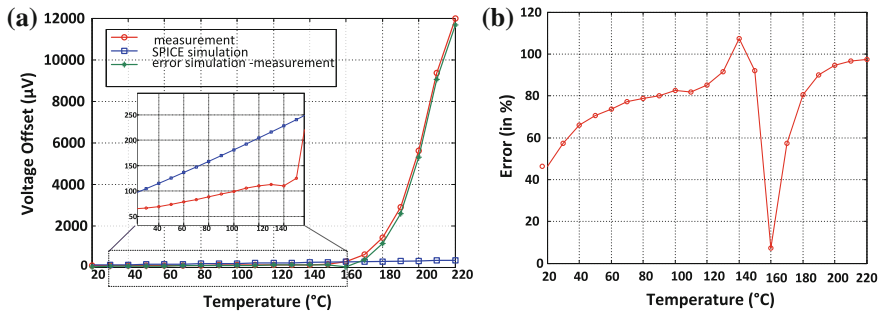
The temperature simulation in SPICE is based on a parametric analysis of temporal, DC or AC analysis. This means that SPICE program has the capacity to simulate the temperature of complex circuits, but models should be accurate in order to obtain precise simulation. The simulation of  $V_{os}$  is performed for each temperature separately. The circuit simulation is in an open loop in which a macromodel of the tested op-amp is connected to a variable DC voltage source by its positive input. A DC simulation with a parametric analysis of the temperature enables to plot the output voltage  $V_{out}$  as the function of the input voltage  $V_{in}$ . The obtained curve is the characteristic curve of the op-amp for the chosen temperature. Referring to the offset voltage definition, the simulated  $V_{os}$  will be the intersection point of the linear part of the characteristic curve with the input voltages axis. From such DC analysis output, we can read in the same time, the voltage offset, the open-loop gain, and positive and negative saturation voltages. Finally, we notice that, as measurement were done many times, the measured value is considered as the mean of all the measured values for each temperature.

The measured and simulated values of  $V_{os}$  are plotted for different temperatures in [20 °C, 220 °C]. The simulated values are weak compared to the measured ones. SPICE predicts a linear evolution of  $V_{os}$  in accordance with technical specifications. We make a zoom on both the measurement curve and the simulation curve to observe clearly the linear evolution of SPICE values. However, the measurement curve has two parts: the first part whose variation is quasi-linear between 20 and 140 °C and a second exponential part from 150 °C. The maximal simulated value corresponds to 220 °C, and it is equal to 323.34  $\mu\text{V}$ . The maximal measured value is 119988.2  $\mu\text{V}$  and occurs in the same temperature. We remind that the technical specification gives also the maximal and the typical values in [−40 °C, 85 °C]. They are 180 and 650  $\mu\text{V}$ , respectively (Table 7.1). The typical value is in good agreement with SPICE simulation that predicts values ranging from 94.84  $\mu\text{V}$  at 20 °C to 158.43  $\mu\text{V}$  at 80 °C. The linear part is characterized in the datasheet by the coefficient VTC having a typical value of 1  $\mu\text{V}/^\circ\text{C}$  and a maximal one of 3  $\mu\text{V}/^\circ\text{C}$ . The simulated value of VTC is the slope of the linear curve found by SPICE simulation in [20 °C; 80 °C]. It equals 1.16  $\mu\text{V}/^\circ\text{C}$ . In the linear part of the measurement curve and by using a fitting tool also, we have obtained the value of VTC in this part; it equals 0.47  $\mu\text{V}/^\circ\text{C}$ . This shows that the SPICE simulation is inaccurate even in the linear part of the offset voltage curve. Moreover, in the second part of the measurement curve, the evolution seems to be rather exponential,



**Table 7.2** Some numerical values of  $V_{os}$  simulation and measurement results of  $V_{os}$

Temperature (°C)	SPICE (μV)	Mean value of measurement (μV)	Error in absolute value (μV)	Error in percentage (%)
20	94.84	64.8	30.04	46.35
40	115.24	69.4	45.84	66.05
80	158.43	88.65	69.78	78.71
120	204.5	110.40	94.1	85.23
160	253.09	273.1	20.01	7.32
180	277.96	1427.75	1149.79	80.53
220	323.34	11988.2	11664.86	97.30



**Fig. 7.6** SPICE simulation of  $V_{os}$  (a) and the error in percentage (b) in [20 °C, 220 °C]

whereas it is always linear in the simulation curve. Actually between 160 and 220 °C, the value of the measured  $V_{os}$  almost doubles each 10 °C. This represents successive elements of a geometric progression that can be always approximated by an exponential law. Table 7.2 summarizes some values of measurement results, simulation outputs, and simulation errors. We note especially that the error is very huge in the HT region [120 °C, 220 °C]. This is also clear from Fig. 7.6b. Thus, this confirms the limitation of the tested SPICE model and invites us to develop a new modeling methodology to correct such imprecision. It will be possible to interpret and understand these SPICE simulations’ error sources if we analyze the structure of the SPICE model.

### 7.2.2.1 Analysis and Interpretation of the Source of the Offset Voltage Error

In order to understand better the offset voltage error in SPICE simulation, we review the structure of the SPICE op-amp model. The detailed structure of an industrial circuit cannot be given in a technical specification mainly because of confidential reasons and also the complexity of industrial circuits. Moreover, as



**Fig. 7.7** Declaration of one of the two transistors in the op-amp SPICE model

industrial circuits comprise an enormous number of transistors, giving their detailed structure in SPICE format will make simulation very slow. Generally, a much simpler circuit having almost the same performances is provided by the constructor. In our case, besides the op-amp model, an equivalent circuit containing 16 transistors was presented in the technical document. The op-amp SPICE model is simpler than this equivalent circuit since it contains only two transistors. However, there are more usual components such as resistors, capacitances, and diodes of currents. A part of the SPICE model is presented in Fig. 7.7, which illustrates the definition of one of the two transistors in the SPICE model with its parameters.

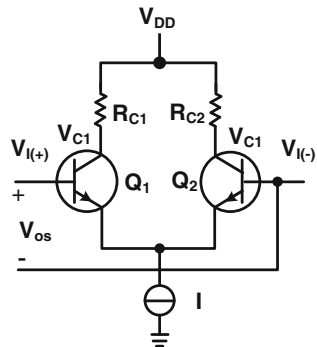
The two transistors are denoted Q1 and Q2. The nodes of each transistor are also declared (11, 2, and 13 for Q1). The models of the transistor Q1 is denoted with the same name as illustrated in Fig. 7.7b. This model is recognized by the program SPICE. It belongs to BSIM models that were developed by “BSIM Research Group” in the department “Electrical Engineering and Computer Sciences” of Berkeley University [14, 15]. IS, BF, XTB, XTI, and KF are the parameters of the model of each transistor. The values of the two transistor parameters are not identical which means that there is a dissymmetry in the part of the model containing the two transistors (called the input stage). We will detail the source of this dissymmetry by exploring Kirchhoff laws and characteristic equations of each transistor. Finally, only XTB and XTI are present in equations in which the temperature parameter is present. Temperature simulation of the op-amp with SPICE is based on these equations and parameter values.

$$BF(T) = BF \cdot \left( \frac{T}{T_{\text{nom}}} \right)^{XTB} \quad (7.1)$$

$$I_s(T) = I_s(T_{\text{nom}}) \cdot \left( \frac{T}{T_{\text{nom}}} \right)^{XTI} \cdot \exp \left[ \frac{E_g \cdot q(T_1 \cdot T_0)}{k \cdot (T_1 - T_0)} \right] \quad (7.2)$$

XTB and XTI are the unique responsible of the behavior of the op-amp over temperature in the SPICE model. Unfortunately, the parameter values are supposed only valid in the zone  $[-40 \text{ }^\circ\text{C}, 85 \text{ }^\circ\text{C}]$ . Reviewing them is a complex task as it requires a perfect knowledge of the transistor technology. It requires also a precise campaign measurement on a physical abstraction level. In an industrial context, this is not possible since the access to the manufactured circuit’s technology is only possible for the constructor. That is why another modeling alternative that will be more convenient for industrial constraints should be developed. The next part will

**Fig. 7.8** A macromodel on an input op-amp stage: a differential pair implemented with BJTs



be devoted to discussing all possible alternatives that can meet industrial constraints. Before that, we finish this part with the reviewing of the equations linking the offset voltage to op-amp input-stage parameters in order to interpret the error source in SPICE simulation of  $V_{os}$ .

Transistors Q1 and Q2 have a common emitter and a common collector also. RC1 and RC2, collector resistances of, respectively, Q1 and Q2 are connected to the node VDD ( Fig 7.8). The voltage offset results from the dissymmetry between the two transistors. Referring to [16],  $V_{os}$  has the expression given by Eq. (7.3), where  $T$  is temperature.  $A_1$ ,  $B_1$ ,  $A_2$ , and  $B_2$  expressions are given by Eqs. (7.4)–(7.7).  $q$  and  $k$  are, respectively, the electron charge constant and Boltzmann constant.

$$V_{os} = A_1 \exp(B_1 \cdot T) - A_2 \exp(B_2 \cdot T) \quad (7.3)$$

$$A_1 = I_{S1}/g_m = I_S(1 + \Delta I_S/2I_S)/g_m \quad (7.4)$$

$$A_2 = I_{S2}/g_m = I_S(1 - \Delta I_S/2I_S)/g_m \quad (7.5)$$

$$B_1 = V_{BE1} \cdot q/k \quad (7.6)$$

$$B_2 = V_{BE2} \cdot q/k. \quad (7.7)$$

Voltages  $V_{BE1}$  and  $V_{BE2}$  are, respectively, the base–emitter voltage of Q1 and Q2.  $I_{S1}$  and  $I_{S2}$  are, respectively, saturation currents of Q1 and Q2. The mean of these two currents is denoted  $I_S$ , and their difference is denoted  $\Delta I_S$ . SPICE linearizes any nonlinear expression including exponential functions by using specific methods of linearization [17]. As coefficients  $B_1 \cdot T$  and  $B_2 \cdot T$  keep small values, it is possible to linearize the exponential functions in Eq. (7.3) as shown in Eq. (7.8) by using Taylor series of exponential function.

$$V_{os} \equiv (A_1 - A_2) + (A_1 B_1 - A_2 B_2)T + \dots \quad (7.8)$$

However, depending on the values of  $B_1$  and  $B_2$  and if the temperature increases, the approximation of the exponential function in Eq. (7.8) is no longer valid. This is

the main error source in the simulation of the offset voltage in HT. Thus, a polynomial function should approximate more precisely the dependency of  $V_{OS}$  to temperature. This will be possible in appropriate approach and a relevant modeling environment in which such equations and functions can be inserted easily.

### 7.2.3 Choosing a Modeling Approach, a Modeling Tool and a Modeling Software

We compare different modeling approaches of AMS circuits (geometrical, electrical, and behavioral) to choose the most suitable one (Fig 7.9). We focus on their advantages and disadvantages for solving the modeling issue of HTE circuits. All approaches will be compared regarding to some specific criteria that can evaluate their performances.

Structural modeling approach is based on a circuit description with elementary and usual components such as resistances, capacitances, transistors and diodes. The developed model is called device-level abstraction model. A macromodel is an “electrical” model that is supposed to present a maximal simplification of the circuit. Moreover, generally, the quasi-totality of transistors is eliminated in a macromodel. The advantage of a macromodel is that nonlinear equations of transistors are replaced by linear equations of usual components.

SPICE program is the most used simulator of electrical models and macromodels. Commercial versions such as SPICE contain specific libraries of each semiconductor constructor. In addition, SPICE simulator allows temperature effect description in SPICE models and the simulation in temperature can be achieved easily with a parametric analysis. Nevertheless, industrial SPICE models of conventional electronics could be inaccurate in HT, and their revision cannot be done by designers. This is mainly due to the presence of some transistors. Reviewing their parameters is not possible in HT because only manufacturers can have detailed

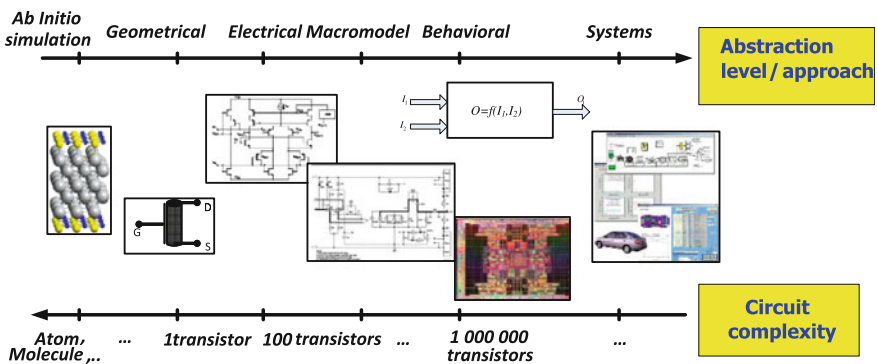


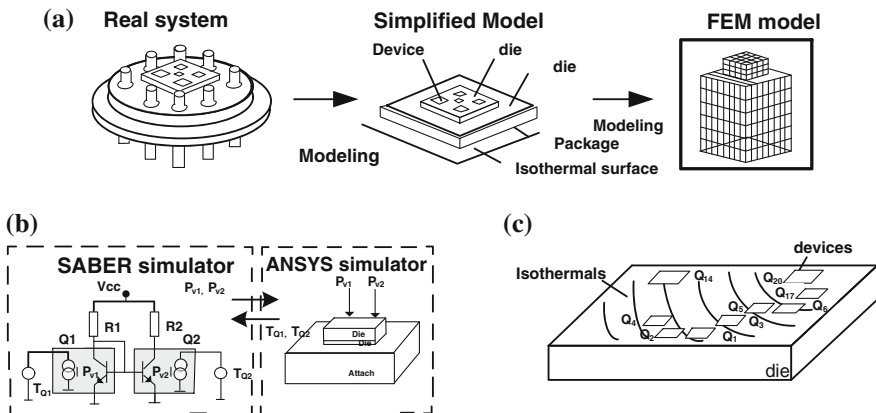
Fig. 7.9 Model evolution depending on granularity

idea about the used technology. Physical equations of transistors have to be reviewed also as their behavior changes in HT. This involves a great experimental work for characterizing transistors of the model. For all these reasons, extension SPICE model validity to HT cannot be an appropriate choice for developing an accurate op-amp model in HT.

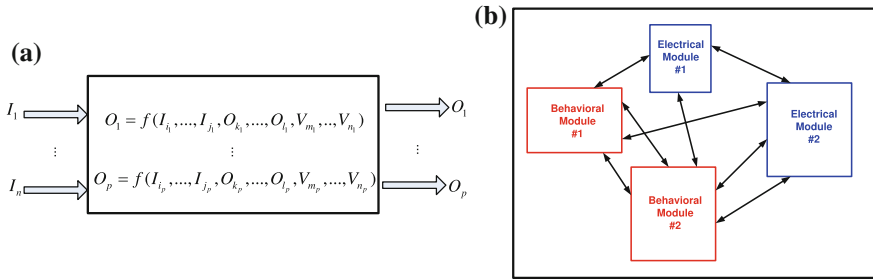
Geometrical modeling approach uses generally mathematical formulation for discretizing the space and the time in order to solve differential equations that describe the circuit behavior. This technique can be used either for transistors or more complex circuits. It uses numerical methods such as finite difference time-domain (FDTD) method and finite element method (FEM) that solve, respectively, Maxwell equations and the equation of the heat (Fig. 7.10).

The advantage of this approach consists in their great accuracy. However, its major drawback is the fact that it requires huge computing resources and a great time for simulation. Moreover, it requires a perfect knowledge about the details of internal and external geometry of the integrated circuit with all its material properties' details. For a commercial circuit, this information is not available, which presents another limitation for using this method in an industrial context. It is commonly used, however, in academic research works in order to investigate capacities and performances of new and modern circuits.

A behavioral modeling approach describes a circuit with only mathematical equations instead of electronic devices. A behavioral model does not focus on the structure of the circuit nor on its electronic details and physical realization or even its equivalent electronic representation. The circuit is described by some relations linking the input parameters ( $I_1, I_2, \dots, I_n$ ) to output parameters ( $O_1, \dots, O_p$ ). Other internal or local variables can exist in these equations to describe either global behavior or a local and specific behavior of the circuit. As illustrated in Fig. 7.11, we note that a mixed modeling approach consists in modeling behaviorally some parts of the circuit and representing other parts with structural models. This is



**Fig. 7.10** Geometrical modeling of uA741 op-amp [18]. **a** Steps of thermal modeling process. **b** Values transfer between the two models. **c** Simplified layout of uA741



**Fig. 7.11** Behavioral modeling (a) and mixed modeling (b)

especially useful when these structural models are not complex and available or when it seems to be difficult to find mathematical relations between parameters. Such models give more flexibility for designers but require a convenient modeling language and simulation environment to take into account the two model types.

It will be also an appropriate choice even for developing more AMS circuits for HT industrial applications. Actually, as circuits become more and more complex, even macromodels require non-neglected computing time and memory. As behavioral models represent a great simplification of the circuit, we expect that such model will accelerate simulation. Then, extension of SPICE model is a difficult and limited solution that requires an access to the technology information each time transistors are present in the model.

Moreover, nowadays, behavioral modeling has benefitted from the advances in computational intelligence in order to improve its performances and remedy to some limitations, especially for an industrial using. Actually, as behavioral modeling is a technique that becomes more and more used and integrated in EDA tools, automated generation of models with these tools from measurement and experimental data becomes a new challenging research topic. Indeed, simple behavioral models are developed manually by humans and are easily interpretable by their user. Generally, such behavioral models are developed in the basis of performance parameters and figures of merit. They are developed for a simple device and are generally limited to describe a global behavior of the circuit. By increasing the number of performance parameters, the manual development of behavioral model becomes a complex task. Describing the interdependency between these parameters makes the manual development harder and slower which cannot be acceptable in an industrial application. There is rather a more attractive alternative: automating generation of behavioral models for AMS circuits. There are many techniques that enable such automation and that have been studied in literature [19–22]. We cite especially the method of model order reduction (MOR) [23] and the regression method [24].

Thus, behavioral modeling offers many advantages: possibility of extending the model, automating its generation, reduction time and computing resources, and avoiding the restriction of the perfect knowledge of the inner and physical structure

**Table 7.3** Comparison of different modeling approaches of AMS circuits (example of analog-to-digital converter) [25]

Modeling approach	Complexity	Precision	Flexibility	Application to indus. HTE
Transistor level	--	++	--	--
Macromodeling/electrical	+	+	+	-
LUT method	-	+	--	--
Behavioral modeling	+	+	+	++
Mixed modeling	+	+	++	++

of the circuit. For all these reasons, we have selected finally the behavioral/mixed modeling technique to develop a new and accurate op-amp model in HT. However, the model will be developed manually as we focus here only on the evolution of each performance parameter of the op-amp. The obtained model can be then revised and enriched by using in a second step an appropriate automatic generation tool of behavioral model. This can be treated in a future works.

A more detailed discussion about the choice of the modeling approach for AMS circuits was already achieved in [25, 26]. Results of this discussion with different considered criterion are summarized by Table 7.3.

After choosing the modeling approach, we will select the modeling and simulation tools. Two categories of tools are generally used: the environment MATLAB/Simulink and hardware description languages (HDLs). MATLAB/Simulink has the ability to develop behavioral models by using the rich mathematical functions in MATLAB and mathematical function boxes of Simulink. Moreover, MATLAB/Simulink includes many useful toolboxes that are specialized in many simulation topics (signal processing, mobile communications, etc.). Recent versions of MATLAB include some useful toolboxes dedicated to electronic simulation such as “SimPowerSystems” and “SimElectronics.” The first Simulink toolbox extends capacities of Simulink to modeling and simulating many power electronics circuits. This toolbox enables a multi-domain simulation. Differential equations are elaborated from these models and integrated in the global Simulink model. They will be solved by one of the different Simulink solvers. The library “SPICE compatible components” of “SimElectronics” contains SPICE models of usual AMS circuits with the same predefined parameters. These models define particularly the temperature parameter. Thus, thanks to “SimElectronics” toolbox, it is possible to develop with MATLAB/Simulink a mixed model in which some parts are described with mathematical functions, whereas some parts are described with electrical components represented in SPICE format. This offers a great flexibility since in SPICE simulators, adding customized blocks with mathematical functions is not well developed. However, the representation of the temperature effect is only limited to some circuits in the library “SimElectronics.” So, in spite of the interest of the new added toolbox, MATLAB/Simulink cannot still defy other common tools for modeling and simulation of electronic circuits. The idea of mixing

electrical description with mathematical equations can be kept. This is possible if we use a HDL as modeling tool.

HDLs are more known in digital electronics modeling and simulation (examples of VHDL and Verilog). These languages with their particular syntax have the advantages to model circuits by mixing structural and behavioral descriptions. This is the great asset of these languages as there is a great flexibility to move from an abstraction level to another depending on the modeling constraints, specifications, and characteristic of the circuit. VHDL and Verilog languages were extended to model and simulate AMS circuits [27–29]. The results were two new modeling languages: VHDL-AMS and Verilog-AMS that kept almost the same syntax. Some extensions were added to describe analog quantities (voltage, current, etc.) and to describe circuit details (nodes, branches, etc.). In opposition to Verilog-AMS which was first used by industrials, VHDL-AMS was standardized since 1999 [30]. Moreover, a main and interesting feature of VHDL-AMS consists in dividing a VHDL-AMS model in two parts: an entity and architecture. The entity defines the generic parameters of the model, its output and its input ports. The architecture part defines the structure that models the circuits with a mixing of structural, physical, and behavioral description. By considering these features, we have chosen VHDL-AMS to develop an op-amp model in HT. We have used ADVanceMS of Mentor Graphics to develop this VHDL-AMS model which will be then stored in Cadence library after its validation. The test-bench circuits were also simulated in Cadence Virtuoso environment. We have chosen Cadence because it is largely used by designers and because it can be easily interfaced to ADVanceMS.

### 7.2.3.1 A Summary of the Modeling Methodology

Figure 7.12 summarizes different steps that were followed to reach the final model in HT. Some steps were already described: the study of the circuit specification, SPICE simulation of some op-amp parameters, and the study of the structure of the SPICE model. These tasks will be useful when studying the precision of the VHDL-AMS op-amp model. The characterization will let to know the evolution of each parameter over temperature. Such variation can be expressed with mathematical function (especially exponential or polynomial) by using a fitting tool. This function can be then easily inserted in the VHDL-AMS model. This task is called “parameters extraction.” In order to optimize the VHDL-AMS model precision, the fitting operation is achieved many times until reaching an acceptable mean fitting error. The specification error evaluates the difference between measurement results and specification data. The VHDL-AMS op-amp model is set in two steps. First, a model in VHDL-AMS that does not depend on temperature is developed. We start with a model of an ideal op-amp that will be enriched by adding at each time a new part that describes an imperfection and inserting its respective parameter (Table 7.4). The new model is each time validated with an appropriate test-bench circuit.



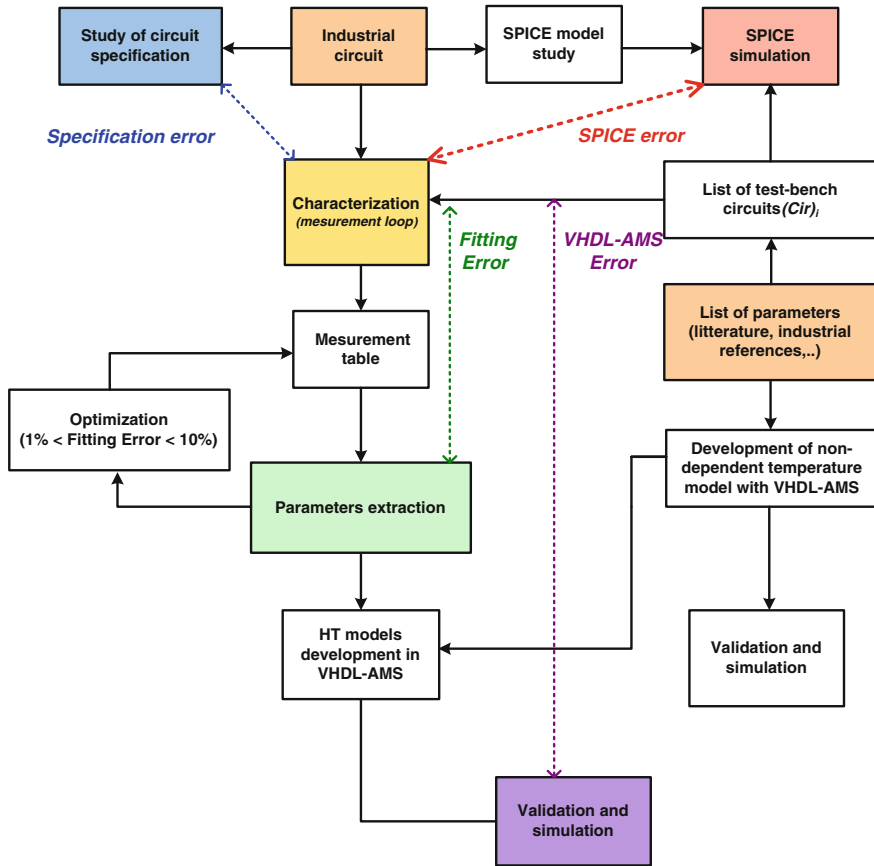


Fig. 7.12 Modeling methodology and interaction between different steps

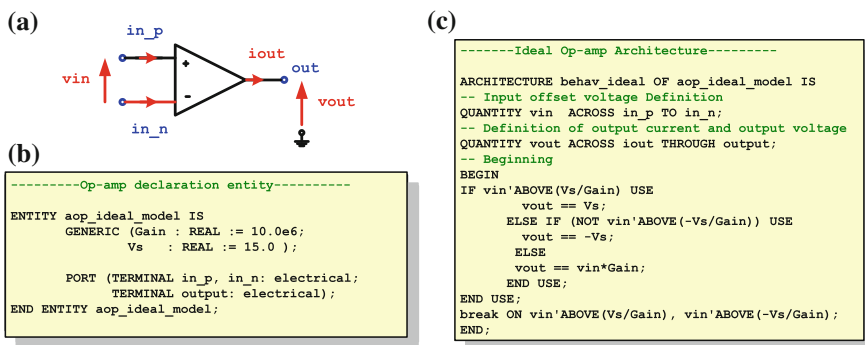
### 7.3 Optimizing HT Model Precision with Behavioral Approach

#### 7.3.1 Ideal Op-Amp Model

The ideal op-amp model will be the “core” of the non-ideal model as non-idealities will be added to it gradually with their respective performance parameters. The model starts with the definition of required libraries to use operators and particular types in the model. Figure 7.13 shows the “ENTITY” and “ARCHITECTURE” parts of the op-amp model. The ideality means that the op-amp gain has a great value (instead of an infinite value in theory) and the saturation voltages equal to nominal values (generally equal to the power supply voltages). More precisely, the open-loop gain equals  $10^7$  in linear scale, and the saturation voltages are symmetric

**Table 7.4** List of inserted parameters in the non-ideal model of the op-amp

Symbol	Name	Symbol	Name
$V_{os}$	Offset voltage	PSRR <sub>-</sub>	Negative power supply rejection ratio
$V_{oh}$	High saturation voltage	$I_{omax}$	Maximal current delivered by the op-amp
$V_{ol}$	Low saturation voltage	$I_{cc+}$	Parasitic current in the positive power supply port
$I_{os}$	Input offset current	$I_{cc-}$	Parasitic current in the negative power supply port
$I_b$	Input bias current	$R_{in}$	Differential Input resistance
$A_{ol}$	Open-loop gain	$R_{cm}$	Common-mode input resistance
GBWP	Gain bandwidth product	$C_{in}$	Differential input capacitance
SR <sub>+</sub>	Positive slew rate	$R_{out}$	Output resistance
SR <sub>-</sub>	Negative slew rate	$R_L$	Load resistance
CMRR	Common-mode rejection ratio	$C_L$	Load capacitance
PSRR <sub>+</sub>	Positive power supply rejection ratio	$g_m$	Op-amp trans-conductance

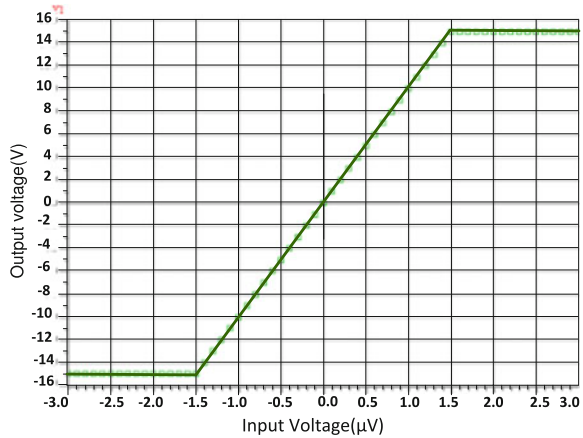


**Fig. 7.13** Parts of the op-amp model, **a** op-amp symbol, **b** ENTITY, and **c** ARCHITECTURE

and equal  $\pm 15$  V. No errors are present in this first model. Furthermore, the effects of temperature or frequency are not taken into account.

Generic parameters are **Gain** (the gain) and **Vs** (positive voltage saturation). **ENTITY** part defines also with **PORT** the inputs and outputs of the op-amp and their natures (**ELECTRICAL**). The model comprises exactly two inputs (**in\_p** and **in\_n**) and one output (**output**). The power supply voltages are not present in the model. They are simply assimilated to the values of the saturation voltages. In non-ideal op-amps, there is a little difference in values of these two parameters. **ARCHITECTURE** part describes the op-amp behavior. It starts with the definition of the currents and voltages in the op-amp inputs using **QUANTITY**. **ARCHITECTURE** describes the different zones of the op-amp characteristic curve by using **IF** loop. This is a purely behavioral description of the characteristic that cannot be

**Fig. 7.14** Simulation result of an open-loop circuit using the ideal VHDL-AMS op-amp model



defined in SPICE-based simulators. Using of **BREAK** and **ABOVE** attributes enables to prepare the simulator to slope discontinuity when changing from the linear part to one of the saturation parts. Once the model compiled, a symbol is generated and will be stored in the library in which the VHDL-AMS model was created. It can be reused in more complex circuits. After that, the model will be tested in Cadence in order to observe the op-amp characteristic curve. The simulation result is depicted in Fig. 7.14. The test-bench circuit is an open-loop circuit. The simulation result presents two saturation zones and a linear zone. Moreover, the simulated characteristic presents a perfect symmetry. This means that there is no offset voltage error. We can read also easily the input voltages from which the zone is no longer linear. Their values are  $-1.5$  and  $1.5 \mu\text{V}$ . Their output values are, respectively,  $-15$  and  $15 \text{ V}$ . So, we can calculate graphically the gain value, and it is equal to the linear slope:  $p = 30/3 \times 10^6 = 10^7$ . This is exactly the inserted value of the gain in the model. Thus, the first ideal op-amp model was validated.

### 7.3.2 Non-ideal Op-Amp Model

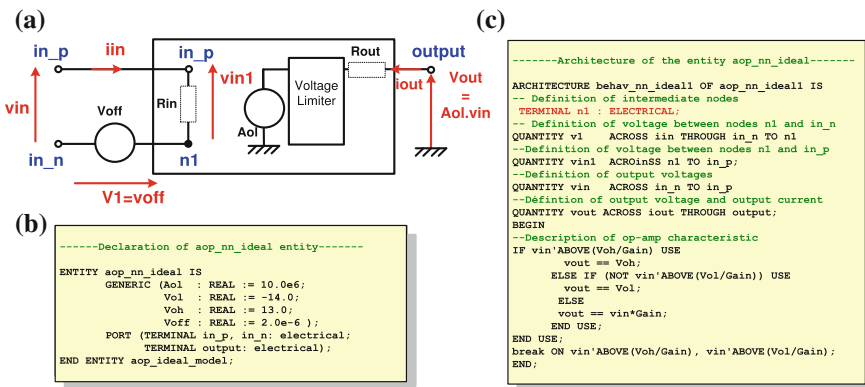
The definitive list of used performance parameters of the non-ideal op-amp is detailed in Table 7.5. We describe in this paragraph the modeling of each parameter, its validation, and its simulation with the appropriate test-bench circuit.

#### 7.3.2.1 Offset Voltage, Saturation Voltages, and Open-Loop Gain

In order to model these parameters, we carry out some modifications to the initial VHDL-AMS model. First, ports' configuration is kept, but some generic parameters are added: **Ao1** (open-loop gain), **Voff** (offset voltage), **Vol** (low saturation

**Table 7.5** Comparison of offset voltage values found by VHDL-AMS simulation

Temperature (°C)	Spice (μV)	Measurement (μV)	Fitting value (μV)	VHDL-AMS (μV)
20	94.84	64.8	64.3735	64,3714
40	115.24	69.4	70.0983	70.0927
80	158.43	88.65	88.4068	88.3793
120	204.5	110.40	110.061	109.97
160	253.09	273.1	288.14	287.16
180	277.96	1427.75	1412.64	1409.27
220	323.34	11988.2	11983.66	11982.16

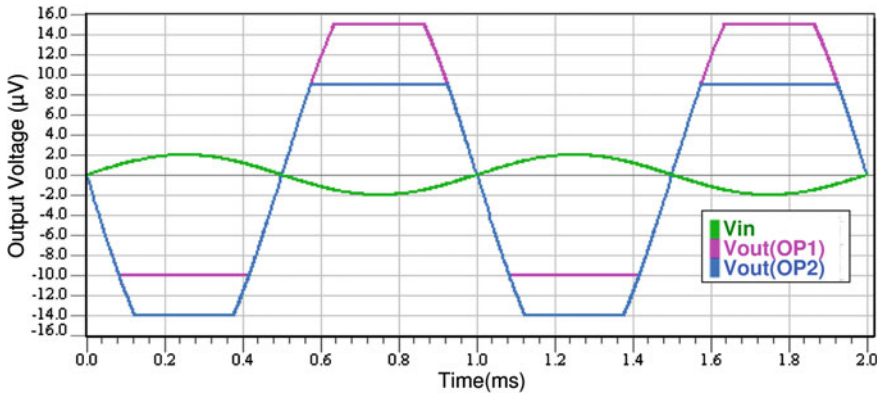


**Fig. 7.15** VHDL-AMS model of a non-ideal op-amp model including offset voltage and saturation voltages

voltage), and **Voh** (high saturation voltage). Saturation zones are described by **Vol** and **Voh** parameters (Fig. 7.15).

A further node (**n1**) is added in order to define a voltage source between the input **in\_n** and **n1**. This voltage source delivers a voltage **v1**. The offset voltage is affected to **v1**. Thus, by defining so, the different nodes and voltages and as the simulator uses Kirchhoff laws, the offset voltage will be added to the op-amp input. The parameters **Voh** and **Vol** are affected to **Vout** in **IF** loop depending on the saturation zone. The test bench consists in an open-loop circuit in which a DC voltage in a first time then a sinusoidal source are connected to the op-amp. Two analyses are achieved: a DC analysis for a DC source and a transient analysis for the sinusoidal source. Two samples of the op-amp were used with different parameter values (op-amp 1:  $V_{os} = 5 \mu V$ ,  $V_{oh} = 15 V$ ,  $V_{ol} = -14 V$ , Gain =  $10^5$ ; op-amp 2:  $V_{os} = 2 \mu V$ ,  $V_{oh} = 9 V$ ,  $V_{ol} = -10 V$ , Gain =  $10^6$ ).

By using a DC analysis, we have observed two op-amp characteristic curves similar to the curve plotted in Fig. 7.14, one for each op-amp. These two curves were not presented. The values of high saturation voltages obtained by DC analysis



**Fig. 7.16** Transient simulation of saturation voltages using a sinusoidal source

were 15 V for op-amp 1 and 9 V for op-amp 2. The low saturation voltages obtained by the same analysis are  $-14$  V for op-amp 1 and  $-10$  V for op-amp 2. The slope of the linear part is, respectively,  $10^5$  and  $10^6$ . The offset voltage value was read by making a zoom in abscise axis. The simulated values are equal to  $5 \mu\text{V}$  for op-amp 1 and  $2 \mu\text{V}$  for op-amp 2. In the transient analysis, illustrated by Fig. 7.16, the same values of saturation voltage are found. They are the same values that were inserted into the model. Thus, modeling saturation voltages were validated by these two simulations.

### 7.3.2.2 Input Offset and Bias Currents

Ideally, there are no currents in the inputs of an op-amp. Industrial op-amps are not perfectly ideal. Some currents are measured in their inputs even if the input voltage equals zero. These currents are caused by the same factors that cause the offset voltage: imperfections of transistors in the input stage [11]. The measured currents in the positive and negative inputs are called, respectively, positive input bias current and negative input bias current and are denoted, respectively,  $I_{b+}$  and  $I_{b-}$ . Input bias currents are always not equals. Their difference is called the input offset current and is denoted  $I_{os}$ . Relations between  $I_{b+}$ ,  $I_{b-}$ , and  $I_{os}$  are defined by Eqs. (7.9), (7.11), and (7.12). The mean of  $I_{b+}$  and  $I_{b-}$  is called input bias current and is defined by (7.10). From Eqs. (7.9) and (7.10) and as Kirchhoff’s current laws are used by the simulator, we can define three current sources to insert the bias and offset currents in the VHDL-AMS model as illustrated in Fig. 7.17.

$$I_{os} = I_{b+} - I_{b-} \tag{7.9}$$

$$I_b = \frac{I_{b+} + I_{b-}}{2}. \tag{7.10}$$

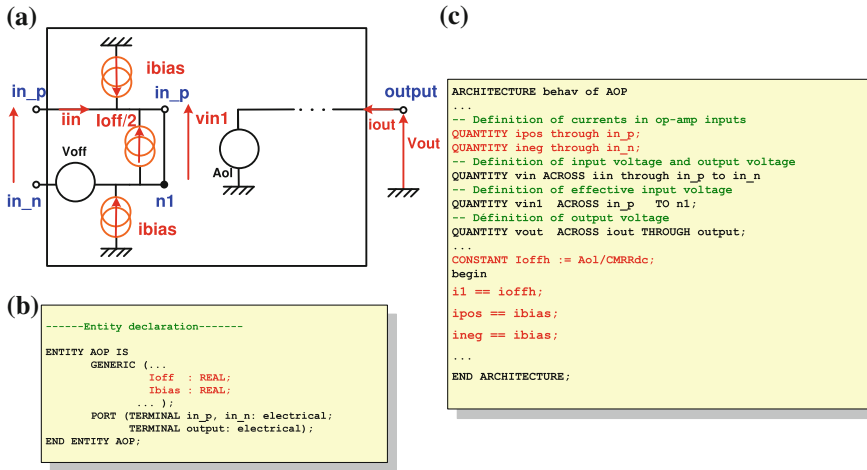


Fig. 7.17 Modeling the input bias current and the offset current in the VHDL-AMS model

$$I_{b+} = I_b + \frac{I_{os}}{2}. \tag{7.11}$$

$$I_{b-} = I_b - \frac{I_{OS}}{2}. \tag{7.12}$$

**I<sub>b</sub>** and **I<sub>off</sub>** implantations were tested and validated with a simple circuit in which an op-amp is directly linked to the ground without inserting any source. Then, a transient analysis is performed, and currents in the op-amp inputs are simulated. The chosen values of generic parameters **I<sub>off</sub>** and **I<sub>bias</sub>** are, respectively, 3 and 2.5 nA. Theoretical values of  $I_{b+}$  and  $I_{b-}$  are, respectively, 1 and 4 nA. All these values were found also by simulation, which means that their modeling is validated.

### 7.3.2.3 Differential Input Resistance, Differential Input Capacitance

Ideal op-amp has an infinite resistance and a null capacity. Industrial op-amps represent very big resistance and very low capacitance values. We have inserted these components in the VHDL-AMS model. Modeling of the resistance and capacitance is done by using their characteristic equations (Ohm’s law equation and the equation linking the current to capacitance and the derivative of the voltage) (Fig. 7.18).

The simulation is again performed with an open-loop circuit in which the developed op-amp model is connected to a sinusoidal voltage source. The chosen analysis is the transient analysis. The simulation with a sinusoidal source enables to observe the effect of the operator ‘**dot**’ that calculates the derivative regarding time.

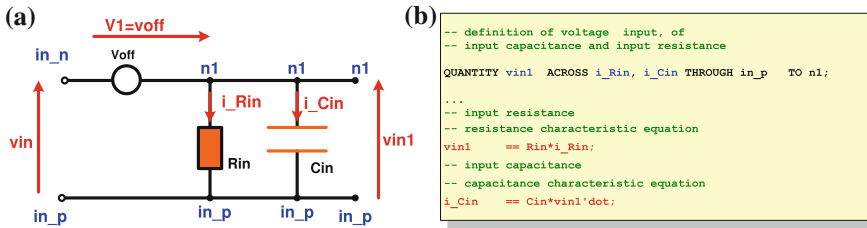


Fig. 7.18 Modeling the input resistance and the input capacitance

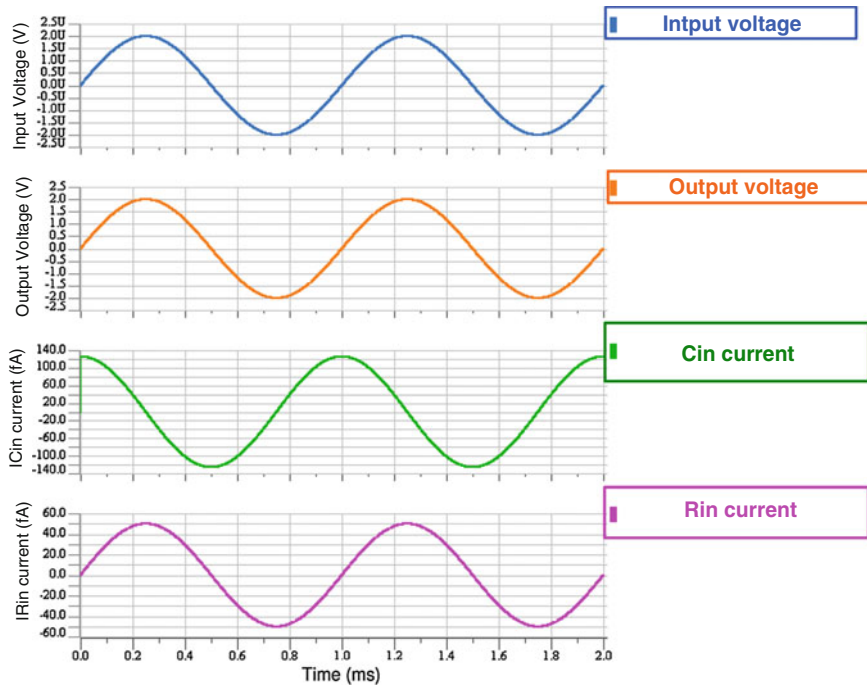


Fig. 7.19 Simulation results of input resistance and input capacitance

The chosen frequency is 1 kHz, and the magnitude equals 1 V. The inserted values of  $R_{in}$  and  $C_{in}$  are, respectively, 40 M $\Omega$  and 10 pF. The currents  $i_{Rin}$  and  $i_{Cin}$  in the ports of the resistance and the capacitance are sinusoidal. However, the phase of  $i_{Cin}$  equals  $\pi/2$  contrarily to  $i_{Rin}$  whose phase equals 0. This is due to the operator 'dot' in the expression of the capacitance current. Frequencies of all plotted currents are the same and equal 1 kHz. The theoretical value of the magnitude of  $i_{Cin}$  is  $I_{Cin} = C_{in} \cdot 4\pi \cdot f \times 10^{-6} = 125.66 \times 10^{-12}$  A. This is almost the magnitude of the capacitance current in Fig. 7.19. Thus, the implantation of differential resistance and capacitance is validated.

### 7.3.2.4 Modeling and Simulation of CMRR, PSRR, $R_{cm}$ , and $C_{cm}$ Parameters

The insertion of CMRR,  $R_{cm}$ , and  $C_{cm}$  parameters enables to model the effects of the common mode. Contrarily to the differential mode, the common mode consists in connecting the two op-amp inputs to a unique node. In such case, an ideal op-amp will deliver a null value. However, for an imperfect op-amp, the output voltage is different from zero. The common-mode voltage  $V_{in\_cm}$  is the input voltage in a common mode and  $V_{out\_cm}$  is the output voltage in the same mode. We define the common-mode gain, denoted  $A_{cm}$ , as the ratio of  $V_{out\_cm}$  by  $V_{in\_cm}$ . So, the total output voltage  $V_{out}$  of an op-amp depends rather on both the differential mode and the common mode as shown in (7.13) where  $A_d$  is the differential gain. Expressions of  $V_{in\_cm}$  and  $V_{in\_d}$  as the function of voltages  $V_p$  and  $V_n$  (respectively, voltage in positive and negative ports) are given by Eqs. (7.14) and (7.15).

$$V_{out} = A_d \cdot V_{in\_d} + A_{cm} \cdot V_{in\_cm} \quad (7.13)$$

$$V_{in\_d} = V_p - V_n \quad (7.14)$$

$$V_{in\_cm} = (V_p + V_n)/2 \quad (7.15)$$

CMRR is defined as the ratio of the differential gain  $A_d$  by the common-mode gain  $A_{cm}$ . For an ideal op-amp, the CMRR value equals the infinity. For a real op-amp, CMRR values in dB are very high and they exceed generally 100 dB. The expression of the output voltage becomes as shown in (7.17).

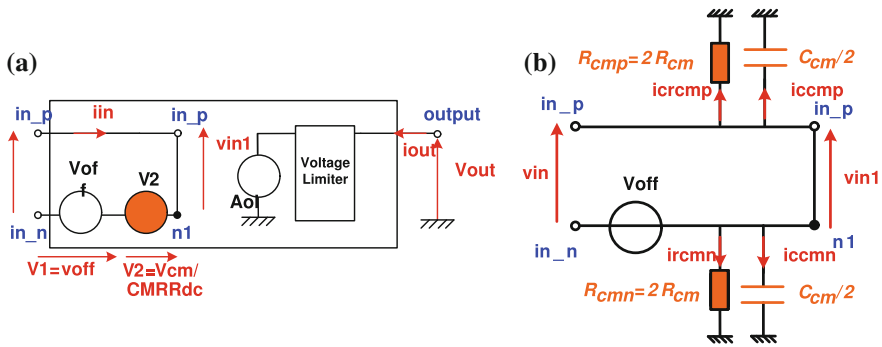
$$CMRR = A_d/A_{cm} \quad (7.16)$$

$$V_{out} = A_d(V_{in\_d} + V_{in\_cm}/CMRR) \quad (7.17)$$

$V_{in\_cm}/CMRR$  is considered as a further offset voltage whose value depends on the common-mode voltage  $V_{in\_cm}$  and mainly on CMRR. Thus, we insert in the model the effect of the common mode almost in the same manner with it the voltage offset was inserted. Moreover, CMRR has generally a frequency dependency that will be described in the paragraph dealing with the frequency response of the op-amp. Here, only the DC part of CMRR is considered, that is why the parameter was called **CMRRdc** in the VHDL-AMS model (Fig. 7.20).

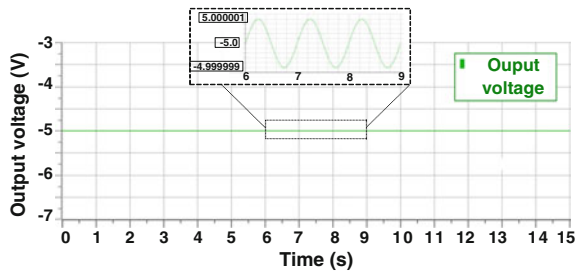
In order to validate the implantation of CMRR, we used a test-bench circuit in which a sinusoidal source is connected to an op-amp in a common mode. Simulation results of a transient analysis are illustrated by Fig. 7.21. The inserted values of parameters are  $V_{os} = 5.0 \mu V$ ,  $CMRR = 120.0 \text{ dB}$ , and  $A_{oi} = 120 \text{ dB}$ . The expression of the voltage source is given by Eq. (7.18). The theoretical value of  $V_{in\_d}$  is zero. Consequently,  $V_{in\_cm}$  equals the source voltage as shown also in (7.18). The new expression of  $V_{out}$  is given by Eq. (7.19). A zoom shows a sinusoidal variation with amplitude equaling  $1 \mu V$  added to a DC part that equals  $-5 V$ .





**Fig. 7.20** Modeling CMRR parameter and defining the common-mode voltage **a** modeling the common-mode resistance and **b** capacitance

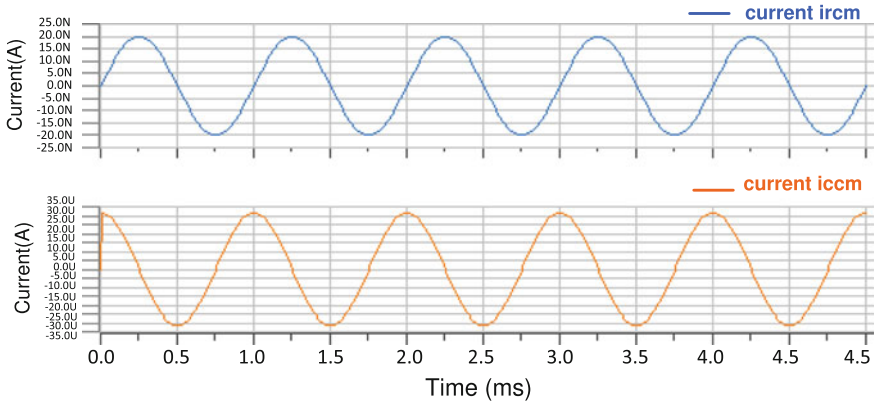
**Fig. 7.21** Simulation result of the DC part of CMRR



$$V_{in,cm} = 1.0 \mu\text{V} \sin(2\pi \cdot f_{in}t) = V_{in} \tag{7.18}$$

$$V_{out} = A_{ol} \cdot V_{os} + A_{ol} \cdot V_{in,cm}/\text{CMRR} = -5 + 1.0 \times 10^{-6}(\sin 2\pi \cdot f_{in}t) \tag{7.19}$$

This theoretical result is the same that the obtained value by simulation. This validates the implantation of the CMRR parameter in the model. The common mode is also characterized by a common-mode resistance  $R_{cm}$  and a common-mode capacitance  $C_{cm}$ . In the model, two resistances are defined in order to represent  $R_{cm}$  ( $R_{cmp}$  is connected to the positive input, and  $R_{cmn}$  is connected to the negative input). In the same manner, two capacitances are defined in order to represent  $C_{cm}$ . Ideally, there is symmetry between these resistances and capacitances. Consequently,  $R_{cmp} = R_{cmn} = 2R_{cm}$  and  $C_{cmp} = C_{cmn} = C_{cm}/2$ . Again, the same test-bench circuit is used to test the implantation of  $R_{cm}$  and  $C_{cm}$ . We define  $i_{rcm}$  and  $i_{ccm}$  as the currents crossing, respectively, common-mode capacitance and resistance in one of the op-amp inputs. The considered values of  $R_{cm}$  and  $C_{cm}$  are, respectively, 25 M $\Omega$  and 10 nF. The expression of the input voltage is  $V_{in} = A \cdot \sin(2\pi \cdot f_{in}t)$ . The expressions of  $i_{ccm}$  and  $i_{rcm}$  are given by Eqs. (7.20) and (7.21).  $V_{rcm}$  is the common-mode voltage and is equal to  $V_{in}$ .



**Fig. 7.22** simulation results of  $i_{ccm}$  and  $i_{rcm}$

$$i_{ccm} = (C_{cm}/2) dV_{rcm}/dt \quad (7.20)$$

$$i_{rcm} = V_{rcm}/(2R_{cm}) \quad (7.21)$$

Because of the derivative, the current phase changes from 0 to  $\pi/2$ . The theoretical value of the magnitude of  $i_{rcm}$  is then equal to  $V_{rcm}/(2R_{cm}) = 1/(2 \times 25.10^8) = 20$  nA and that of  $i_{ccm}$  is equal to  $(C_{cm}/2) \times 2\pi f_{in} = 31.44$   $\mu$ A ( $A = 1$  V and  $f_{in} = 1$  kHz). These are almost the same values obtained by simulation (Fig. 7.22). We have thus validated the implantation of the common resistance and common capacitance.

We model now the PSRR parameter. It evaluates the effect of a power supply voltage changing  $\Delta V_{sup}$  leading to a variation of the output voltage  $\Delta V_{out}$ . Its expression is then  $PSRR = \Delta V_{sup}/\Delta V_{out}$ . Such variation is due also to imperfections of transistors that are close to power supply ports. Since there is not a perfect symmetry between the positive and the negative power supply ports, two PSRRs are defined in some references:  $PSRR_+ = \Delta V_{sup+}/\Delta V_{out+}$  for the positive power supply port and  $PSRR_- = \Delta V_{sup-}/\Delta V_{out-}$  for the negative power supply port. In order to insert these two parameters in the VHDL-AMS model, two power supply ports were added in **ENTITY**. The parameters **PSRRdc\_p** and **PSRRdc\_n** are added to the generic parameters list in **ENTITY**. The initial values of other parameters are CMRR = 120 dB,  $A_{ol} = 60$  dB, and  $V_{os} = 0$  V. Values of  $PSRR_+$  and  $PSRR_-$  equal both 120 dB. Instead of nominal values of 15 V and  $-15$ , the power supply voltages are 14 V in the positive port and  $-15$  in the negative port. This means that  $\Delta V_{sup+} = 1$  V and  $\Delta V_{sup-} = 0$ . The new expression of  $V_{out}$ , with these different numerical values, is given by Eq. (7.22). The test bench contains only the op-amp model and the two DC voltage sources for power supply (having 14 and  $-15$  V values). There are no input voltages in this circuit. We choose transient analysis and plot  $V_{out}$ . Theoretical value of  $V_{out}$  is the same that of the simulated

value (1 mV). By inverting the ports and the values of  $\Delta V_{\text{sup}+}$  and  $\Delta V_{\text{sup}-}$ , we find the same results. This validates the implantation of  $\text{PSRR}_+$  and  $\text{PSRR}_-$ .

$$V_{\text{out}} = A_{\text{ol}} \cdot \underbrace{V_{\text{in,diff}}}_0 + A_{\text{ol}} \cdot \underbrace{V_{\text{os}}}_0 + A_{\text{ol}} \cdot \underbrace{V_{\text{in,cm}}}_0 / \text{CMRR} + A_{\text{ol}} \cdot \underbrace{\Delta V_{\text{sup}+}}_1 / \text{PSRR}_+ + A_{\text{ol}} \cdot \underbrace{\Delta V_{\text{sup}-}}_0 / \text{PSRR}_- \quad (7.22)$$

### 7.3.2.5 Modeling the Frequency Response of the Op-Amp

By frequency response, we mean not only the frequency dependency of open-loop gain but also frequency dependency of some main other parameters such as CMRR,  $\text{PSRR}_+$ , and  $\text{PSRR}_-$ . Generally, an op-amp has a behavior of a first-order filter. This behavior is characterized by GBWP parameter. GBWP equals the product of the cutoff frequency  $f_{\text{ol}}$  by the gain  $A_{\text{OLDC}}$ . It equals also the frequency for which the gain is 0 dB.

$$A_{\text{OLDC}} \cdot f_{\text{ol}} = f(A_{\text{ol}} = 0 \text{ dB}) = \text{GBWP}. \quad (7.23)$$

In industrial SPICE models, a first-order filter is modeled by a RC circuit ( $f_{\text{ol}} = 1/\text{RC}$ ). With VHDL-AMS, we can model such behavior in many manners. It is possible for example to use the Eq. (7.24) defining the complex gain. It is possible also to use the differential Eq. (7.26) that is resulting from Laplace transform of Eq. (7.24). The transform equation is represented by Eq. (7.25).

$$A_{\text{ol}} = \frac{A_{\text{OLDC}}}{1 + \frac{jf}{f_{\text{ol}}}} \quad (7.24)$$

$$\text{LTF} = \frac{K}{1 + \tau s} \quad (7.25)$$

$$V_{\text{in}1} = (1/A_{\text{ol}}) \left( \frac{1}{\omega_{\text{OL}}} \cdot \frac{dV_{\text{out}}}{dt} + V_{\text{out}} \right) \quad (7.26)$$

All these three equations can be easily implanted in VHDL-AMS. As shown in Fig. 7.23a, we have used the differential equation as it will be easier to integrate it with precedent equations of the other parameters. Describing the frequency behavior of CMRR,  $\text{PSRR}_+$ , and  $\text{PSRR}_-$  has been achieved with the ‘**LTF**’ attribute. This attribute is defined regarding the parameters defining the Eq. (7.25) (Fig. 7.24).

We test the implantation of GBWP with an open-loop circuit in which the op-amp model is related to an AC source. An AC analysis enables to plot the spectrum of  $V_{\text{out}}$ . The magnitude of the sinusoidal source is 1  $\mu\text{V}$ . Power supply voltages equal both  $\pm 15 \text{ V}$ . Inserted values of GBWP and  $A_{\text{ol}}$  are, respectively, 5 MHz and 120 dB. Simulation result is illustrated by Fig. 7.25. It indicates that GBWP corresponds to  $V_{\text{out}} = -120 \text{ dB}$  and a frequency value of 5 MHz.

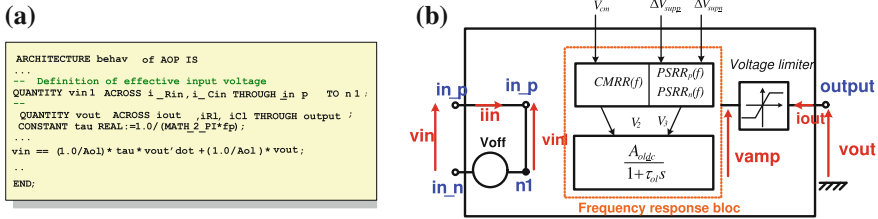


Fig. 7.23 Implantation of frequency behavior of  $A_{ol}$ , CMRR,  $PSRR_+$ , and  $PSRR_-$  (a) and VHDL-AMS modifications (b)

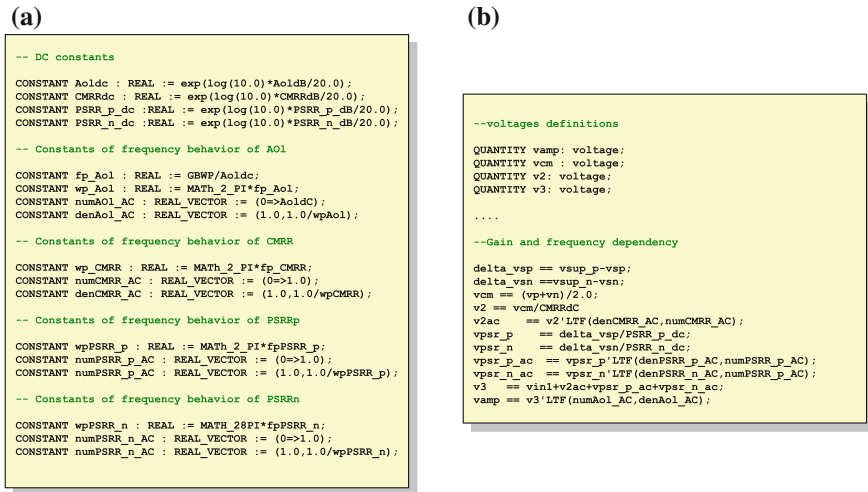


Fig. 7.24 Description of the frequency behavior of  $A_{ol}$ , CMRR,  $PSRR_+$ , and  $PSRR_-$  in the VHDL-AMS model

The simulated value of GBWP is 5 MHz. It is clear also that the cutoff frequency equals 5 Hz. By dividing 5 MHz/5 Hz and converting the value in dB, we find the simulated value of  $A_{ol}$ : 120 dB.

Simulation of the frequency behavior of CMRR uses the same circuit as that used to simulate  $CMRR_{dc}$ . However, we use here an AC analysis. The expression of the input voltage is  $V_{in} = 1 \text{ V} \cdot \sin(2\pi f_{in}t)$ . The inserted values of differential input voltage and common-mode voltages are, respectively,  $V_{in,d} = 0 \text{ V}$  and  $V_{in,cm} = V_{in}$ . Power supply voltage equal  $\pm 15 \text{ V}$ . In order to simplify the computing of theoretical values,  $V_{os}$  and  $A_{ol}$  (in dB) are taken null. Moreover, frequency cutoff of  $CMRR_n$  equals 1 MHz and  $CMRR$  equals 120 dB. By replacing these numerical values in the expression of  $V_{out}$ , we obtain first Eq. (7.27) and then Eq. (7.28) which links CMRR,  $V_{in}$ , and  $V_{out}$ .

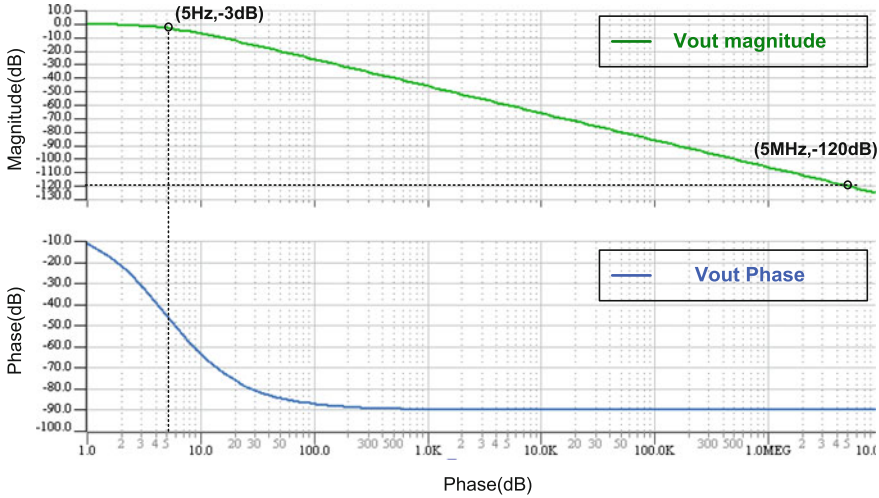


Fig. 7.25 Simulation result of  $f_{ol}$ ,  $A_{ol}$ , and GBWP

$$V_{out} = A_{ol} \cdot \underbrace{V_{in,diff}}_0 + A_{ol} \cdot \underbrace{V_{os}}_0 + \underbrace{A_{ol}}_{=1} \cdot \underbrace{V_{in,cm}}_{\neq 0} / CMRR + A_{ol} \cdot \underbrace{\Delta V_{sup+}}_0 / PSRR_+ + A_{ol} \cdot \underbrace{\Delta V_{sup-}}_0 / PSRR_- \tag{7.27}$$

$$V_{out}(dB) = (V_{in\_cm} / CMRR)_{dB} = (1 / CMRR)_{dB} = -CMRR(dB) \tag{7.28}$$

By converting in dB, the relation (7.27) becomes  $V_{out}(dB) = -CMRR (dB)$ . We can see the same evolution in Fig. 7.26a. Moreover, the value of the cutoff frequency equals the inserted value (1 MHz). This validates the implantation of the frequency behavior of CMRR.

In order to simulate the frequency behavior of  $PSRR_+$ , we connect a sinusoidal source to the positive power supply port and we apply 15 V. The expression of the total power supply voltage becomes as described by Eq. (7.29). We choose, respectively, the following values for  $V_{os}$ ,  $f_{cc+}$ ,  $A_{ol}$ ,  $PSRR_+$ , and the  $PSRR_+$  cutoff frequency are, respectively, 0 V, 1 kHz, 0 dB, 100 dB, and 1 kHz.

$$V_{cc+} = 1 V \cdot \sin(2\pi \times f_{cc+}t) + 15 V \tag{7.29}$$

$$V_{out} = A_{ol} \cdot \underbrace{V_{in,diff}}_0 + A_{ol} \cdot \underbrace{V_{os}}_0 + \underbrace{A_{ol}}_0 \cdot \underbrace{V_{in,cm}}_0 / CMRR + A_{ol} \cdot \underbrace{\Delta V_{sup+}}_{\neq 0} / PSRR_+ + A_{ol} \cdot \underbrace{\Delta V_{sup-}}_0 / PSRR_- \tag{7.30}$$

$$V_{out}(dB) = -PSRR_+(dB) \tag{7.31}$$

The expression of  $V_{out}$  is given by Eqs. (7.29) and (7.30), which is more simplified and is given in dB. The final theoretical Eq. (7.31) is in good agreement with

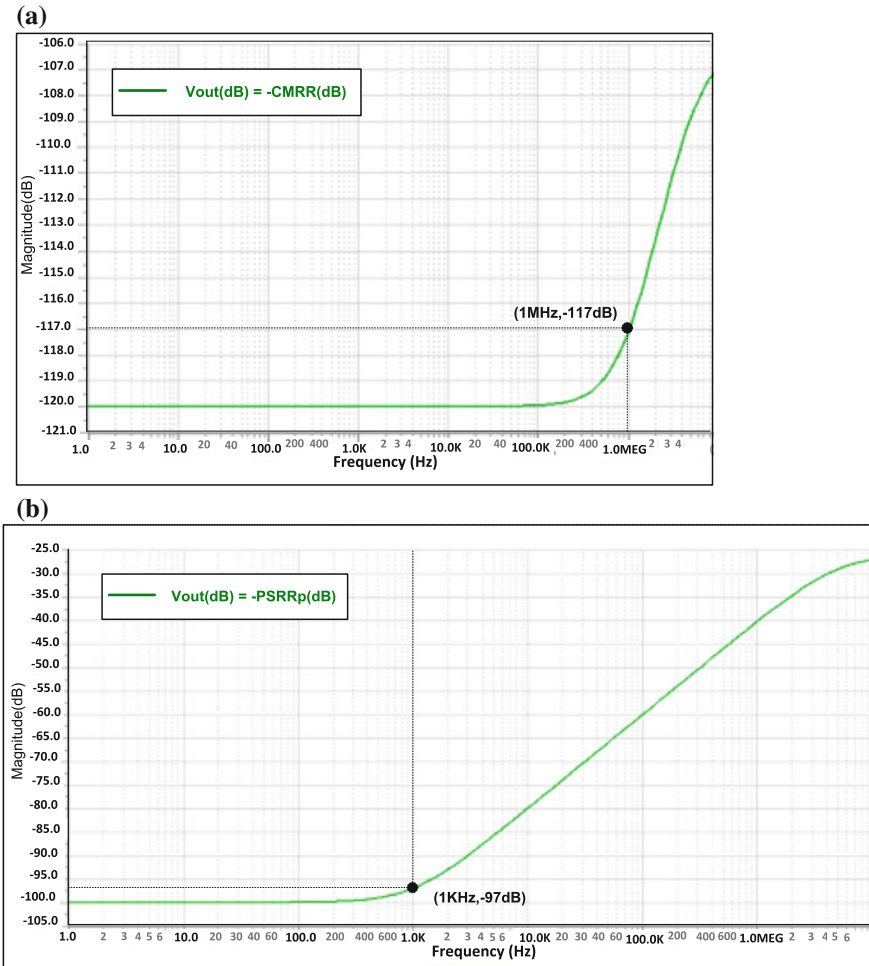


Fig. 7.26 Simulation results of frequency behavior of CMRR and PSRR<sub>+</sub>

simulation result as shown in Fig. 7.26b. Moreover, we find the same value of the cutoff frequency which is 1 kHz. This validates the implantation of frequency behavior of PSRR<sub>+</sub>.

### 7.3.2.6 Modeling of the Slew Rate

The slew rate evaluates the maximal variation in time of the output voltage with respect to input voltage. The definition of this parameter is given by Eq. (7.32), and its unity is V/ $\mu$ s. Depending on the sign of the variation, we define two parameters: SR<sub>+</sub> and SR<sub>-</sub>. In VHDL-AMS, the 'slew attribute enables to insert directly such parameter as shown in Fig. 7.27.

```
vamp_s == vamp' slew(SRp, -SRn);
```

**Fig. 7.27** Implantation of  $SR_+$  and  $SR_-$  with slew attribute in VHDL-AMS

$$SR = \text{Max} \left( \frac{dV_{\text{out}}}{dt} \right) \quad (7.32)$$

We use, however, two test-bench circuits with two different voltage sources. The first is an open-loop circuit that contains a sinusoidal source and the second a pulse voltage source. The frequency of the sinusoidal source is  $f_{\text{in}} = 10$  MHz. The period of the pulse signal is  $100 \mu\text{s}$ . It oscillates between 0 and 10 V values. We choose the value  $1 \text{ V}/\mu\text{s}$  for  $SR_+$  and  $SR_-$ . Simulation results are depicted in Fig. 7.28. The output voltage is in green, and the input voltage is in magenta.

We can see that with a sinusoidal source, the output voltage is no longer sinusoidal. It has rather a triangular shape. In the case of a pulse voltage, the output is no longer perfectly rectangular, but the signal makes a time to rise from 0 to 10 mV. The speed of the rising and the speed of falling of the output voltage equal both  $1 \text{ V}/\mu\text{s}$ . This is the same value that was inserted for  $SR_+$  and  $SR_-$  parameters. We have thus validated the implantation of these two parameters.

### 7.3.2.7 Modeling Maximal Current, the Trans-Conductance, Quiescent Currents, Output Impedance, and Load Impedance

The slew rate is related to trans-conductance and maximal current delivered by the op-amp denoted, respectively,  $g_m$  and  $I_{\text{omax}}$ . Actually, if  $C_p$  is the capacitance that is used in Boyle model, SR is related to the maximal current  $I_{\text{omax}}$  as expressed in Eq. (7.33).

$$SR = \frac{2I_{\text{omax}}}{C_p}. \quad (7.33)$$

In order to take into account  $I_{\text{omax}}$ , a current limiter is added to the VHDL-AMS model. It was implanted also with an **IF** loop similarly to the implantation of the voltage limiter defining the characteristic curve of the op-amp. Moreover, we insert a controlled voltage–current source in order to model  $g_m$ . Besides  $PSRR_+$  and  $PSRR_-$  ratios, non-idealities in power supply ports can be characterized by some currents that circulate from these ports to the output port [31]. These currents are called quiescent currents and were added to the model. We have also inserted the load resistance, the load capacitance, and the output resistance. Figure 7.29 illustrates an enriched structure of the op-amp after representing these parameters.

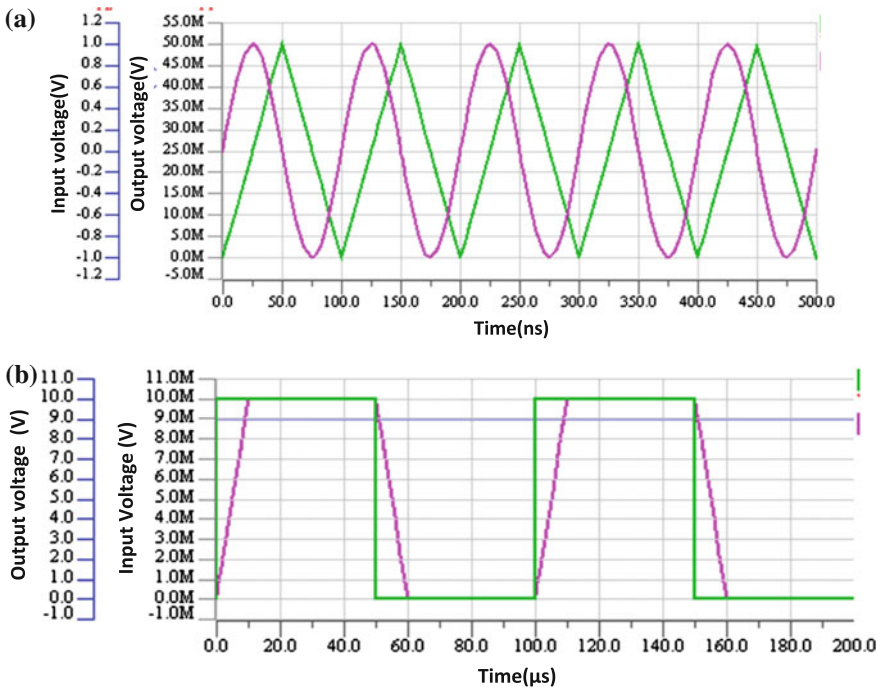


Fig. 7.28 Simulation result of the slew rate

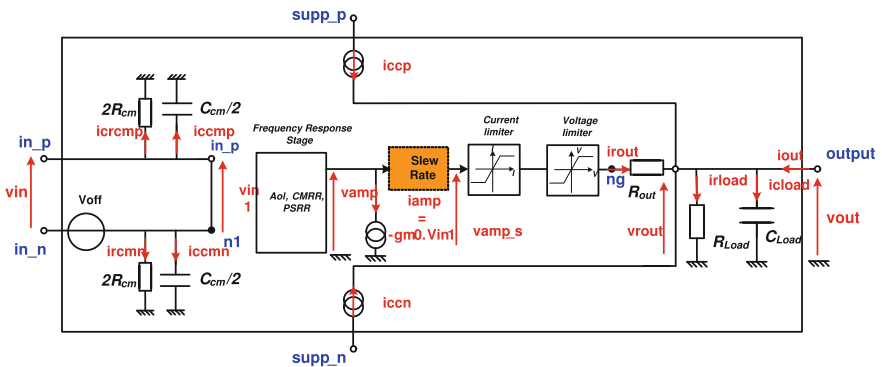
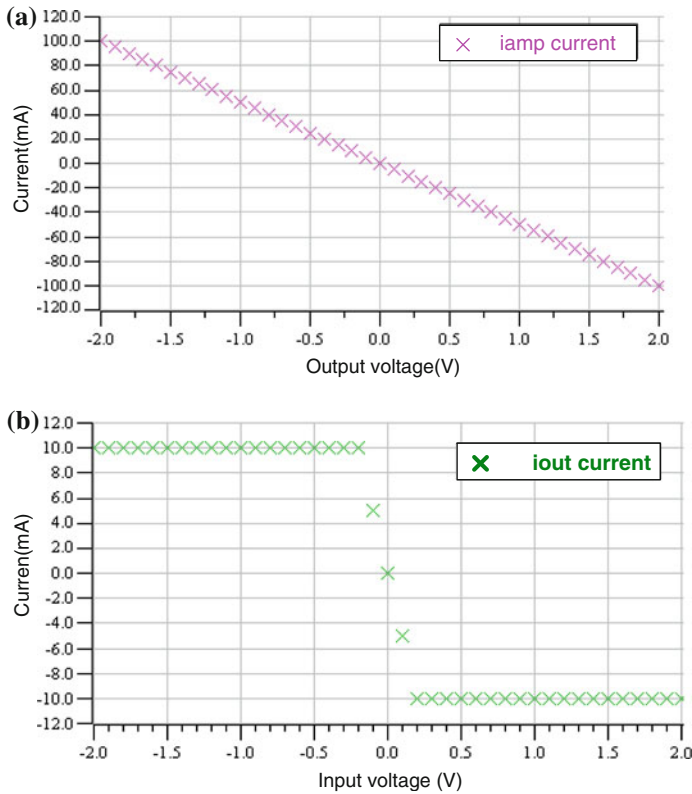


Fig. 7.29 Insertion of slew rate, trans-conductance, output resistance, load resistance, and load capacitance parameters

To test the implantation of the maximal current and the trans-conductance, we have used again an open-loop circuit. The power supply voltage values are  $\pm 15$  V. We have inserted the following values:  $R_{out} = 1 \Omega$ ,  $C_L = 100$  pF,  $R_L = 2$  k $\Omega$ ,  $I_{max} = 10$  mA,  $SR_+ = SR_- = 1$  V/ $\mu$ s,  $GBWP = 5$  MHz,  $g_m = (I_{max} \cdot GBWP) / (SR_+ \times 10^6) = 50 \times 10^{-3}$  S, and  $I_{cc+} = 1$  nA. The model is simulated with a DC





**Fig. 7.30** simulation of **i\_amp** the current generated by the trans-conductance

analysis. We plot the current **i\_amp** which is delivered by the trans-conductance in Fig. 7.30a. It is clear that evolution of **i\_amp** is linear. The slope of this line is the simulated trans-conductance of the op-amp. This value is graphically equal to  $50 \times 10^{-3}$  S. Thus, theoretical and simulated values are equal. The op-amp characteristic ( $I_{out}$ ,  $V_{in}$ ) is plotted in Fig. 7.30b. This characteristic has also two saturation zones and a linear zone. The minus sign in the equation defining the trans-conductance makes the slope negative. Moreover, the maximal current in absolute value is 10 mA. This value equals the inserted value of  $I_{omax}$ . We have thus validated the added parameters.

### 7.3.3 Parameters Extraction and Insertion of Temperature Dependency

In the previous paragraph, we have developed a non-ideal model containing many parameters by using a behavioral modeling approach and VHDL-AMS language.

All of the inserted parameters were simulated, and their implantation was validated by simple test-bench circuits. The model does not yet include a temperature dependency of these parameters. However, the resulting model is well-parameterized and includes almost all the used op-amp parameters in technical documents and references. The following step will consist in converting this model to a temperature-dependent model. In order to reach that, we define first the parameter **temp** in the generic parameters of **ENTITY**. This parameter will represent the simulated temperature value. We will express each performance parameter as a function of **temp**. Some parameters will be defined as **quantity** or **constant** in **ARCHITECTURE** part. Some others will be considered as **constant** and will not be dependent on the temperature (resistance load, capacitance load, etc.). The equation linking performance parameters to **temp** were obtained by fitting by using *Cftool* of MATLAB. Most of them are described by either polynomial or exponential functions. *Cftool* uses specific algorithms such as trust-region method, Levenberg–Maquardt method, and Newton–Gauss algorithm. This tool can achieve interpolation operation by using linear and polynomial functions or by using cubic interpolation (spline).

Evolutions of different parameters in [20 °C, 220 °C] are not similar. Actually, some parameters have almost a linear evolution. Some others have a more important evolution especially in the HT region. In some cases, we have found some difficulties to find a unique function that approximates the evolution in the entire interval [20 °C, 220 °C]. That is why we were constrained to describe the evolution of such parameters by different mathematical functions so that each function describes the parameter evolution in a subinterval of [20 °C, 220 °C]. An example of the fitting error of  $V_{os}$  in [20 °C, 140 °C] and [150 °C, 220 °C] is plotted in Fig. 7.31. Fitting error does not exceed 6 %, and its mean value is very weak in the two intervals.

We have inserted the obtained equation by fitting in the VHDL-AMS model. The validation of the implantation of the evolution for each parameter will be done with a test bench that is inspired from the experimental circuit. This test bench is more complex than the used test-bench circuits in simulation when developing the non-ideal model. It will give more credibility when comparing measurement results to simulation results of the final VHDL-AMS model. We have chosen again the offset voltage to show how its temperature dependency was implanted and to give its simulation results. Other parameters were implanted and tested in almost the same manner. In the VHDL-AMS model, as shown in Fig. 7.32, we insert first  $p_i$  coefficients defining the polynomial function. Finally, by using **IF** loop, **voff\_ht** and **voff\_bt** are affected to **voff**. **voff** will be then assigned to **v1** the voltage defined between **n1** and **in**.

From the simulation results, we have calculated the VHDL-AMS error. We have plotted this error and the SPICE error in Fig. 7.33. SPICE error is superior to 40 % for almost all temperatures. However, this error does not exceed 5 % for the VHDL-AMS model.

Table 7.5 gives some numerical values of the different obtained offset voltage values (by SPICE simulation, measurement, fitting, and VHDL-AMS simulation).

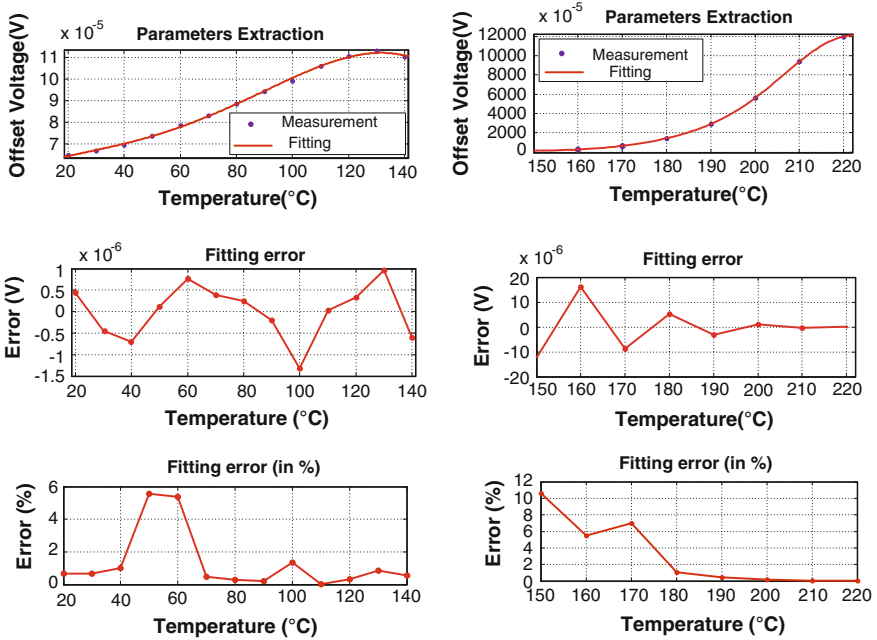


Fig. 7.31 Evolution of the fitting error in [20 °C, 140 °C] and [140 °C, 220 °C]

**(a)**

```

-- Constants definition
--//definition of offset voltage in low
--temperature (<120°C)//
CONSTANT p1 : REAL := -6.361e-13;
CONSTANT p2 : REAL := 1.472 e-10
CONSTANT p3 : REAL := -8.951 e-9;
CONSTANT p4 : REAL := 4.813 e-7;
CONSTANT p5 : REAL := 5.721 e-5;
CONSTANT voff_bt : REAL := p1*temp**4+p2*temp**3+p3*temp**2+p4*temp+p5
        
```

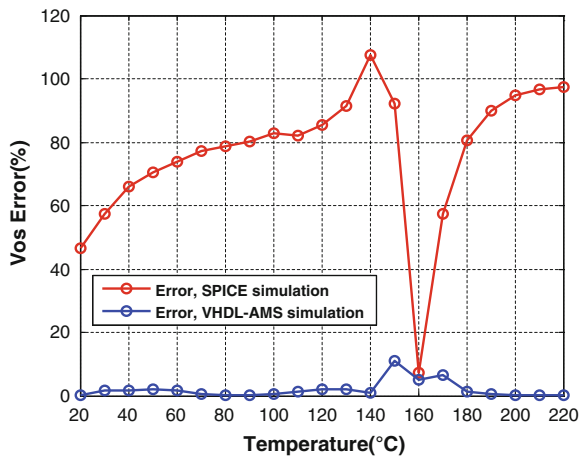
**(b)**

```

-- Affectation de la tension de décalage en fonction de la zone de
-- température
IF (temp < 140) use
  voff == voff_bt;
else
  voff == voff_ht;
end use;
end use;
...
--Affectation de la tension d'offset à V1
V1 == voff;
...
        
```

Fig. 7.32 Implantation of **voff** dependency to temperature

Fig. 7.33 Comparison of simulation results with SPICE and VHDL-AMS of  $V_{os}$  error



**Table 7.6** Comparison of different error values in different temperatures in [20 °C, 220 °C]

Temperature (°C)	SPICE-measurement error (%)	Fitting-measurement error (%)	VHDL-AMS-fitting error (%)	VHDL-AMS-measurement error (%)
20	46.11	0.66	0.004	0.66
40	65.89	1.01	0.009	0.99
80	78.64	1.33	0.032	0.306
120	85.21	0.55	1.9	2.2
160	7.33	1.06	0.34	5.14
180	94.59	0.2	0.14	1.19
220	97.3	0.0125	0.037	0.05

In Table 7.6, we compare different errors in many temperature points and we can see clearly that the error VHDL-AMS-measurement has low values. We have thus validated the implantation of the evolution of the offset voltage in temperature. The implantation of the evolution of other parameters is achieved in the same manner. We succeed finally to realize a reusable model whose performances are more accurate than the industrial SPICE model in HT.

## 7.4 Conclusion

The goal of this work was to develop an accurate op-amp model in HT for industrial applications since the used SPICE models of commercial circuits are inaccurate in HT. In the beginning of the chapter, we have shown that using conventional electronics for HT applications with compensation circuits will be a good trade-off. It will reduce the cost, minimize the size, and keep an acceptable SNR. However, the only difficulty for this scenario consists in using accurate models in HT. Actually, as the conventional electronics are manufactured for being used in temperature lower than those of HT region, their industrial models are expected to be not precise. The goal of this work was to develop a methodology in order to obtain accurate model of commercial circuits dedicated for HT applications. We focused especially in the structure of AFE used in data acquisition systems. As most of the AMS circuits of AFE are based on op-amps, we were particularly interested in the study of this device. We have considered a commercial op-amp with its SPICE model that was provided by its constructor. This model was tested in HT region by simulating  $V_{os}$  in the interval [20 °C, 220 °C]. A campaign of measurement was performed in order to evaluate the precision of the SPICE model in HT. The experimental results have shown that  $V_{os}$  has an exponential evolution in HT, whereas SPICE simulation and technical documents predict a linear evolution in HT. We have analyzed the inner structure of the SPICE model in order to understand the origin of the error in the simulation of  $V_{os}$ . We have shown that extending the industrial SPICE model to HT will be a complex task. It will require the reviewing of the structure of the model and especially the

reviewing of parameter values of the two model transistors. So, we decided to search another modeling approach that could profit from the achieved measurements of performance parameters and that could be compatible with industrial constraints. After comparing different approaches, we have chosen the behavioral modeling approach. We started with an ideal op-amp model that will be enriched gradually by adding different parameters. Once the non-ideal model written in VHDL-AMS is developed and validated, we use the measurement data in order to convert the evolution of measurement parameters to polynomial or exponential functions. These equations were inserted in the VHDL-AMS model after making some modifications in **ENTITY** and **ARCHITECTURE** parts. We have simulated then  $V_{os}$  in many temperatures of [20 °C, 220 °C] by using the final VHDL-AMS model. By comparing SPICE and VHDL-AMS errors, we have concluded that the simulation error was largely optimized. Thus, simulating industrial conventional electronics and their adaptation circuits will be possible with such VHDL-AMS model. It will be possible to simulate in a first step some op-amp-based circuits such as instrumentation amplifier, analog filter, and analog-to-digital converter.

This work has shown the usefulness of behavioral modeling approach to remedy some limitations of classical SPICE models especially in industrial applications. Such modeling approach has many advantages. First, it accelerates simulation as it minimizes the description of the structure of the circuit. It minimizes also the number of transistors whose models contains many nonlinear equations. Then, it enables to get rid of the confidentiality limitation of commercial circuits since it models the circuits as a black box and focuses on its inputs and outputs relations. Finally, there are more and more tools that implant such approach. This makes it a modern technique that could solve many limitations and difficulties in designing new and performant circuits for industrial applications. This work can be extended by studying the possibilities of developing a multi-domain model in which the mutual transfer between electrical and thermal domains is well described. Finally, a great extension and improvement of the work will be in developing an algorithm that will automate the modeling methodology and generates then the model automatically. We can profit from some works that have already studied and developed methods and algorithms that generate automatically circuits' models. The obtained result will be a behavioral model that is developed automatically by just providing experimental data.

## References

1. Ilreike, P.L., Fleetwood, D.M., King, D.B., Sprauer, D.C., Zipperian, T.E.: An overview of high-temperature electronic device technologies and potential applications. *IEEE Trans. Compon. Packag. Manuf.* **17**(4), 594–609 (1994)
2. Romanko, T.: Applications extreme design: developing integrated circuits for –55 °C to +250 °C. *EETimes/Planet Analog Magazine*, 10 Nov 2008

3. Committee on Materials for High-Temperature Semiconductor Devices: National Materials Advisory Board, Commission on Engineering and Technical Systems, National Research Council "Materials for high-temperature semiconductor devices". National Academy Press, Washington, D.C. (1995)
4. McCluskey, P., Podlesak, T., Grzybowski, R.: High Temperature Electronics. CRC Press Inc., Boca Raton (1996)
5. Willander, M., Hartnagel, H.L.: High-Temperature Electronics. Chapman & Hall, UK (1997)
6. Sharp, R.: High temperature electronics-possibilities and prospects. In: IEE Colloquium on Advances in Semiconductor Devices, London, Jan (1999)
7. Buttay, C., Planson, D., Allard, B., Bergogne, D., Bevilacqua, P., Joubert, C., Lazar, M., Martin, C., Morel, H., Tournier, D., Raynaud, C.: State of the art of high temperature power electronics. *Mater. Sci. Eng. B* **176**(4), 283–288 (2011)
8. Amalu, E.H., Ekere, N.N., Bhatti, R.S.: High temperature electronics: R&D challenges and trends in materials, packaging and interconnection technology. In: Proceedings of 2nd International Conference on Adaptive Science and Technology (ICAST), Dec 2009
9. Carter, B., Brown, T.R.: Handbook of operational amplifier applications. Application Report, Texas Instruments, SBOA092 A, Oct 2001
10. Carter, B.: Fully differential op amps made easy. Application Report, Texas Instruments, SLOA099, May 2002
11. Analog Devices.: Tutorial MT-037 Tutorial, Rev. 0. Op-amp Input Offset Voltage
12. Williams, T.: The Circuit Designer's Companion, 2 edn. Elsevier, Amsterdam (2005)
13. Burns, M., Roberts, G.W.: An Introduction to Mixed-Signal IC Test and Measurement. Oxford University Press, Oxford (2011)
14. Cheng, Y., Hu, C.: MOSFET Modeling and BSIM3 User's Guide. Springer, Berlin (1999)
15. Xi, X.J., Dunga, M., He, J., Liu, W., Cao, K.M., Jin, X., Ou, J.J., Chan, M., Niknejad, A.M.: BSIM4.3.0 MOSFET model-user's manual. Chapter 12 temperature dependency model. Department of Electrical Engineering and Computer Sciences, University of California, Berkeley (2003)
16. Nguyen, B., Smith, W.D.: Nulling Input Offset Voltage of Operational Amplifier. Texas Instruments Corp. Application Report, SLOA045, Aug 2000
17. Keilkowski, R.: Inside SPICE. McGraw-Hill, New York (1994). ISBN 0-07-913712-1
18. Wünsche, S., Clauß, C., Schwarz, P.: Electro-thermal simulation using circuit coupling. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **5**(3), 277 (1999)
19. McConaghy, T., Gielen, G.: IBMG: interpretable behavioral model generator for nonlinear analog circuits via canonical form functions and genetic programming. In: Proceedings of International Symposium on Circuits and Systems, ISCAS 2005, vol. 5, pp. 5170–5173
20. Al-Kashef, A., Zaky, M.M., Dessouky, M., El-Ghitani, H.: A case-based reasoning approach for the automatic generation of VHDL-AMS models. In: Proceedings of IEEE Behavioral Modeling and Simulation (BMAS) Workshop 2008, pp. 100–105
21. Little, S.R.: Efficient modeling and verification of analog/mixed-signal circuits using labeled hybrid petri nets. Ph.D dissertation, The University of Utah, Dec 2008
22. Little, S.R., Sen, A., Myers, C.: Application of automated model generation techniques to analog/mixed-signal circuits. In: Proceedings of 8th International Workshop on Microprocessor Test and Verification, MTV, pp. 109–115 (2007)
23. Benner, P., Hinze, M., Ter Maten, E.J.W.: Model reduction for circuit simulation. *Lecture Notes in Electrical Engineering*, vol. 74 (2011)
24. Gielen, G.: Design methodology and model generation for complex analog blocks. Chapter in "analog circuit design, RF circuits: wide band, front-ends, DAC's, design methodology and verification for RF and mixed-signal systems, low power and low voltage. Springer (2006)
25. Baccar, S., Qaisar, S.M., Dallet, D., Levi, T., Shitikov, V., Barbara, F.: Analog to digital converters for high temperature applications: The modeling approach issue. In: Proceedings of IEEE International Instrumentation and Measurement Technology Conference, Austin, May 2010

26. Baccar, S., Levi, T., Dallet, D., Shitikov, V., Barbara, F.: Modeling methodology for analog front-end circuits dedicated to high-temperature instrumentation and measurement applications. In: IEEE Transactions on Instrumentation and Measurement, vol. 60, no. 5, May 2011
27. Ashenden, P.J., Peterson, G.D., Teagarden, D.A.: The System Designer's Guide to VHDL-AMS: Analog, Mixed-Signal, and Mixed-Technology Modelling. Elsevier Science, 2003
28. E. Christen, K. Bakalar, "VHDL-AMS A Hardware Description Language for Analog and Mixed-Signal Applications", IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 46, no. 10, October 1999
29. R. Fervert, J. Haase, R. Jancke, U. Knochel, P. Schwarz, R. Kakerow, M. Darianian, "Modeling and Simulation for RF System Design", Springer 2005
30. K. Kundert, "The Designer's Guide to Verilog-AMS", Kluwer Academic Publishers, May 2004
31. Chalk, C., Zwolinski, M.: Macromodel of CMOS operational amplifier: including supply current version. Electron. Lett. **31**(17), 1398 (1995)

# Chapter 8

## Nonlinearities Behavioral Modeling and Analysis of Pipelined ADC Building Blocks

Carlos Silva, Philippe Ayzac, Nuno Horta and Jorge Guilherme

**Abstract** This chapter presents a high-speed simulation tool for the design and analysis of pipelined analog-to-digital converters (ADCs) implemented using the Python programming language. The development of an ADC simulator requires the behavior modeling of the basic building blocks and their possible interconnections to form the final converter. This chapter presents a pipeline ADC simulator tool that allows topology selection and digital calibration of the frontend blocks. Several block nonlinearities are included in the simulation, such as thermal noise, capacitor mismatch, gain and offset errors, parasitic capacitances, settling errors, and other error sources.

### 8.1 Introduction

ADCs are the key blocks in today's modern systems by providing the link between the analog world and the digital systems. Due to the extensive use of analog and mixed analog–digital operations, analog-to-digital converters (ADCs) often appear as the bottleneck in data processing applications, limiting the overall speed or precision. Thus, efficient design strategies and tools are fundamental to cope with

---

C. Silva  
Portugal Telecom, Lisbon, Portugal  
e-mail: carlos.costa.s@sapo.pt

P. Ayzac  
Thales Alenia Space, Toulouse, France  
e-mail: philippe.ayzac@thalesalieniaspace.com

N. Horta  
Instituto de Telecomunicações, Lisbon, Portugal  
e-mail: nuno.horta@lx.it.pt

J. Guilherme (✉)  
Instituto Politecnico Tomar, Tomar, Portugal  
e-mail: jorge.guilherme@ipt.pt



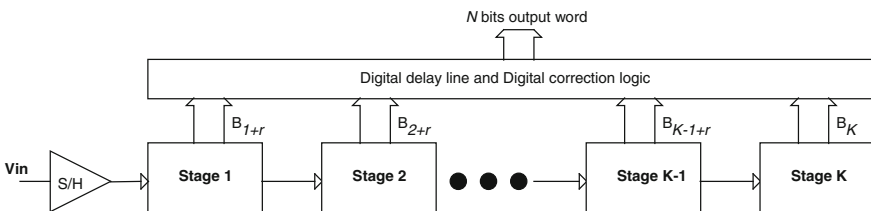
the design complexity of high-performance ADCs and with extremely large simulation times at transistor level. In the last decades, some approaches have been proposed to model, simulate, and synthesize different ADC topologies, e.g., Flash [1], Sigma-Delta [2–4], Pipeline [5, 6], Successive Approximation [7, 8], and Nonlinear [9], mainly at system-level and aiming at generating the sub-block specifications [10–14]. Technology evolution to deep nanometer integration nodes brings new challenges in terms of non-idealities; therefore, more accurate modeling and simulation techniques, as well as topology exploration facilities, are mandatory [15, 16].

This chapter describes a state-of-the-art tool for behavior simulation of pipeline ADCs [17]. The simulation tool operates through a user interface (GUI) to choose the converter topology (number of stages and resolution per stage) and provides integral and differential nonlinearities (INL and DNL) profiles, together with the output spectrum analysis (FFT). Foreground calibration can be chosen on the frontend stages to improve performance in high-resolution converters [18]. Additionally, circuit nonlinearities such as offset voltage originated in comparators and amplifiers, capacitor mismatch, amplifier gain, thermal noise, amplifier slewing and linear settling, and clock jitter are modeled [19–22] and can be defined for each stage independently. Finally, to speed up simulations, a multiprocessor option can be enabled. The tool includes a module to automatically generate the required Verilog code used to implement the digital calibration, for high-resolution pipeline ADC, up to 3 calibrated stages. Component nonlinearity as capacitor and resistor voltage-dependent second-order effects is also modeled.

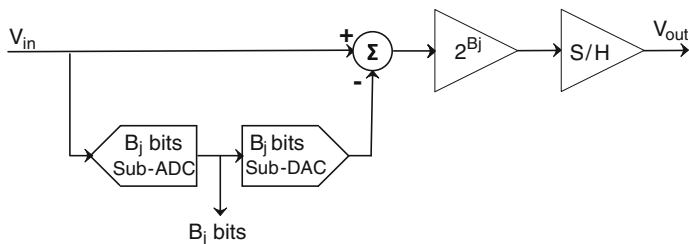
## 8.2 Pipelined ADC Structure

The pipeline architecture provides an elegant way to achieve high-speed and high-resolution A/D conversion in a CMOS process. A pipeline ADC (Fig. 8.1) consists of a cascade of several stages which could have different resolution per stage. In case of SCALES simulator, this resolution can be 1.5, 2.5, 3.5, 4.5, or 5.5 bits/stage and can be simulated up to a maximum of 16 stages plus the final Flash, which can be defined with a resolution of 2, 3, 4, or 5 bits.

Each stage digitizes its analog input with a resolution  $B_j$  and sends a residue voltage to the next stage. The use of digital correction algorithm (RSD) [18, 19]



**Fig. 8.1** Implemented pipeline ADC topology



**Fig. 8.2** Topology of a  $B_j$  bits stage on the SCALES simulator

minimizes comparator offset errors that affect the linearity of the converter. RSD algorithm uses a redundant sign bit  $r$  that is added to total number of bits of the stage. All stages have one redundancy bit, except the final stage. Each stage has  $B_i + r$  bits of resolution, in which  $B_i$  represents the effective resolution of stage and  $r$  represents the redundancy bit.

Each stage has two main components (Fig. 8.2), a sub-ADC which performs the input signal quantization and determines the digital output, and a multiplying DAC which generates a residue voltage using a switched capacitor’s array. This residue voltage is amplified by a power of two and sent to the remaining stages.

The full scale range of the converter varies from  $-V_{ref}$  to  $+V_{ref}$ . In the Flash converter, that is the basic component of the sub-ADC in a pipeline stage, the quantization is performed by a voltage divider and a set of comparators.

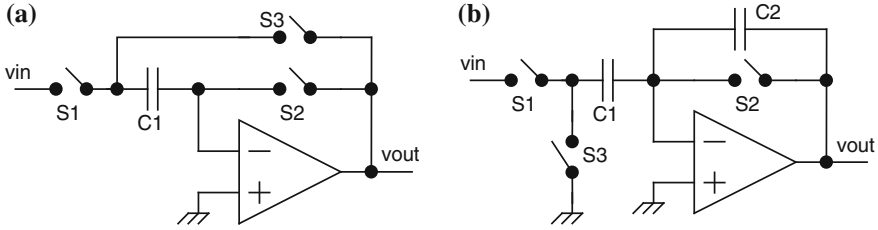
The resolution of a Pipeline A/D converter with  $k$  stages of  $m$  different individual resolutions  $B_j$ , where  $B_k$  is the resolution of the last stage (final Flash), is given by:

$$N = \sum_{j=1}^m k_j B_j + B_k \tag{8.1}$$

Most errors that affect a Pipeline A/D converter are originated in the multiplying DAC, MDAC. In addition to  $kT/C$  noise, other errors are analyzed and modeled by SCALES [17]: capacitance errors (capacitor mismatch, nonlinearity and charge injection); time errors (settling and jitter errors); amplifier errors (offset, slew rate, finite open-loop gain, nonlinear DC gain); and comparators offset [18, 19]. Each non-ideality can be activated individually or simultaneously. A description of the implemented nonlinearities is done on the following paragraphs.

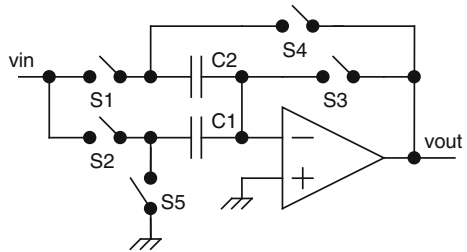
### 8.2.1 Sample and Hold

The input signal has to be sampled by the first stage of the pipeline or by an independently S&H block. This S&H can be designed to have unity gain or a gain higher than one. The gain value is dependent on the capacitor network usually build



**Fig. 8.3** Sample and hold circuit with gain = 1, (1 capacitor—**a**; 2 capacitor—**b**) [20]

**Fig. 8.4** Sample and hold circuit with gain >1 [20]



using one or two capacitors determined by noise constrains. The sampling capacitor has a value that is a multiple of the  $C_U$  (Capacitor Unit). The tool allows the user to build the pipeline converter with or without the S&H stage, and select the gain as unity or gain >1 as shown in Figs. 8.3 and 8.4. The use of an S&H affects the total input referred noise  $V_{ni}$  as given by [20–22]:

$$V_{ni} = \sqrt{\frac{v_{no,S/H}^2}{G_0^2} + \frac{v_{no,Stage1}^2}{G_0^2 G_1^2} + \frac{v_{no,Stage2}^2}{G_0^2 G_1^2 G_2^2} + \dots + \frac{v_{no,Stage(k-1)}^2}{G_0^2 G_1^2 G_2^2 \dots G_{k-1}^2}} \quad (8.2)$$

where  $V_{no}$  is the corresponding output thermal noise voltage in each stage, and  $G_i$  ( $G_0$  is the S/H gain) is the gain of the residue amplifier, and  $k$  is the number of pipeline stages.

- If selectable S&H gain = 1: (1 capacitor):  $G_0 = 1.0$
- If selectable S&H gain = 1: (2 capacitor):  $G_0 = C_1/C_2 \approx 1.0$
- If selectable S&H gain >1:  $G_0 = 1 + C_1/C_2$

### 8.2.2 1.5 Bits Stage

The 1.5-bit stage is a basic stage of the pipeline that can be extrapolated to higher stage resolutions. Other bit resolutions are similar, with the corresponding increase in the number of components such as the number of sampling capacitors  $C_s$ , the

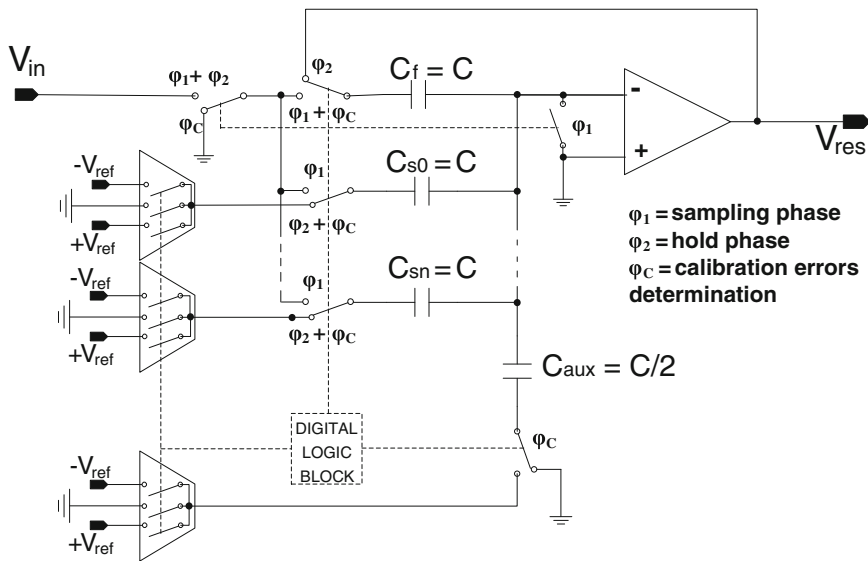


Fig. 8.5 Pipeline 1.5-bit stage

number of comparators and consequent number of transition levels in the voltage divider.

Figure 8.5 presents the 1.5-bit stage implemented in SCALES simulator. The diagram also includes a logical module used in the calibration process. It contains the control unit for the associated switches and the capacitor  $C_{aux}$  that generates a forced offset used during the referred calibration process. The  $\varphi_C$  switch position is controlled by this logical module and is only used if calibration process was checked for active simulation. During normal conversion process,  $C_{aux}$  is kept with its lower plate connected to GND.

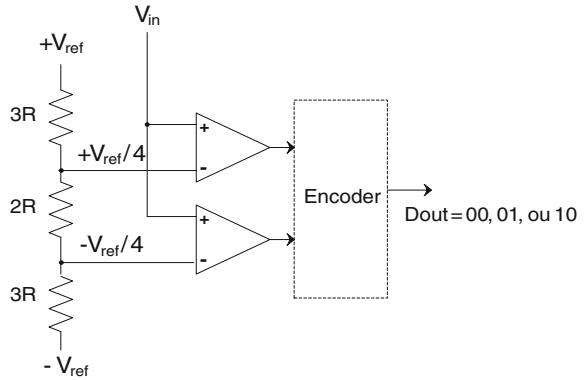
### 8.2.3 Sub-ADC

The sub-ADC is a simple Flash converter with an extra comparator to allow the implementation of digital correction. The structure of the 1.5-bit sub-ADC is presented in Fig. 8.6.

The digital output of sub-ADC is given by:

$$D_{out} = \begin{cases} 00 & \text{if } V_i < -\frac{V_{ref}}{4} \\ 01 & \text{if } -\frac{V_{ref}}{4} \leq V_i \leq \frac{V_{ref}}{4} \\ 10 & \text{if } V_i > \frac{V_{ref}}{4} \end{cases} \quad (8.3)$$

**Fig. 8.6** Structure of the 1.5-bit sub-ADC

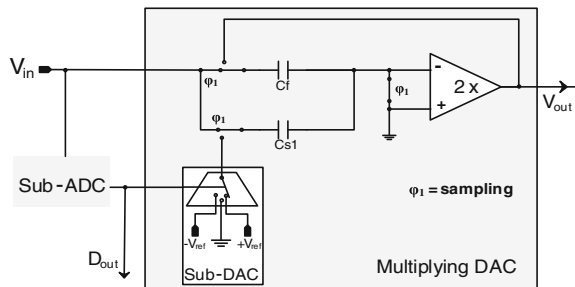


### 8.2.4 MDAC (Multiplying DAC)

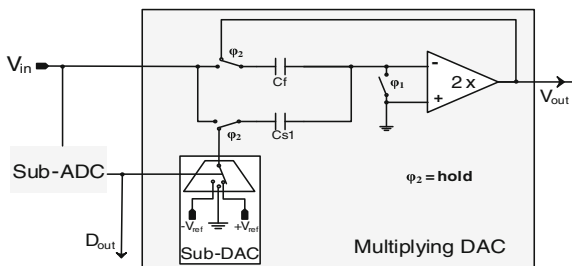
The MDAC module calculates the residue of the stage and amplifies it before delivering to the next stage. It consists of three components with separate functions, but that complement each other. The first element is a sub-DAC converter (D/A converter, which converts digital data into analog signals). It converts the output of the sub-ADC to an analog signal that will be subtracted to the input voltage, in the second element, the Sample and Hold, using an array of switched capacitors. The difference signal is amplified in the third element, the operational amplifier. The output residue voltage is then evaluated through the remaining stages of the Pipeline.

The sampling phase is presented in Fig. 8.7. An input signal  $V_{in}$  is connected to  $C_s$  and  $C_f$  that have their bottom plates connected to GND, and in the same time, the sub-ADC converts the analog input into a digital code. During the hold phase (Fig. 8.8), the bottom plate of  $C_f$  is connected in a closed loop mode to the output of the amplifier. To the bottom plates of the sampling capacitors  $C_s$ , it is applied  $+V_{ref}$ ,  $-V_{ref}$  or  $GND$ , depending on the digital output of the stage, determined by the sub-ADC. The output voltage is calculated by subtracting the equivalent analog value of the digital output code of sub-DAC with the input signal, and amplifying this residue such that [4]:

**Fig. 8.7** Sampling phase on the 1.5-bit stage



**Fig. 8.8** Hold phase on the 1.5-bit stage



$$V_{out} = \begin{cases} \left(1 + \frac{C_s}{C_f}\right) V_{in} - \frac{C_s}{C_f} V_{ref} & \text{if } V_{in} > \frac{+V_{ref}}{4} \\ \left(1 + \frac{C_s}{C_f}\right) V_{in} & \text{if } \frac{-V_{ref}}{4} \leq V_{in} \leq \frac{+V_{ref}}{4} \\ \left(1 + \frac{C_s}{C_f}\right) V_{in} + \frac{C_s}{C_f} V_{ref} & \text{if } V_{in} < \frac{-V_{ref}}{4} \end{cases} \quad (8.4)$$

Figure 8.9 presents the residue of a 1.5-bit stage when a ramp is applied in the input of the pipeline stage.

The residue voltage can be determined by [20]:

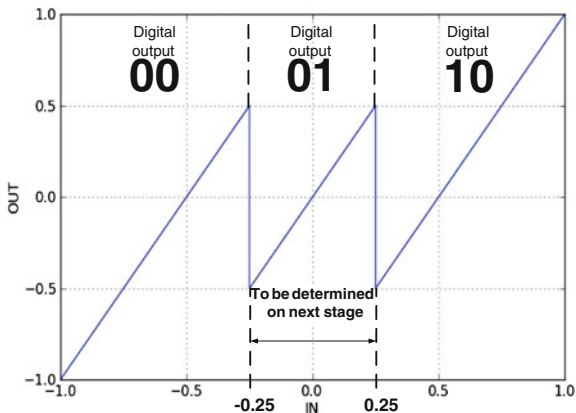
$$V_{out,i} = G_i \cdot V_{in} + D_i \cdot V_{ref} \quad (8.5)$$

where  $D_i$  is an integer corresponding to the output  $B_i$  of the sub-ADC and:

$$D_i \in [-(2 \cdot B_i - 1), +(2 \cdot B_i - 1)] \quad (8.6)$$

The ideal gain  $G$  is given by [18, 22]:

**Fig. 8.9** Residue of a 1.5-bit stage (ramp input)



$$G = 2^{B_i+1-r} \text{ or } G = 2^n \quad (8.7)$$

where  $n$  is the number of sampling capacitors  $C_s$  of the stage, and  $r$  is the value of the redundant bit for digital correction. Real gain  $G_i$  is different from ideal gain  $G$  due to the capacitors mismatch and is given by [20, 22]:

$$G_i = \frac{C_f + \sum_{j=0}^{n-1} C_{s,j}}{C_f} \quad (8.8)$$

where  $n = (2^N - 1)$  is the number of sampling capacitors of  $i$  stage. The multiplying factor  $D_i$  used to determine  $V_{\text{dac}} = (D_i \cdot V_{\text{ref}})$  can be given by:

$$D_i = \frac{\sum_{j=0}^{n-1} (m_j \cdot C_{s,j})}{C_f} \quad (8.9)$$

where  $m$  is a multiplying factor applied to the values of the sampling capacitors  $C_s$ , equal to +1, 0, or -1, dependent of the digital output code of sub-ADC ( $B_i + r$ ) and is given by:

$$\begin{cases} m_i = -1 & \text{if } D_i < -1 \\ m_i = 0 & \text{if } D_i = 0 \\ m_i = +1 & \text{if } D_i > 1 \end{cases} \quad (8.10)$$

Finally, the residue output of the stage can be rewritten as [4]:

$$V_{\text{out},i} = \frac{C_f + \sum_{j=0}^{n-1} C_{s,j}}{C_f} \cdot V_{\text{in}} + \frac{\sum_{j=0}^{n-1} (m_j \cdot C_{s,j})}{C_f} \cdot V_{\text{ref}} \quad (8.11)$$

## 8.3 Errors on a Pipeline A/D Converter

### 8.3.1 Non-idealities

Most errors that affect a pipeline A/D converter have origin in the MDAC. The analysis of those errors in this block is fundamental in the sense of developing calibration algorithms that increase the performance of the pipeline converter. Several types of error can be simulated:

- Thermal noise  $kT/C$
- Comparators offset  $V_{\text{OS}}$

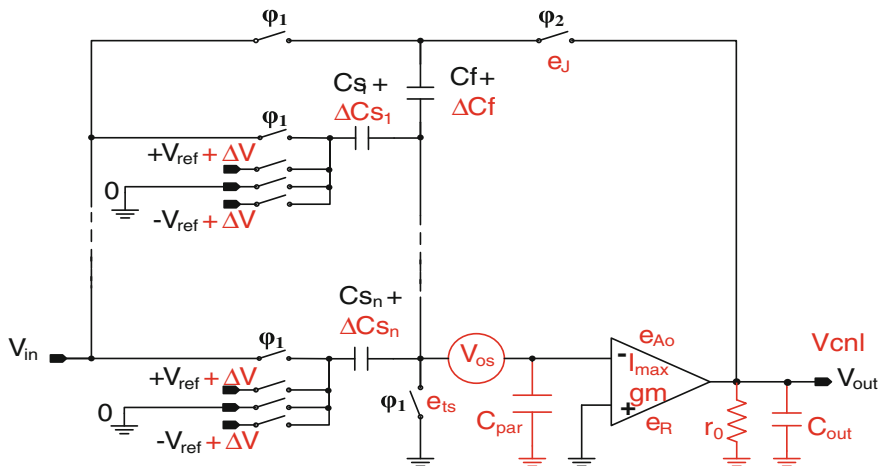


Fig. 8.10 Simplified diagram of a generic MDAC with error sources in red

- Comparators errors (offset, resistors mismatch)
- Capacitance errors (mismatch, charge injection, nonlinearity)
- Time errors (Settling error, jitter error)
- Amplifier errors (offset, limited GBW, slew rate, finite open-loop gain), non-linear gain, and parasitic capacitance)

Some of those non-idealities are represented in Fig. 8.10, a simplified diagram of a generic MDAC that includes the simulated error sources. Amplifier noise and offset are always referred to the input, therefore, independent of the stage gain. Comparators' offset is also referred to the input.  $V_{cnl}$  represents the effect of CNL (capacitors nonlinearity) at the output of the stage.

### 8.3.2 Offset Errors

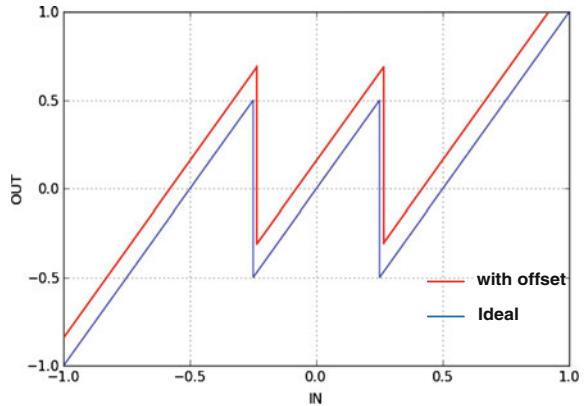
Comparators offset are defined by a random value based on an average one defined by user. The offset in comparators is very important; however, with the use of digital correction, we can minimize their impact. The maximum allowed offset in comparators is given by [19]:

$$V_{osC(max)} = \pm \frac{r}{2^{B_i+r}} \cdot V_{ref} \tag{8.12}$$

The residue amplifier offset  $V_{osC}$  is measured on his negative input during the sampling phase. Capacitors  $C_s$  and  $C_f$  have their bottom plate connected to GND and the amplifier offset does not affect their charge. During the hold phase, however, the effect of the amplifier offset on his output is given by [20], where  $C_{aux}$  is



**Fig. 8.11** Effect of amplifier offset errors on a 1.5-bit stage



only taken into account on simulations with calibration activated, and  $V_{osA,i}$  is the offset voltage at the amplifier output:

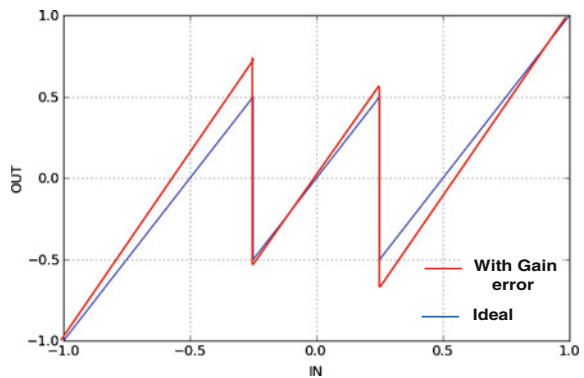
$$V_{osA,i} = G_i \cdot V_{os} = 1 + \frac{\sum_{j=0}^{n-1} C_{s,j} + C_{aux}}{C_f} \cdot V_{os} \tag{8.13}$$

$V_{os}$  is an average value defined by user, on the simulator GUI. The effect of the offset on a 1.5-bit stage output is shown in Fig. 8.11.

### 8.3.3 Gain Error

The gain error is a multiplicative factor that affects the input signal of a pipeline A/D converter. The most common source of gain error is due to the capacitors mismatch and reduced amplifier gain. The incomplete establishment of the sampled

**Fig. 8.12** Effect of gain errors on a 1.5-bit stage



signal also creates a gain error. Another gain error source is the reference voltage mismatch but his mismatch is typically so low that it can be negligible (Fig. 8.12).

### 8.3.4 Open-Loop Gain Errors

Some static errors also affect the output voltage of the stage. The effect of the open-loop gain error is represented in Fig. 8.13.

The amplifier finite open-loop gain  $A_o$  and parasitic capacitance  $C_{par}$  affect output of the stage. So, the output  $V_{out}$  is given by [20]:

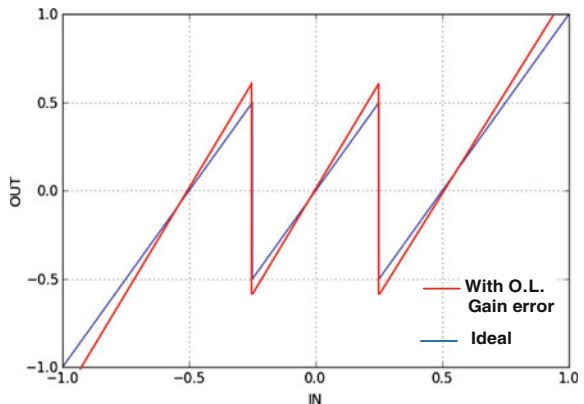
$$\begin{aligned}
 &V_{out,j} \cdot \left( Cf + \sum_{j=0}^{n-1} Cs_j + C_{aux} \right) \\
 &= V_{out,j} \cdot \left[ \left( 1 + \frac{1}{A_O} \right) Cf + \frac{\sum_{j=0}^{n-1} Cs_j + C_{aux}}{A_O} + \frac{C_{par}}{A_O} \right] + V_{ref} \cdot \left( \sum_{j=0}^{n-1} m_j \cdot Cs_j \right) \\
 \Leftrightarrow V_{out,j} &= \left[ \frac{Cf + \sum_{j=0}^{n-1} Cs_j + C_{aux}}{Cf} V_{os,j} - \frac{\sum_{j=0}^{n-1} m_j \cdot Cs_j + C_{aux}}{Cf} V_{ref} \right] \cdot \frac{1}{1 + \frac{1}{A_O f}}
 \end{aligned} \tag{8.14}$$

When  $1/A_O \cdot f \gg 1$ , Eq. (8.14) can be expressed as [4]:

$$\begin{aligned}
 V_{out,j} &= \left[ \frac{Cf + \sum_{j=0}^{n-1} Cs_j + C_{aux}}{Cf} V_{os,j} - \frac{\sum_{j=0}^{n-1} m_j \cdot Cs_j + C_{aux}}{Cf} V_{ref} \right] \cdot 1 - \frac{1}{A_O \cdot f}, \\
 A_O &= 10^{\frac{\text{Gain[dB]}}{20}}
 \end{aligned} \tag{8.15}$$

Amplifier finite open-loop gain  $A_o$  affects the output by an error  $e_{A_o}$  that is given by [4]:

**Fig. 8.13** Effect of finite open-loop gain errors on a 1.5-bit stage



$$e_{A_o} = \frac{1}{A_o \cdot f} \quad (8.16)$$

where  $f$  is the feedback factor given by [20]:

$$f = \frac{Cf}{Cf + \sum_{j=0}^{n-1} Cs_j + C_{\text{par}} + C_{\text{aux}}} \quad (8.17)$$

On Eqs. (8.16) and (8.19),  $C_{\text{aux}}$  is only important when calibration is activated during simulation; otherwise, its value is considered equal to 0.

### 8.3.5 Capacitors Mismatch

Capacitors mismatch errors are one of the main causes of nonlinearity in the transfer function of the converter. The mismatch value is dependent on the used CMOS technology. By introducing capacitors mismatch  $\Delta C$  in the equation of the output residue voltage, it leads to [20]:

$$V_{\text{out},i} = \frac{Cf + \Delta Cf + \sum_{j=0}^{n-1} (Cs_j + \Delta Cs_j)}{Cf + \Delta Cf} V_{\text{in},i} - \frac{\sum_{j=0}^{n-1} [m_j \cdot (Cs_j + \Delta Cs_j)]}{Cf + \Delta Cf} V_{\text{ref}} \quad (8.18)$$

In the multiplicative part that affects  $V_{\text{in}}$ , capacitors mismatch originates a gain error, and the part that affects  $V_{\text{ref}}$  originates an amplitude error in the comparators thresholds levels. The capacitors standard deviation is modeled based on a unit capacitor value defined by the user with the following expression:

$$\sigma_C(C) = \frac{AC}{\left( \frac{-4Cp + \sqrt{16Cp^2 + 4Ca \cdot C}}{2Ca} \right) \cdot \sqrt{2} \cdot 100} \quad (8.19)$$

where  $AC$ ,  $Cp$ , and  $Ca$  are technology-dependent parameters defined by the user and are, respectively, the typical mismatch (%), the capacitance per perimeter (fF/ $\mu\text{m}$ ) and the capacitance per area (fF/ $\mu\text{m}^2$ ).

### 8.3.6 Flash Resistors Mismatch

Flash resistor mismatch affects sub-ADC's threshold voltages, creating undesired variations on threshold levels, on all comparators. The unity resistor is defined by

the user (20 kΩ by default were considered as typical value), and one of the resistors dimensions must be considered as fixed, and the other dimension is determined based on the first dimension and on the unit resistor (UR) value. Some parameters are technology dependent, like AR (resistors mismatch [% of UR]) and the sheet resistance  $R_S$  [kΩ/\*]. The Flash resistor ladder is divided into several groups of resistors. Usually, the middle groups have 2 resistors (2R) and the upper and lower groups are formed with 3 resistors (3R), as we can see in Fig. 8.14 for a 1.5-bit sub-ADC stage:

When the user fixes  $L$  (Length), the width  $W$  is determined by:

$$W = \frac{R_S \cdot L}{UR} \tag{8.20}$$

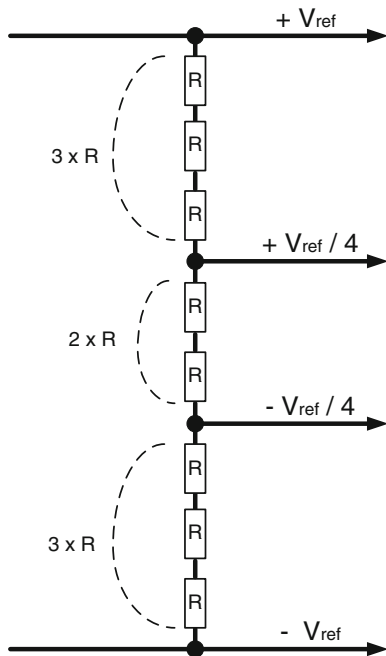
When the user fixes  $W$  (Width), the length  $L$  is determined by:

$$L = \frac{UR \cdot W}{R_S} \tag{8.21}$$

The variation of the resistors values can be calculated by:

$$\sigma_R = \frac{AR \times 10^{-6}}{\sqrt{(W - \Delta W) \cdot (L - \Delta L)}} \cdot 100 \tag{8.22}$$

**Fig. 8.14** Resistors ladder of a 1.5-bit stage



Resistance also varies with temperature, and these non-idealities are taken into account on the simulation. In practice, temperature is the same for all resistors, on all sub-ADCs, so temperature does not affects the threshold levels. We calculate the effect of temperature on resistors only to obtain the most accurate values for the resistors. The temperature effect on resistors, considering a room temperature of  $T_r$  [K], follows the expression:

$$R = 1 + TC_1 \cdot (T - T_r) + TC_2 \cdot (T - T_r)^2 \tag{8.23}$$

with  $TC_1$  and  $TC_2$  the first- and second-order temperature coefficients.

### 8.3.7 Slew Rate, -3 dB Corner Frequency and Settling Error

In the beginning of the Hold mode, we have a settling phase where the amplifier outputs the maximum current  $I_{max}$ . Over time, the transition level tends to approximate real input value, limited only by  $gm$  of the amplifier and the effective capacitance  $C_L$  (Fig. 8.15).

The slew rate is defined by the user, but can also be calculated by [5]:

$$SR = \frac{I_{max}}{C_L + C_{par} + Cf} \quad \text{with} \quad C_L = \sum_{j=0}^{n-1} Cs_j + C_{aux} + Cf \tag{8.24}$$

$C_{par}$  and  $C_{out}$  are determined as a percentage ( $\delta$ ) of the sum of the sampling capacitors  $C_s$  and  $C_{aux}$ , and  $\delta$  is defined by the user (different values of  $\delta$  can be defined for  $C_{par}$  and  $C_{out}$ ):

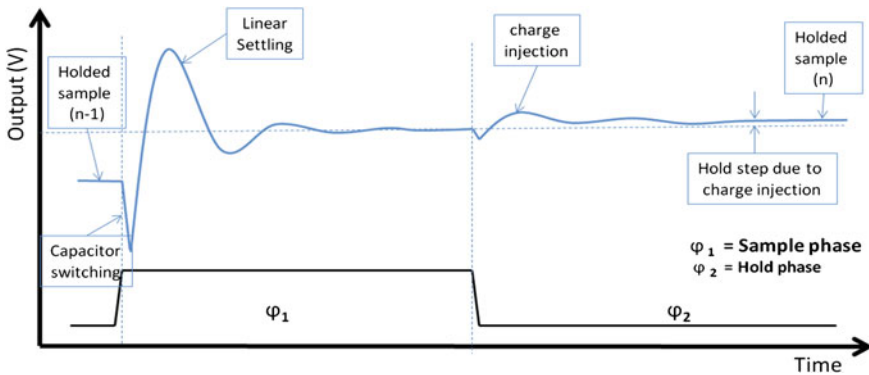


Fig. 8.15 Amplifier settling on hold mode [20]

$$C_{\text{par}}, C_{\text{out}} = \left( \sum_{j=0}^{n-1} C_{S_j} + C_{\text{aux}} \right) \times \delta \quad (8.25)$$

And  $C_{L,\text{total}}$  is given by:

$$C_{L,\text{total}} = C_L + C_{\text{out}} \quad (8.26)$$

where  $C_{\text{out}}$  is the parasitic capacitance of the amplifier. During exponential settling, the output of the stage is given by [22]:

$$V_{\text{out},i}(t) = (1 - e^{-\omega_{-3\text{dB}} \cdot t}) \cdot V'_{\text{out},i} \quad (8.27)$$

where the  $-3$  dB corner frequency  $\omega_{-3\text{dB}}$  is given by [20]:

$$\omega_{-3\text{dB}} = \omega_u \cdot f = \frac{gm}{C_{L,H}} \cdot f \quad (8.28)$$

The effective charge capacitance during Hold mode,  $C_{L,H}$ , is given by:

$$C_{L,H} = C_L + C_{\text{out}} + \frac{Cf \left( \sum_{j=0}^{n-1} C_{S_j} + C_{\text{par}} + C_{\text{aux}} \right)}{Cf + \sum_{j=0}^{n-1} C_{S_j} + C_{\text{par}} + C_{\text{aux}}} \quad (8.29)$$

So, the  $-3$  dB corner frequency expression can be rewritten as:

$$\omega_{-3\text{dB}} = \omega_u \cdot f = \frac{gm}{\frac{C_{L,\text{total}}}{f} + \sum_{j=0}^{n-1} C_{S_j} + C_{\text{par}} + C_{\text{aux}}} \quad (8.30)$$

Settling time is dependent of exponential time  $t_{\text{exp}}$  and slew rate and is given by:

$$t_{\text{st}} = \frac{1}{3 \cdot f_s} - \frac{V_{\text{ref}}}{2 \cdot \text{SR}} \times 10^{-6} \quad (8.31)$$

The settling error that affect output voltage residue of the stage is given by [4]:

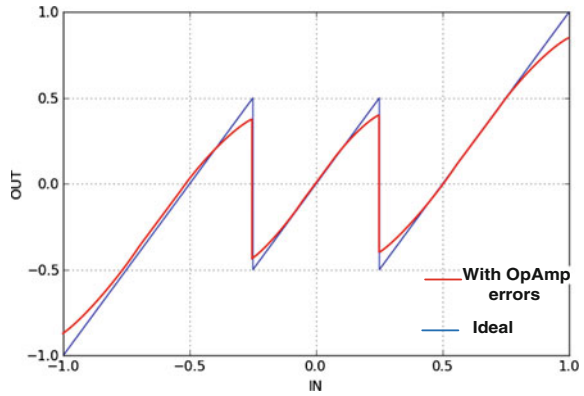
$$e_{\text{ts}} = e^{-\omega_{-3\text{dB}} \cdot t_{\text{st}}} \quad (8.32)$$

Assuming a finite open-loop gain  $e_{A_o}$ , both errors affect output voltage of the stage as follows [4]:

$$V_{\text{out},i} = \left( V'_{\text{out},i} + V_{\text{os},i} \right) (1 - e_{A_o}) \cdot (1 - e_{\text{ts}}) \quad (8.33)$$

where  $V'_{\text{out},i}$  is the ideal output of the stage. The effect of those errors in the output of the stage is presented in Fig. 8.16.

**Fig. 8.16** Effects of settling and finite amplifier gain errors on a 1.5-bit stage



### 8.3.8 Thermal Noise $kT/C$

Thermal noise affects the SNR of the converter. The source of thermal noise is the noise generated in active circuits such as amplifiers and in the switched capacitor circuits. The power of the generated noise is given by (8.34), based on an average value  $V_{nA}$  defined by user:

$$e_R = 1 + \frac{\sum_{j=0}^{n-1} C_{Sj} + C_{aux}}{C_f} \cdot \sqrt{V_{nA}^2 + \frac{k \cdot T}{C_f + \sum_{j=0}^{n-1} C_{Sj} + C_{aux}}} \tag{8.34}$$

### 8.3.9 Jitter Error

The jitter or aperture jitter is originated by uncertainty in the exact aperture time and can originate noise that can affect the effective amplitude of the hold signal, due to the effective time when the signal was sampled. To be certain that the jitter error does not affect the output; the input signal must not varies more than  $\frac{1}{2}$  LSB (least significant bit) during the jitter time. For an input signal  $V_{in}$  [20, 21]:

$$V_{in} = A \cdot \sin(2\pi \cdot f_i \cdot t) \tag{8.35}$$

Jitter time must be less than:

$$\frac{dV}{dt} - 2\pi \cdot f_i \cdot A \cdot \cos(2\pi \cdot f_i \cdot t) < \frac{\pm 1/2LSB}{t_{aj}} \tag{8.36}$$

where  $A$  is the amplitude of half full scale, and  $t_{aj}$  is the jitter time. As  $\frac{1}{2}$  LSB is given by  $A/2^N$ , the maximum frequency that jitter noise does not affect conversion process is given by [21]:

$$f < \frac{1}{2\pi \cdot 2^N \cdot t_{aj}} \quad (8.37)$$

The error due to sampling jitter is proportional to the amplitude of the input signal and is most significant for larger amplitudes. If we apply a sinusoid as input signal with amplitude  $V_{FS}/2$  and a frequency  $f_i$ , a sampling jitter with a deviation  $\sigma_a$  results in an error voltage given by:

$$v_{\text{rms},a} = V_{FS} \cdot \pi \cdot f_i \cdot \sigma_a \quad (8.38)$$

that originates a noise power at output given by:

$$e_{J,\sigma_a}^2 = \frac{v_{\text{rms},a}^2}{2} = \frac{\pi^2}{2} \cdot V_{FS}^2 \cdot f_i^2 \cdot \sigma_a^2 \quad (8.39)$$

The jitter error is determined based on the average jitter time  $t_j$  defined by user as:

$$e_J = \sqrt{\frac{(gm \cdot t_j)^2}{2 \cdot e^{-(2 \cdot \sigma_{-3dB} \cdot t_{\text{exp}})}}} \quad t_{\text{exp}} = \frac{1}{3 \cdot f_s} \quad (8.40)$$

that must be added to quantization and thermal noise and affect negatively the SNR.

### 8.3.10 Capacitors Nonlinearity

For any capacitor  $C_k$ , the capacitor real value is dependent on the applied voltage due to capacitors nonlinearity and the charge  $Q$  that can be modeled by the following expression [9]:

$$Q = (C_k, V_{\text{in}}) \cdot V_{\text{in}} = C_k \cdot (1 + \alpha \cdot V_{\text{in}} + \beta \cdot V_{\text{in}}^2 + \dots) \cdot V_{\text{in}} \quad (8.41)$$

$\alpha$  and  $\beta$  are the linear and quadratic capacitance voltage dependence factors of the capacitor. Applying above equation to the output of stage MDAC, the effect of capacitors nonlinearity can be modeled as [23]:



$$\begin{aligned}
& V_{\text{cnl}}^3 + \frac{\alpha}{\beta} V_{\text{out}}^2 + \frac{1}{\beta} V_{\text{out}} \\
& = G \cdot V_{\text{in}} \left( V_{\text{in}}^2 + \frac{\alpha}{\beta} V_{\text{in}} + \frac{1}{\beta} \right) - V_{\text{ref}} \cdot V_{\text{dac}} \left[ (V_{\text{ref}} \cdot V_{\text{dac}})^2 + \frac{\alpha}{\beta} (V_{\text{ref}} \cdot V_{\text{dac}}) + \frac{1}{\beta} \right]
\end{aligned}$$

doing:

$$G \cdot V_{\text{in}} \left( V_{\text{in}}^2 + \frac{\alpha}{\beta} V_{\text{in}} + \frac{1}{\beta} \right) - V_{\text{ref}} \cdot V_{\text{dac}} \left[ (V_{\text{ref}} \cdot V_{\text{dac}})^2 + \frac{\alpha}{\beta} (V_{\text{ref}} \cdot V_{\text{dac}}) + \frac{1}{\beta} \right] = c$$

we have:

$$V_{\text{cnl}}^3 + \frac{\alpha}{\beta} V_{\text{cnl}}^2 + \frac{1}{\beta} V_{\text{cnl}} - c = 0 \tag{8.42}$$

where  $V_{\text{cnl}}$  is the ideal output affected by capacitors nonlinearity. The resulting third degree equation must be solved in order to  $V_{\text{cnl}}$  and  $c$  is given by:

$$c = - \frac{Q}{\beta \cdot \sum_{j=0}^{n-1} C S_j} \tag{8.43}$$

### 8.3.11 Effect of Errors in the Residue Stage Output

On the SCALES simulator, the various errors previously described were taken into account as can be seen in the equation, implemented in code [17, 20]:

$$V_{\text{out},i} = (V_{\text{out},i} + V_{\text{osA}} + e_{\text{R}} + e_{\text{J}}) \cdot (1 - e_{\text{Ao}}) \cdot (1 - e_{\text{ts}}) \tag{8.44}$$

where  $V_{\text{osA}}$  is the amplifier offset,  $e_{\text{R}}$  is the thermal noise,  $e_{\text{J}}$  is the jitter error,  $e_{\text{Ao}}$  is the open-loop finite gain error of the amplifier, and  $e_{\text{ts}}$  is the settling time error.

### 8.3.12 Digital Correction

Digital correction is an algorithm that uses a coding system with a redundant digit  $r$  at each stage of A/D converters and which enables to improve the accuracy of the conversion, controlling the offset errors, provided that their value does not exceed certain limits, dependent on the parameter  $r$  and voltage  $V_{\text{ref}}$ , and avoiding the saturation of the output. Table 8.1 shows the method used to correct the output of the pipeline A/D converter, where the less significant bit of a stage is added with the most significant bit of the next stage (for  $r = 1$ ). The less significant bit of the final stage, the final flash, is not corrected. The last stage usually does not have a redundant bit ( $r = 0$ ). Table 8.2 presents the characteristics of stages up to 3.5 bits.

**Table 8.1** Digital code correction with RSD algorithm

	B <sub>1(MSB)</sub>	...	B <sub>1(LSB)</sub>							
			B <sub>2(MSB)</sub>	...	B <sub>2(LSB)</sub>					
					...	...	...			
							B <sub>k-1(MSB)</sub>	...	B <sub>k-1(LSB)</sub>	
+								B <sub>k (MSB)</sub>	...	B <sub>k(LSB)</sub>
	D <sub>N</sub>	D <sub>N-1</sub>	D <sub>N-2</sub>	...	...	...	...	...	D <sub>1</sub>	D <sub>0</sub>

**Table 8.2** Basic characteristics of pipeline stages

Stage resolution	r	B <sub>i</sub>	Q <sub>i</sub>	b <sub>i</sub>	n <sub>max</sub>	Quantization levels V <sub>Q<sub>i,n</sub></sub>	Output bits B <sub>i</sub>
1 bit	0	1	1	2	0	0	0; 1
1.5 bits	1	1	2	3	0	±0,25	00; 01; 10
2 bits	0	2	3	4	1	0; ±0,5	00; 01; 10; 11
2.5 bits	1	2	6	7	2	±0.125; ±0.375; ±0.625	000; 001; 010; 011; 100; 101; 110
3 bits	0	3	7	8	3	0; ±0.25; ±0.5; ±0.75	000; 001; 010; 011; 100; 101; 110; 111
3.5 bits	1	3	14	15	6	±0.0625; ±0.1875; ±0.3125; ±0.4375; ±0.5625; ±0.6875; ±0.8125	0000; 0001; 0010; 0011; 0100; 0101; 0110; 0111; 1000; 1001; 1010; 1011; 1100; 1101; 1110

### 8.4 Simulated Converter Characteristics

#### 8.4.1 Nonlinearity Errors—Sinusoidal Input Signal

*Histogram method to determine DNL and INL from a sinusoidal input signal*

This method is used to obtain INL and DNL data directly from the cumulative histogram. Due to the high number of samples necessary to obtain more accurate data, dependent on density per code, it is necessary to determine a minimum number of samples to ensure that the data obtained have the desired accuracy, within a specified level of confidence. The expression that allows us to calculate the minimum number of samples is [24, 25]:

$$M_{\min} = \frac{\pi \cdot 2^{N-1} \cdot Z_{\alpha/2}^2}{\beta^2} \tag{8.45}$$

where  $N$  is the resolution of the converter, and  $\beta$  is the pretended bit resolution for DNL determination. For a confidence level of 99 %, the value of  $Z_{\alpha/2}$  is available in normal distribution tables and has the value of 2.576 [24].

### 8.4.2 Total Harmonic Distortion (THD)

Errors derived from dynamic behavior of the converter and INL contribute greatly to harmonic distortion in the converter during the signals conversion. The total harmonic distortion (THD) is the RMS value of the sum of all harmonics in the spectrum of the output signal and is obtained by means of FFT algorithm. By definition of THD, all harmonics would be really summed, but due to the lower amplitude of higher order harmonics, only the low-order harmonics are summed. In a perfect sinusoid, summing to the third order can be sufficient; however, it may be necessary to include a few more harmonic values. So, THD is given by [21]:

$$\text{THD} = \frac{\sqrt{\sum_{n=2}^i A_{n \cdot f_{in}}^2(\text{RMS})}}{A_{f_{in}}(\text{RMS})} \quad (8.46)$$

where  $A_{f_{in}}(\text{RMS})$  is the RMS amplitude of fundamental frequency,  $A_{n \cdot f_{in}}(\text{RMS})$  is the RMS amplitude of the harmonics, from order  $n = 2$  to order  $n = i$ . From the spectrum of the FFT, the value of THD in dB can also be obtained using the expression [21]:

$$\text{THD}_{\text{dB}} = 10 \log \left( \frac{\text{Total Harmonics power}}{\text{Power of fundamental}} \right) \quad (8.47)$$

On the other hand, those bins, including those of the fundamental frequency, are not considered for the determination of the overall power due to noise and distortion, necessary for the calculation of SNDR.

### 8.4.3 Total SNR

The signal-to-noise ratio (SNR) accounts for the total output noise in the signal. It includes several types of noises. Using the FFT plot, total SNR in dB's can be obtained by:

$$\text{SNR}_{\text{Total(dB)}} = 10 \log \left( \frac{\text{Power of fundamental frequency}}{\text{Total power due to noise}} \right) \quad (8.48)$$

For a given stage  $i$ , total SNR due to quantization errors, jitter errors, and thermal noise can be expressed by [25]:

$$\text{SNR}_{\text{Total},i} = 10 \cdot \log \left( 1 / \left( \frac{1}{10^{\frac{\text{SNR}_{J,i}}{10}}} + \frac{1}{10^{\frac{\text{SNR}_{Q,i}}{10}}} + \frac{1}{10^{\frac{\text{SNR}_{R,i}}{10}}} \right) \right) \quad (8.49)$$

#### 8.4.4 Signal-to-Noise Distortion Ratio (SNDR)

SNDR (*Signal-to-Noise Distortion Ratio*), also designated as SINAD, represents the ratio between the RMS amplitude of the signal and the sum of the RMS amplitude of the noise with the initial harmonics of the THD, usually from the second to the fifth order). SNDR in dB's allows evaluating the quality of the dynamic range of A/D converters and is given by [26]:

$$\text{SNDR}_{\text{dB}} = 10 \cdot \log \left( \frac{\text{Power of fundamental frequency}}{\text{Power of noise} + \text{Power of all harmonics}} \right) \quad (8.50)$$

#### 8.4.5 Effective Number of Bits (ENOB)

Another characteristic that is used often in dynamic analysis of a converter is the effective number of bits (ENOB) which serves as a comprehensive indicator of accuracy of the A/D converter, for a given input signal frequency and sampling frequency. The converter ENOB is calculated using the following expression, where  $N$  is the number of bits [21]:

$$\text{ENOB} = N - \log_2 \left( \frac{\text{Mean amplitude of noise}_{(\text{RMS})}}{\text{Amplitude of quantization noise}_{(\text{RMS})}} \right) \quad (8.51)$$

The ENOB can, however, be obtained via the digital data output of the converter, using the SNDR and linking it with the value of the ideal SNR, by expression:

$$\text{ENOB} = \frac{\text{SNDR}_{\text{measured}} - 1.763}{6.02} \quad (8.52)$$

We must, however, take into consideration that the ideal SNR is based on a uniform distribution of codes, which typically does not exist in reality. In an ideal converter, the noise is only sourced by quantization errors. In real converters, however, the noise sources are diverse, such as the nonlinearity errors, jitter errors,

and missing codes. The noise of reference voltage  $V_{ref}$  originated in the power supply also affects the value of ENOB. The value of  $ENOB_{FS}$  related to the fundamental frequency can be determined by one of the following expressions:

$$ENOB_{FS} = \left( \frac{SNDR - 10 \cdot \log(\text{Power of fundamental frequency}) - 1.76}{6.02} \right) \tag{8.53}$$

$$ENOB_{FS} = ENOB - \frac{\text{Amplitude of fundamental frequency}}{6.02} \tag{8.54}$$

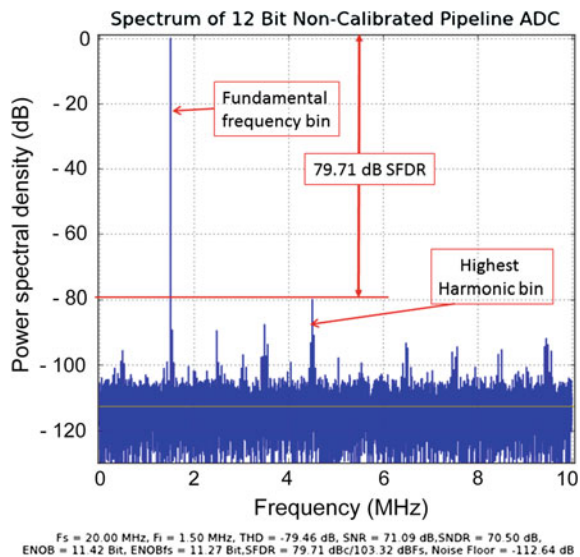
### 8.4.6 Spurs-Free Dynamic Range (SFDR)

Spurs-free dynamic range (SFDR) refers to the range of power output signal in the spectrum of FFT, distortion-free, from the power of the fundamental frequency to the power of the bin with highest power in the remaining bandwidth of the spectrum. The SFDR indicates the usable dynamic range of the converter free of distortion (Fig. 8.17).

The value of SFDR in  $dB_c$ 's related to the fundamental frequency is then given by the expression [20]:

$$SFDR_{dB_c} = 10 \cdot \log \left( \frac{\text{Power of fundamental frequency}}{\text{Power of the highest distortion bin}} \right) \tag{8.55}$$

Fig. 8.17 Determination of SFDR in the FFT [18]



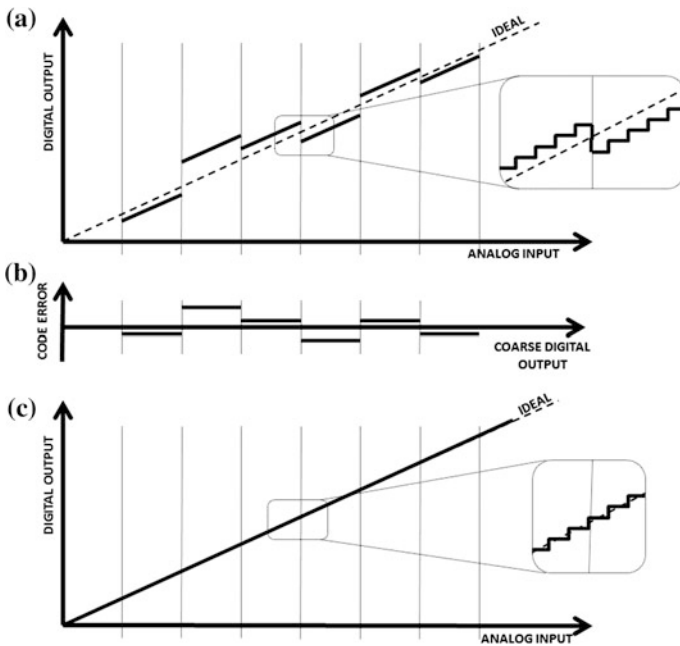
The value of SFDR in  $\text{dB}_{\text{FS}}$ 's, related to the Full Scale range, is given by:

$$\text{SFDR}_{\text{dB}_{\text{FS}}} = 10 \cdot \log\left(\frac{1}{\text{Power of the highest distortion bin}}\right) \quad (8.56)$$

### 8.5 Calibration Process

The method used for calibration in the tool is performed offline (foreground digital calibration) and is based on the method described by Song and Gustavsson [18, 21], which requires a forced offset voltage. Whatever the method used to generate the forced offset voltage, it must always keep output voltage below saturation.

Looking to a non-ideal transfer function of a pipeline ADC, generated with a ramp input signal [21], we can see that each segment deviates a given distance from an ideal straight line (Fig. 8.18a). The digital values different from the ideal line can be defined as code errors (Fig. 8.18b). That is, each segment can be realigned in a straight line by digitally subtracting the offset of each digital output that occurs in the segment range (Fig. 8.18c). The offsets are measured digitally, with the remaining pipeline stages, stored in memory, and later used on the pipeline output determination. Calibration errors are obtained by calculating the difference between



**Fig. 8.18** a Transfer function of a non-ideal pipeline ADC; b digital code errors; c transfer function after calibration [21]

the digital value of the residue corresponding to a given code, and the value of the expected residue for the following code.

The calibration algorithm starts by placing the auxiliary capacitor  $C_{aux}$  in parallel with the sampling capacitors array of MDAC. This auxiliary capacitor keeps the residue voltage of the MADC below saturation during the calibration of each capacitor. Its value should be smaller than  $C_s$  to minimize effect on amplifier feedback factor. During normal conversion,  $C_{aux}$  is always inhibited connecting bottom plate to GND.

### 8.5.1 Calibration Process Description

The first stage to be calibrated is the one with highest order in the range of initial stages defined for calibration, so that its errors are taken into account in the calibration of the earlier stages. Calculated errors are cumulative and affect converter linearity [21].

In the first step (Fig. 8.19) of the calibration process, a positive and negative reference voltage is applied to the  $C_{aux}$ . Those two resultant residue voltages are converted and measured by the remaining stages of the pipeline, obtaining first digital codes needed to determine the calibration errors, which are stored on a temporary memory. During this process, all remaining sampling capacitors are connected to GND. In the second step (Fig. 8.20),  $C_{aux}$  maintains its bottom plate linked either to  $+V_{ref}$  or  $-V_{ref}$ , and one of the sampling capacitors is activated connecting their bottom plate to a symmetric voltage of the one connected to  $C_{aux}$ ,

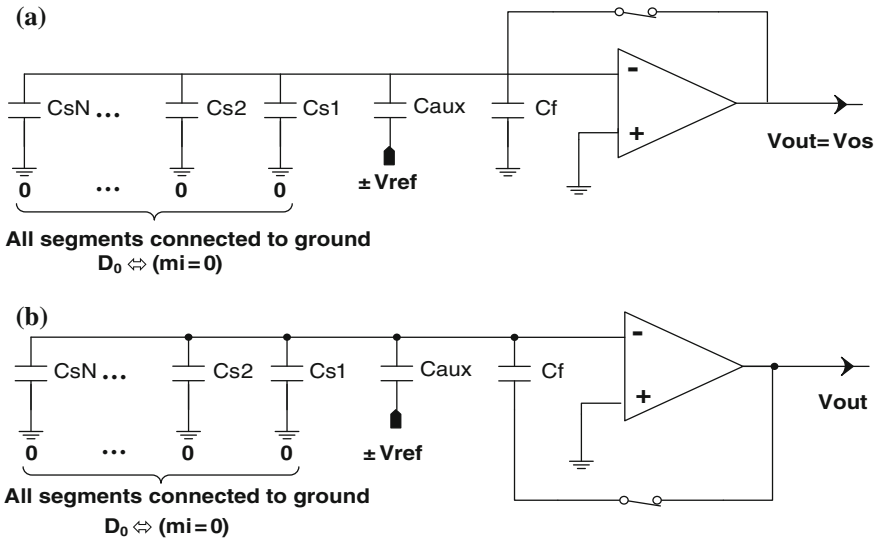


Fig. 8.19 Residue without active segments: **a** sampling phase; **b** amplification phase

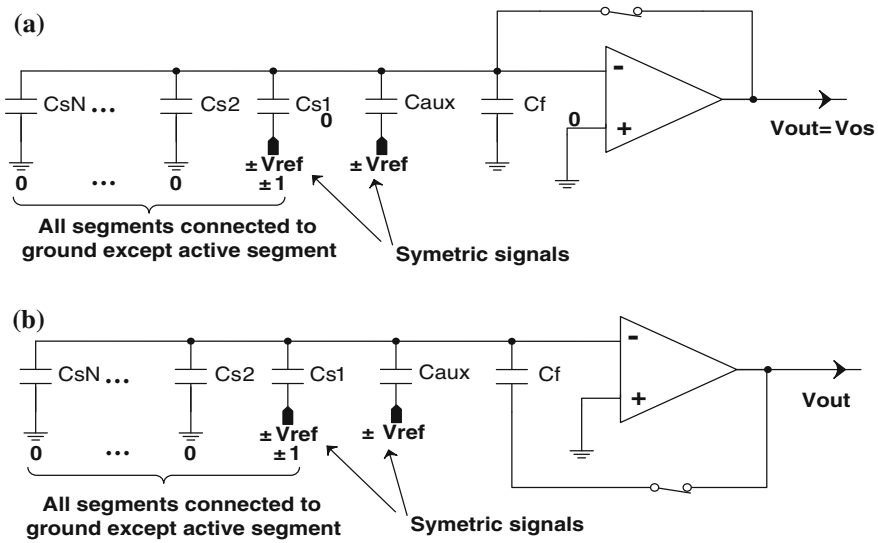


Fig. 8.20 Residue with one active segment: **a** sampling phase; **b** amplification phase

and a new measurement is carried out using the remaining stages of the pipeline. Once a digital code is generated, it is memorized and the calibration error code for that segment is calculated. The difference between the two measured digital codes should have ideally half the output full range. The difference to half the full range corresponds to an error that is stored for calibration. This second step is repeated for all sampling capacitors  $C_s$  errors are stored.

To obtain more accurate results, several cycles of the calibration process are carried out. Figure 8.21 represents a simplified sequence of correction and calibration of the pipeline output code. Initially, the code generated in the pipeline is corrected using digital correction RSD algorithm.

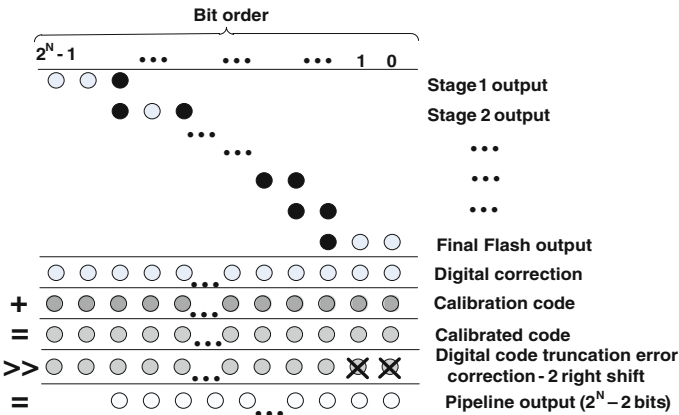


Fig. 8.21 Pipeline digital output code correction and calibration



## 8.6 SCALES—Graphic User Interface (GUI)

### 8.6.1 General Description

The simulation environment SCALES simulates Pipeline ADC’s, and main GUI front-page is represented in Fig. 8.22. The individual stages can be defined with selectable resolutions of 1.5, 2.5, 3.5, 4.5, and 5.5 bits and the final Flash can have 2, 3, 4, or 5 bits. Simulated pipeline can have up to 16 stages plus final Flash. The simulated pipeline can include digital foreground calibration. The tool also provides a statistical analysis of simulated topology to obtain an accurate estimation of the converter yield. Monte Carlo simulations can be performed to analyze the impact of the non-idealities generated by the converter design parameters. The tool was developed using the python language to allow platform independence. Figure 8.23 presents a simplified scheme of tool simulation engine.

### 8.6.2 Simulator Features

The tool provides a wide number of useful plots and characteristics about the simulated converter. Available plots are I/O relation, FFT, INL, DNL, and yield results. Yield results plots and their statistical evaluation are available for INL,

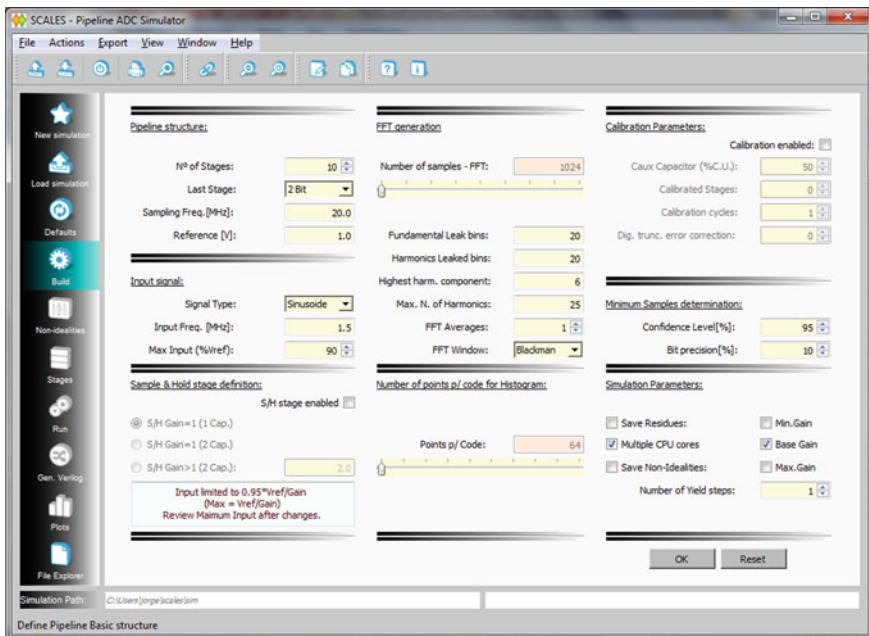


Fig. 8.22 SCALES simulator graphic user interface (GUI) [17]

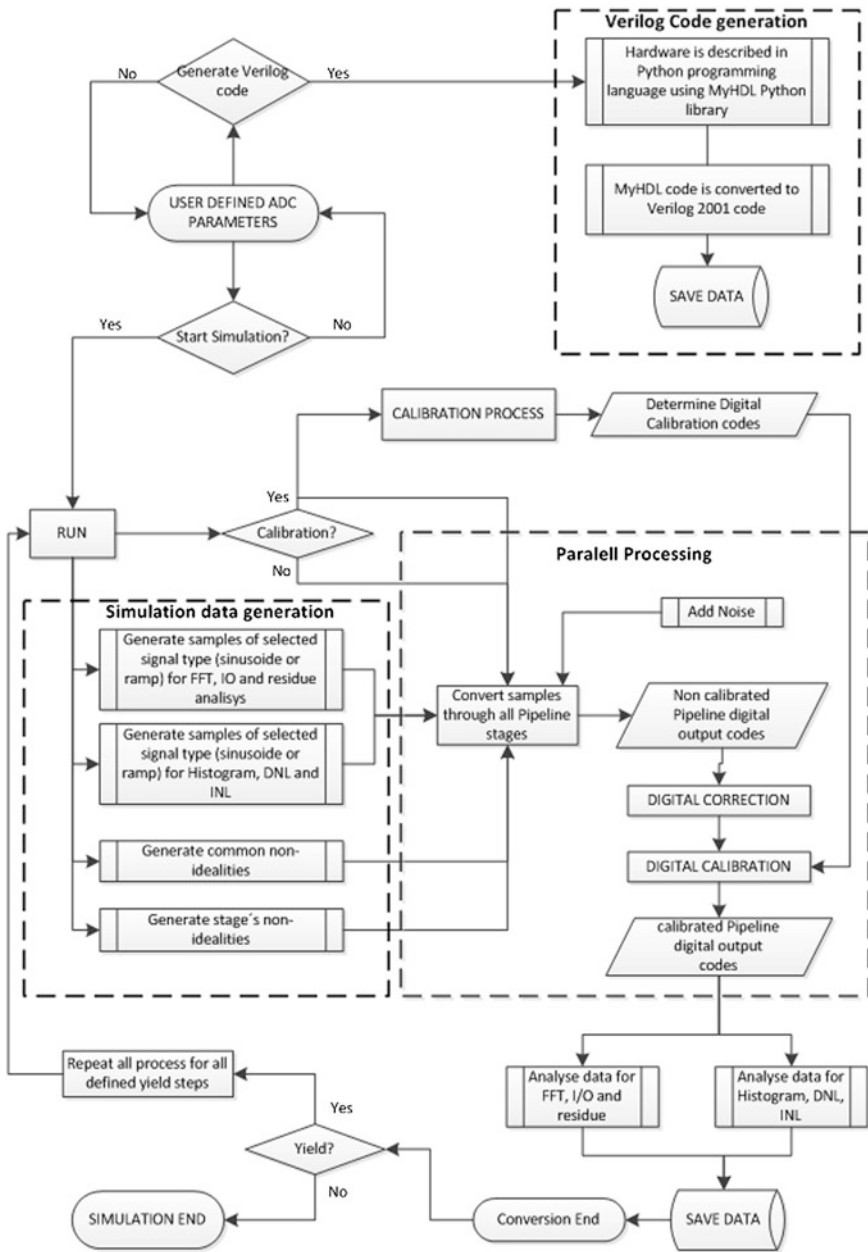


Fig. 8.23 SCALES simulation engine

DNL SNR, SNDR, THD, ENOB, ENOBfs, SFDR (dB<sub>c</sub>), and SFDR (dB<sub>fs</sub>). Sinusoidal or ramp input signals can be generated.

Before start simulation, the user has to define important parameters related to the converter design, such as non-idealities that affects linearity. Simulation parameters can be saved to a configuration file in .csv format and loaded later if needed to repeat the same simulation. GBW, finite amplifier gain, and slew rate can be defined with a base value and worst-case values (minimum and maximum). Also available are the results of the last simulation processed in every CPU. (Ex: Computer with 8 CPU's allows visualization of the last 8 yield steps processed one per CPU, if multiprocessing mode is active). Table 8.3 presents features comparison between SCALES and other ADC simulators.

**Table 8.3** Comparison to other simulators

Features	SCALES	[26]	[27]	[28]
Amplifier finite open-loop gain	Y	Y	Y	Y
Thermal noise	Y	Y	Y	Y
Capacitor mismatch	Y	Y	Y	Y
Capacitor nonlinearity	Y			
Amplifier parasitic capacitance	Y	Y	Y	Y
Amplifier slewing and linear settling	Y	Y	Y	Y
Amplifier offset	Y	Y	Y	Y
Comparators offset	Y	Y	Y	Y
Sampling clock jitter	Y	Y	Y	Y
Input signal type and frequency selection	Y	P	Y	P
Active non-idealities selection	Y		Y	
Worst-case analysis and comparison	Y			
Multiprocessing features	Y			
Yield statistical analysis and plots	Y			
INL/DNL plots	Y	Y	Y	Y
I/O plots	Y	Y	Y	
Stage residue plots	Y			
FFT analysis and plots	Y	Y	Y	Y
Confidence level, bit precision definition	Y			
Save/load parameters	Y	Y	Y	
Save/load complete simulation results	Y	P	P	
Digital self-calibration	Y	Y	Y	
Multiplatform (Unix, Windows, Mac)	Y			P
Advanced FFT parameters (windowing, no. of leakage bins, no. of averages, no. of samples...)	Y			
Multiple converter topologies				P
Reference voltage definition	Y			
High-resolution conversion speed	Y	N	N	N
Automatic digital calibration Verilog code	Y			

Y Fully implemented or presumably implemented; P Partially implemented; N No; Blank Not described, unknown state or not implemented

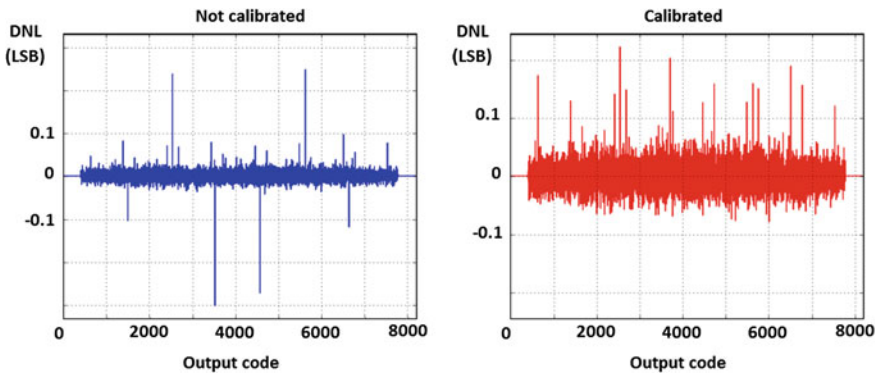
### 8.6.3 13 bits–40 MHz ADC Simulation

A 13 bits–40 MHz Pipeline ADC was simulated using 11 stages, with two initial stages of 2.5 bits and 8 stages with 1.5 bits, and the final flash has 2 bits resolution. Two additional bits were needed for digital correction of code, so we used an initial resolution of 15 bits to obtain a final resolution of 13 bits. Table 8.4 presents the overall results of the simulation showing a significant improvement with the use of the calibration method (Figs. 8.24 and 8.25).

A simulation of another pipeline ADC configuration (12 bit–20 MHz calibrated) with 1500 yield steps, 16.777.216 samples for INL and DNL analysis, using an average of  $4 \times 4096$  samples for FFT and residue analysis, took about 42 h for a sinusoidal input on a computer with 8 CPU’s.

**Table 8.4** Simulation results of a 13 bits, 40 MHz pipeline ADC

	Non calibrated	Calibrated
DNLmax	-0.601631	0.221821
INLmax	-0.826248	0.310333
THD	-81.39 dB	-86.20 dB
SNR	77.14 dB	78.53 dB
SNDR	75.76 dB	77.85 dB
ENOB	12.29 bits	12.64 bits
SFDR	83.97 dB <sub>c</sub>	89.98 dB <sub>c</sub>



**Fig. 8.24** DNL plot of a 13 bits–40 MHz pipeline ADC

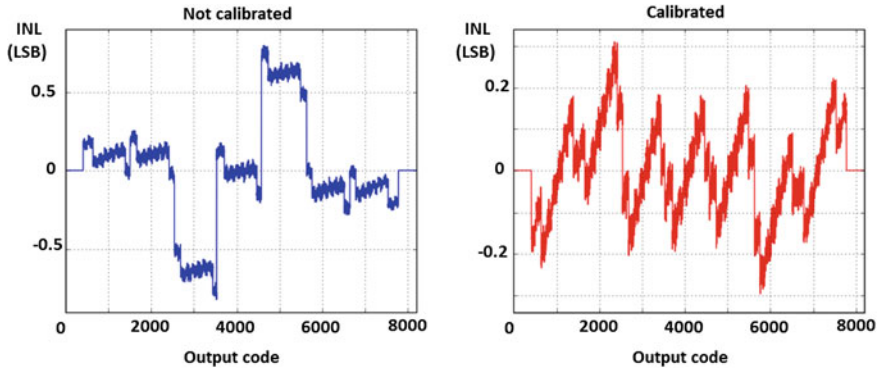


Fig. 8.25 INL plot of a 13 bits-40 MHz pipeline ADC

## 8.7 Conclusions

This work presented a new simulator tool for pipelined ADCs. The main functionality and features were described, including some advanced features not present in other known simulators. Some of those advanced features improve considerably the simulator capabilities once it is possible, with SCALES, to define the value of the auxiliary capacitor used for calibration, and the number of truncature bits applied to the digital output of the Pipeline. Some parameters related to FFT (highest harmonic frequency, fundamental frequency and harmonic leaked bins, and FFT windowing) can improve FFT results. For minimum samples determination, user can also define the confidence level and bit precision. Worst-case analysis can be performed simultaneously (GBW and slew rate referred to minimum base and maximum gain values), with independent plotting options for comparison.

The tool was developed in Python to allow platform independency, so it can run in Windows, Mac, or Linux systems. The multiprocessing feature takes advantage of actual multicore processors. Speed improvement is almost directly proportional to the number of CPUs, with each CPU working simultaneously processing a yield step. The tool allows a very fast simulation of a given converter topology and good modeling of stage nonlinearity.

A standard foreground calibration algorithm was described, which shows significant improvement in the design and performance of the converters simulation. At last, two other useful features are the capability to load and plot results of any simulation previously processed, even in another computer, and the possibility to save and load simulation parameters.

## References

1. Vital, J., Franca, J.: Synthesis of high-speed A/D converter architectures with flexible functional simulation capabilities. In: Proceedings of IEEE International Symposium on Circuits and Systems, pp. 2156–2159 (1992)
2. Malcovati, P. et al.: Behavioral modeling of switched-capacitor sigma-delta modulators. In: IEEE Transactions on Circuits and Systems I, Fundamental Theory and Applications, vol. 50, pp. 352–364 (2003)
3. Zare-Hoseini, H., Kale, I., Shoaee, O.: Modeling of switched-capacitor delta-sigma modulators in SIMULINK. IEEE Trans. Instrum. Meas. **54**, 1646–1654 (2005)
4. Hamoui, A.A., Alhajj, T., Taherzadeh-Sani, M.: Behavioral modeling of opamp gain and dynamic effects for power optimization of delta-sigma modulators and pipelined ADCs. In: Proceedings of International Symposium on Low Power Electronics and Design (ISLPED), pp. 330–333 (2006)
5. Medeiro, F., Pérez-Verdú, B., Rodríguez-Vázquez, A., Huertas, J.L.: A vertically integrated tool for automated design of sigma delta modulators. IEEE J. Solid-State Circuits **30**, 762–772 (1995)
6. Goes, J., Vital, J.C., Franca, J.E.: Systematic design for optimization of high-speed self-calibrated pipelined A/D converters. IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process **45**, 1513–1526 (1998)
7. Horta, N., Franca, J.: Algorithm-Driven Synthesis of Data Conversion Architectures. IEEE Trans. Comput. Aided Des **16**(10), 1116–1135 (1997)
8. Horta, N., Franca, J., Leme, C.: Framework for architecture synthesis of data conversion systems employing binary-weighted capacitor arrays. In: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1789–1782 (1991)
9. Guilherme, J., Horta, N., Franca, J.: Symbolic synthesis of non-linear data converters. In: Proceedings of IEEE International Conference on Electronics Circuits and Systems (ICECS), vol. 3, pp. 219–222 (1998)
10. Vital, J., Horta, N., Silva, N., Franca, J.: CATALYST: a highly flexible CAD tool for architecture-level design and analysis of data converters. In: Proceedings of Joint Conference European Design Automation Conference and European Application Specific Integrated Circuit (EDAC-EUROASIC), pp. 472–477 (1993)
11. Horta, N., Fino, M., Goes, J.: Symbolic techniques applied to switched-current ADCs synthesis. In: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), Geneva, Switzerland, pp. 129–132 (2000)
12. Ruiz-Amaya, J. et al.: High-level synthesis of switched-capacitor, switched-current and continuous-time sigma-delta modulators using Simulink based time-domain behavioral models. IEEE Trans. Circuit Syst. I, Reg. Papers, vol. 52, pp. 1795–1810 (2005)
13. Bilhan, E. et al.: Behavioral model of pipeline ADC by using SIMULINK. In: Proceedings of Southwest Symposium Mixed-Signal Design, pp. 147–151 (2001)
14. Phelps, R. et al. Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **19**, 703–717 (2000)
15. Ochotta, E., Carley, R.L.: Synthesis of high-performance analog circuits in ASTRX/OBLX. IEEE Trans Comput. Aided Des. Integr. Circuits Syst. **15**(3), 273–294 (1996)
16. Kwok, P.T.M., Luong, H.C.: Power optimization for pipeline analog-to-digital converters. IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. **46**, 549–553 (1999)
17. Silva, C., Ayzac, P., Guilherme, J., Horta, N.: SCALES—A behavioral simulator for pipelined analog-to-digital converter design. In: International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design—SMAC, pp. 149–152 (2012)
18. Lee, S.H., Song, B.S.: Digital-domain calibration of multistep analog-to-digital converters. IEEE JSSC **27**(12), 1679–1688 (1992)

19. James, C.: Clocking high-speed A/D converters. National Semiconductors, App. Note 1558 (2007)
20. Andrea, B.: Analog-to-digital data converters. Università degli Studi di Lecce. Lecture presentation, pp. 1–322 (2007)
21. Mikael, G., Jacob, W., Nianxiong, T.: CMOS Data Converters for Communications, pp. 229–256. Kluwer Academic Publishers, NY (2002) (ISBN 0-306-47305-4)
22. Cline, David, Gray, Paul: A power optimized 13-bit 5 Msamples/s pipelined analog to digital converter in 1.2  $\mu\text{m}$  CMOS. IEEE J. Solid State Circuits **30**(4), 443–452 (1995)
23. Zanchi, A., Tsay, F., Papantonopoulos, I.: Impact of capacitor dielectric relaxation on a 14-bit 70MS/s pipeline ADC in 3 V BiCMOS. IEEE J. Solid-State Circuits **38**(12), 2077–2086 (2003)
24. Doernberg, J., Lee, H.S., Hodges, D.A.: Full-speed testing of A/D converters. IEEE J. Solid-State Circuits, **SC-19**(6), 820–827 (1984)
25. Schiff, M.: Spectrum analysis using digital FFT techniques. Agilent Technologies. AN106A (1997)
26. Sahoo, B.D., Razavi, B.: A fast simulator for pipelined A/D converters. In: IEEE Circuits Syst. MWSCAS '09. (2), 402–406 (2009)
27. Navin, V., Hassoun, M., Ray, T., Marwan; Black, W., Lee, E., Soenen, E., Geiger, R.: A simulation environment for pipelined analog-to-digital converters. In: IEEE International Symposium on Circuits and Systems, pp. 1620–1623, 9–12 June 1997
28. Zareba, G., Palusinski, O.: Behavioral simulator of analog-to-digital converters for telecommunication applications. In: Behavioral Modeling and Simulation Conference BMAS 2004, University of Arizona, pp. 1–6 (2004)

**Part II**  
**Radio-Frequency Design**



# Chapter 9

## SMAS: A Generalized and Efficient Framework for Computationally Expensive Electronic Design Optimization Problems

**Bo Liu, Francisco V. Fernández, Georges Gielen, Ammar Karkar, Alex Yakovlev and Vic Grout**

**Abstract** Many electronic design automation (EDA) problems encounter computationally expensive simulations, making simulation-based optimization impractical for many popular synthesis methods. Not only are they computationally expensive, but some EDA problems also have dozens of design variables, tight constraints, and discrete landscapes. Few available computational intelligence (CI) methods can solve them effectively and efficiently. This chapter introduces a surrogate model-aware evolutionary search (SMAS) framework, which is able to use much fewer expensive exact evaluations with comparable or better solution quality. SMAS-based methods for mm-wave integrated circuit synthesis and network-on-chip parameter design optimization are proposed and are tested on several practical problems. Experimental results show that the developed EDA methods can obtain highly optimized designs within practical time limitations.

---

B. Liu (✉) · V. Grout  
Department of Computing, Glyndwr University, Wrexham, UK  
e-mail: b.liu@glyndwr.ac.uk; Bo.Liu@esat.kuleuven.be

V. Grout  
e-mail: v.grout@glyndwr.ac.uk

F.V. Fernández  
IMSE, CSIC and Universidad de Sevilla, Seville, Spain  
e-mail: Francisco.Fernandez@imse-cnms.csic.es

G. Gielen  
ESAT-MICAS, Katholieke Universiteit Leuven, Leuven, Belgium  
e-mail: Georges.Gielen@esat.kuleuven.be

A. Karkar · A. Yakovlev  
School of Electrical Electronic and Computer Engineering, Newcastle University,  
Newcastle, UK  
e-mail: a.j.m.karkar@newcastle.ac.uk

A. Yakovlev  
e-mail: Alex.Yakovlev@newcastle.ac.uk

## 9.1 Introduction

Today, bioinspired innovative design techniques are becoming increasingly important for electronic design automation (EDA). Among various available methods, evolutionary algorithms (EAs) are being widely applied to optimize integrated circuits (ICs) and systems [1, 2]. However, a new challenge appears: The evaluation of fitness functions (i.e., simulation) for many electronic design optimization problems is computationally expensive. Typical examples are the simulation of high-frequency ICs, antennas, networks-on-chip (NoCs), photonic devices, and microelectromechanical systems (MEMSs). A single simulation may take from dozens of minutes to several hours. Standard EAs often need hundreds to thousands of such simulations to obtain optimal design solutions, which may cost an impractical optimization time.

The long simulation time of electronic devices, ICs, and systems is mainly due to the following: (1) solving complex partial differential equations by numerical methods (e.g., high-frequency IC and antenna synthesis) and (2) using Monte Carlo (MC) sampling methods in simulation (e.g., process variation-aware analog IC sizing/yield optimization and NoC parameter design optimization). Thus, two possible solutions are: (1) improving the computational overhead for each simulation and (2) designing new optimization methodologies using fewer simulations in the optimization process. Note that the above two approaches are compatible. A large amount of research has been carried out for the former approach [3], and the latter is an emerging area in recent years and is the focus of this chapter.

This chapter will discuss surrogate model-assisted evolutionary algorithms (SAEAs) for decreasing the number of necessary simulations in electronic design optimization. SAEA is an emerging approach in the computational intelligence (CI) field. SAEA employs (a) surrogate model(s) to replace computationally expensive exact function evaluations (i.e., simulations). Surrogate models are approximation models of the simulation but are computationally much cheaper, and the additional computational overhead of surrogate modeling is often not large. Due to this, the computational cost can be reduced significantly. Note that SAEA is different from off-line surrogate model-based optimization methods. In off-line surrogate model-based optimization methods, a good surrogate model for the whole design space is firstly constructed and then used to replace the simulations. The majority of training data points is often obtained based on a one-shot sampling, although there may be minor updates to the surrogate model in the iterative optimization process. When the dimensionality is higher (e.g., larger than 10), generating sufficient samples to build a good surrogate model itself is very or prohibitively time-consuming. In addition, many samples are not very useful for optimization since they are far from optimal areas [4]. SAEA, on the other hand, carries out online (or active) learning and optimization, in which the sampling, surrogate modeling, and evolutionary search are working simultaneously and the surrogate modeling mainly targets the subregions visited by evolutionary search operators.

The basic idea to design SAEA frameworks for EDA problems and practical methods will be discussed in this chapter. The remainder of this chapter is organized as follows. Section 9.2 briefly reviews SAEA research in the CI field and discusses the requirements of SAEAs for EDA problems. Section 9.3 introduces the basic CI techniques. The surrogate model-aware evolutionary search (SMAS) framework is presented in Sect. 9.4. Practical SAEA methods for mm-wave IC synthesis and NoC parameter design optimization are introduced in Sects. 9.5 and 9.6, respectively. Concluding remarks are presented in Sect. 9.7.

## 9.2 State of the Art and Challenges of SAEA for EDA Problems

Generally, a good solution method for EDA problems should have the following three properties: (1) good global optimization ability, (2) high efficiency, and (3) scalable to medium-dimensional problems (e.g., 20–50 variables). Some SAEA methods and off-line model-based optimization methods are available for small-scale (e.g., less than 10 variables) expensive EDA problems, obtaining good solutions with a reduced number of expensive simulations [5]. However, the third requirement, scalability to medium-dimensional problems, is still an open question even in today's SAEA research.

For a surrogate-based optimization method, an unavoidable problem is to appropriately handle the prediction uncertainty of the surrogate model. Early methods did not consider the model uncertainty in the optimization process. With a number of samples with exact function evaluations serving as the training data, a surrogate model is constructed and the optimal candidate solutions based on surrogate model prediction are evaluated by exact function evaluators. The surrogate model is then updated, and the above step is repeated until convergence. To address the issue of incorrect convergence of the above method, a straightforward solution is a generation-based control framework [6]. In some generations, exact evaluations are used, while in other generations, surrogate model predictions are used. The frequency of using prediction increases when the prediction uncertainty decreases.

In addition to a direct use of prediction, prescreening has been introduced into SAEAs (e.g., expected improvement, probability of improvement) [7]. Instead of directly replacing the exact function evaluation by the surrogate model prediction (the model uncertainty should be as small as possible to this end), prescreening methods aim to select the possible promising candidates from the newly generated candidate solutions utilizing the prediction uncertainty. Because both the EA and the prescreening methods contribute to the global search, methods based on this framework can often detect the globally optimal or near-optimal solutions efficiently for small-scale problems. Successful prescreening-based SAEA examples are reported in [4, 8]. These methods can often obtain high-quality solutions with a relatively small number of function evaluations.

Another popular SAEA framework is the trust-region local search (TLS) [9, 10]. Methods based on this framework use local surrogate models and memetic algorithms. In particular, after using the evolutionary operators to locate the new candidate solutions in a global manner, a local search phase is included to refine these candidate solutions based on cheap local surrogate models. Exact function evaluations are then performed on the selected promising candidates, and the surrogate models are updated.

However, higher dimensionality is challenging for the above SAEA frameworks. A good-quality surrogate model from which predictions can be made without too much uncertainty is essential. The two main factors affecting the quality of the surrogate model are the number of training data points and their locations. Without considering the locations, it is intuitive that more training data points are needed for medium- and high-dimensional problems to construct a reasonably good surrogate model: The higher the dimensionality, the more the training data points are necessary. Nevertheless, the number of exact evaluations for generating the training data points is limited by the practical optimization time. Some state-of-the-art methods using the above frameworks [8–10] were tested using typical 20- and 30-dimensional mathematical benchmark problems. Experimental results show that either they require several thousand exact evaluations to get reasonably good solutions or the obtained solutions still need much improvement with fewer exact evaluations. Note that for many EDA problems, the number of design variables is around 15–30; therefore, new methods enabling efficient global optimization for medium-scale problems are needed. Besides that, other challenges exist for practical EDA problems. For example, tight constraints need to be handled in mm-wave IC synthesis problems, and integer optimization needs to be addressed in NoC design optimization problems.

### 9.3 Basic Techniques

Surrogate modeling methods and EAs are two essential components of an SAEA, and there are various methods available for each. A review of some popular surrogate modeling methods can be found in [11]. An introduction to EAs can be found in [12]. In this chapter, we will introduce Gaussian process (GP) machine learning and differential evolution (DE) optimization, which will be used in the SAEAs presented in Sects. 9.4–9.6.

#### 9.3.1 GP Modeling

To model an unknown function  $y = f(x)$ ,  $x \in R^d$ , the GP modeling assumes that  $f(x)$  at any point  $x$  is a Gaussian random variable  $N(\mu, \sigma^2)$ , where  $\mu$  and  $\sigma$  are two constants independent of  $x$ . For any  $x$ ,  $f(x)$  is a sample of  $\mu + \varepsilon(x)$ , where

$\varepsilon(x) \sim N(0, \sigma^2)$ . For any  $x, x' \in \mathbb{R}^d, c(x, x')$ , the correlation between  $\varepsilon(x)$  and  $\varepsilon(x')$  depends on  $x - x'$ . More precisely,

$$c(x, x') = \exp\left(-\sum_{i=1}^d \theta_i |x_i - x'_i|^{p_i}\right), \tag{9.1}$$

where parameter  $1 \leq p_i \leq 2$  is related to the smoothness of  $f(x)$  with respect to  $x_i$ , and parameter  $\theta_i > 0$  indicates the importance of  $x_i$  on  $f(x)$ . More details about GP modeling can be found in [13].

### 9.3.1.1 Hyper-Parameter Estimation

Given  $K$  points  $x^1, \dots, x^K \in \mathbb{R}^d$  and their  $f$ -function values  $y^1, \dots, y^K$ , then the hyper-parameters  $\mu, \sigma, \theta_1, \dots, \theta_d$ , and  $p_1, \dots, p_d$  can be estimated by maximizing the likelihood that  $f(x) = y^i$  at  $x = x^i$  ( $i = 1, \dots, K$ ) [7]:

$$\frac{1}{(2\pi\sigma^2)^{K/2} \sqrt{\det(C)}} \exp\left[-\frac{(y - \mu\mathbf{1})^T C^{-1} (y - \mu\mathbf{1})}{2\sigma^2}\right] \tag{9.2}$$

where  $C$  is a  $K \times K$  matrix whose  $(i, j)$  element is  $c(x^i, x^j)$ ,  $y = (y^1, \dots, y^K)^T$ , and  $\mathbf{1}$  is a  $K$ -dimensional column vector of ones.

To maximize (9.2), the values of  $\mu$  and  $\sigma^2$  must be:

$$\hat{\mu} = \frac{\mathbf{1}^T C^{-1} y}{\mathbf{1}^T C^{-1} \mathbf{1}} \tag{9.3}$$

and

$$\hat{\sigma}^2 = \frac{(y - \mathbf{1}\hat{\mu})^T C^{-1} (y - \mathbf{1}\hat{\mu})}{K}. \tag{9.4}$$

Substituting (9.3) and (9.4) into (9.2) eliminates the unknown parameters  $\mu$  and  $\sigma$  from (9.2). As a result, the likelihood function depends only on  $\theta_i$  and  $p_i$  for  $i = 1, \dots, d$ . Equation (9.2) can then be maximized to obtain estimates of  $\hat{\theta}_i$  and  $\hat{p}_i$ . The estimates  $\hat{\mu}$  and  $\hat{\sigma}^2$  can then readily be obtained from (9.3) and (9.4).

### 9.3.1.2 The Best Linear Unbiased Prediction and Predictive Distribution

Given the hyper-parameter estimates  $\hat{\theta}_i, \hat{p}_i, \hat{\mu}$ , and  $\hat{\sigma}^2$ , one can predict  $y = f(x)$  at any untested point  $x$  based on the  $f$ -function values  $y^i$  at  $x^i$  for  $i = 1, \dots, K$ . The best linear unbiased predictor of  $f(x)$  is [7]:

$$\hat{f}(x) = \hat{\mu} + r^T C^{-1}(y - \mathbf{1}\hat{\mu}) \quad (9.5)$$

and its mean-squared error is:

$$s^2(x) = \hat{\sigma}^2 \left[ 1 - r^T C^{-1} r + \frac{(1 - \mathbf{1}^T C^{-1} r)^2}{\mathbf{1}^T C^{-1} r} \right] \quad (9.6)$$

where  $r = (c(x, x^1), \dots, c(x, x^K))^T$ .  $N(\hat{f}(x), s^2(x))$  can be regarded as a predictive distribution for  $f(x)$  given the function values  $y^i$  at  $x^i$  for  $i = 1, \dots, K$ .

### 9.3.1.3 Lower Confidence Bound (LCB)

We consider minimization of  $f(x)$  in this chapter. Given the predictive distribution  $N(\hat{f}(x), s^2(x))$  for  $f(x)$ , the LCB of  $f(x)$  can be defined as [14]:

$$f_{\text{lc}}(x) = \hat{f}(x) - \omega s(x) \quad (9.7)$$

where  $\omega$  is a constant. In the following SAEAs,  $f_{\text{lc}}(x)$  is used instead of  $\hat{f}(x)$  itself to measure the quality of  $x$ . The use of LCB can balance the search between promising areas (i.e., with low  $\hat{f}(x)$  values) and less explored areas (i.e., with high  $s(x)$  values).

The surrogate modeling method introduced above is called ordinary GP. There are several variants of GP. More details can be found in [15].

## 9.3.2 DE Optimization

DE is an effective and popular global optimization algorithm. It uses a differential operator to create new candidate solutions [16]. There are quite a few different DE variants. In this chapter, we will introduce the DE/best/1 and DE/current-to-best/1 mutation strategies to generate new solutions.

Suppose that  $P$  is a population and the best individual in  $P$  is  $x^{\text{best}}$ . Let  $x = (x_1, \dots, x_d) \in R^d$  be an individual solution in  $P$ . To generate a child solution  $u = (u_1, \dots, u_d)$  for  $x$ , a mutant vector is first produced by mutation:

1. DE/best/1

$$v_i = x^{\text{best}} + F \cdot (x^{r_1} - x^{r_2}) \quad (9.8)$$

where  $x^{\text{best}}$  is the best individual in  $P$  and  $x^{r_1}$  and  $x^{r_2}$  are two different solutions randomly selected from  $P$  and also different to  $x^{\text{best}}$ .  $v_i$  is the  $i$ th mutant vector in the population after mutation.  $F \in (0, 2]$  is a control parameter, often called the scaling factor [16].

## 2. DE/current-to-best/1<sup>1</sup>

$$v_i = x^i + F \cdot (x^{\text{best}} - x^i) + F \cdot (x^{r1} - x^{r2}) \quad (9.9)$$

where  $x^i$  is the  $i$ th vector in the current population.

After mutation, a crossover operator is applied to produce the child  $u$ . A widely used crossover method is as follows:

1. Randomly select a variable index  $j_{\text{rand}} \in \{1, \dots, d\}$ ,
2. For each  $j = 1$  to  $d$ , generate a uniformly distributed random number  $rand$  from  $(0, 1)$  and set:

$$u_j = \begin{cases} v_j, & \text{if } (rand \leq CR) \mid j = j_{\text{rand}} \\ x_j, & \text{otherwise} \end{cases} \quad (9.10)$$

where  $CR \in [0, 1]$  is a constant called the crossover rate.

The DE algorithm is shown to be very powerful for real parameter optimization problems. For integer parameters (e.g., the number of fingers of transistors), a quantization method needs to be used [16]. In a search, floating point values are still used to handle discrete variables in evolutionary operators and are quantized to their nearest allowed values only in function evaluation.

## 9.4 The Surrogate Model-Aware Evolutionary Search Framework

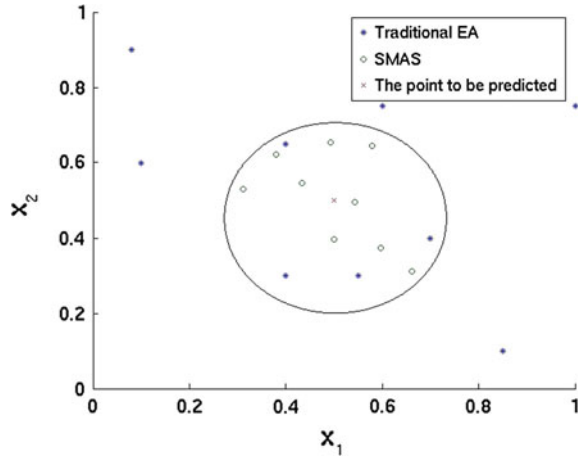
### 9.4.1 Key Ideas

Section 9.2 has reviewed the challenges of SAEAs for EDA problems. The goal of SMAS is to address expensive optimization problems with around 20–50 variables. The key idea of the SMAS framework is to replace the standard EA by a new search mechanism considering both global optimization and high-quality surrogate modeling in such dimensionality. As was described above, most state-of-the-art SAEAs are based on the standard EA structure. This introduces complex population updating, which requires surrogate models with good quality in many subregions to guarantee the correctness of replacements. Clearly, this is not good for surrogate modeling with limited training data points. It becomes paradoxical in SAEAs that the evaluated candidate solutions are determined by the EA according to the optimization goals, but these solutions may not be the most appropriate ones for the surrogate modeling. Figure 9.1 shows a typical spreading of the training data pool

---

<sup>1</sup>This mutation strategy is also referred to as DE/target-to-best/1.

**Fig. 9.1** An illustrative figure for the locations of the training data pool by different search mechanisms



in two of the  $d$  dimensions when using standard EA operators and population updating. The current promising subregion is shown by the ellipse, and a point waiting to be predicted or prescreened is shown by the cross. It can be seen that the already evaluated candidate solutions spread in different search subregions. When using the whole training data pool, the points far away from the point with the cross will, on the contrary, deteriorate the quality of the constructed surrogate model. Note that this is different from off-line surrogate modeling, whose training data points are intentionally located almost uniformly. On the other hand, there are not enough training data points in the current promising area to produce a high-quality surrogate model. This problem becomes more obvious when the training data points are limited compared with the large design space, which occurs for many EDA problems. To that end, a SMAS mechanism that unifies the optimization and surrogate modeling is needed.

Instead of using a standard EA population [8] or using a continuously increasing population [4] in SAEA, the  $\lambda$  current best candidate designs form the parent population (it is reasonable to assume that the search focuses on the promising subregion) and the best candidate among the generated  $\lambda$  child candidates (based on prescreening) is selected to replace the worst one in the parent population at each iteration.<sup>2</sup> In this way, at most one candidate is changed in the parent population at each iteration, so the best candidate in the child solutions in several consecutive iterations may be quite near (they will then be simulated and are used as training data points). Therefore, the training data describing the current promising region can be much denser compared to those generated by a standard EA population updating. The prediction quality can therefore be largely enhanced.

<sup>2</sup>Note that the selected candidate solution is not necessarily the actual best one in terms of exact simulation; it is satisfactory that the prescreened best one is among the top few best candidates in reality.



However, owing to this new search framework, the population diversity and exploration ability may be affected negatively, which are often not serious concerns for SAEA frameworks using standard EAs. On the other hand, there are also research works stating that standard EAs may have too much randomness or excessive diversity [17]. Investigations show that when using appropriate parameters and DE mutation strategies, the search ability of standard EA can almost be maintained in SMAS [18], which is verified by more than 10 mathematical benchmark problems and EDA problems.

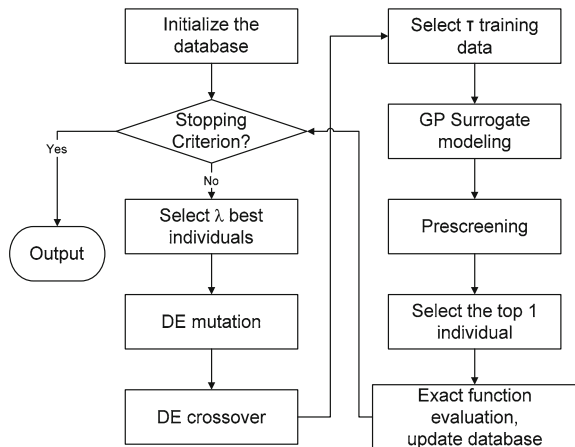
### 9.4.2 The SMAS Framework

SMAS records all the simulated solutions and their function values in a database. Once a simulation has been conducted for a new candidate design  $x$ , the  $x$  and its performances  $y$  will be added to the database. To initialize the database, a Design of Experiments method, Latin Hypercube Sampling (LHS) [19], is used to sample a set of initial points from the search space [20].

The flow diagram of the SMAS framework is shown in Fig. 9.2. It works as follows:

- Step 1: Use LHS to sample  $\alpha$  candidate designs from  $[a, b]^d$ . Perform simulations to all of these candidate designs and let them form the initial database.
- Step 2: If a preset stopping criterion is met (e.g., a threshold of synthesis time, a certain number of iterations), output the best design in the database; otherwise, go to Step 3.
- Step 3: Select from the database the  $\lambda$  best candidate designs in terms of simulation results to form a population  $P$ .
- Step 4: Apply the DE operators ((9.8)/(9.9) and (9.10)) on  $P$  to generate  $\lambda$  child solutions.

Fig. 9.2 The SMAS framework



- Step 5: Select training data to construct GP surrogate models.
- Step 6: Prescreen the  $\lambda$  child solutions generated in Step 4 by using the GP model with LCB prescreening for the objective function and the predicted values for each constraint.
- Step 7: Perform simulation to the estimated best child solution from Step 6. Add this evaluated design and its performances to the database. Go back to Step 2.

Using mathematical benchmark problems and real-world EDA problems, SMAS-based SAEAs show clear improvements on surrogate model quality and the necessary number of expensive exact evaluations or simulations compared to standard EA and several popular SAEAs. More details are in [21–23].

In this general framework, three components need careful investigation and are affected by the types of problems. They are:

1. Criterion to judge the “best design”: The best design is straightforward for single-objective unconstrained optimization, but for constrained optimization and optimization in uncertain environments, different criteria need to be developed.
2. The method to select training data points for surrogate modeling: There are several empirical methods for such selection. Reference [18] investigates them for problems with different kinds of landscapes.
3. Which DE mutation method to use: Again, this depends on different kinds of problems. It is intuitive that for problems with more complex landscapes, more diversity is needed and DE/current-to-best/1 may work better than DE/best/1. Reference [18] provides more details. For EDA problems, the recommended selection method is using DE/best/1 for problems with continuous design variables and with up to approximately 20 dimensions, while DE/current-to-best/1 is recommended for problems with discrete variables and/or with around 30–50 dimensions.

### 9.4.3 Parameter Settings

There are several control parameters in SMAS. Empirical rules based on mathematical benchmarks and real-world problem tests are provided as follows.

- The scaling factor  $F$  and the crossover rate  $CR$  in the DE operators: In standard DE,  $F$  is suggested to be set around 0.5 to balance the exploration and exploitation [16]. In SMAS, a large  $F$  is often necessary. The reason is that SMAS always uses the  $\lambda$  best solutions from the database as the parent population, which emphasizes exploitation. To maintain the exploration ability, a large  $F$  is needed.  $F \in [0.75, 0.95]$  often achieves good results. The crossover rate,  $CR$ , on the other hand, is problem specific. Good values of  $CR$  generally

fall into a small range for a given problem [24]. This implies that a self-adaptation mechanism for  $CR$  in the SMAS framework is useful. On the other hand, problems with rugged landscapes are often sensitive to  $CR$  values, which can seldom be seen for EDA problems to the best of our knowledge. Therefore,  $CR \in [0.7, 0.9]$  is suggested.

- $\omega$  used in LCB: Following the suggestions in [14],  $\omega = 2$  is used.
- The number of initial samples  $\alpha$ : Our empirical rule is that  $\alpha$  should be set to at least  $3 \times d$  or the robustness will decrease ( $d$  is the number of design variables). The parameter  $\alpha$  is affected by the complexity of the function. For highly multimodal problems,  $\alpha = 5 \times d$  is often enough.
- The population size  $\lambda$ : This is a DE parameter. Although SMAS has a completely different population updating method compared to standard DE, pilot experiments showed that the recommended setting of DE population size [16] is still applicable. Using  $30 \leq \lambda \leq 60$  often works well for EDA problems. A large  $\lambda$  value causes slow convergence, and a small value can lead to premature convergence.

## 9.5 GASPAD: An SMAS-Based SAEA for mm-Wave IC Synthesis

This section presents an SMAS-based mm-wave IC synthesis method, called general mm-wave IC synthesis based on Gaussian process model-assisted differential evolution (GASPAD).

### 9.5.1 A Review of RF IC Synthesis

In recent years, design and optimization methodologies for mm-wave ICs are attracting more and more attention. This trend will continue in the foreseeable future, since the demand for high-data-rate wireless communications is constantly increasing [25]. However, mm-wave IC design still depends highly on the designer's experience. The design procedure is often time-consuming and often gets suboptimal results. Two important reasons for this are:

- Equivalent circuit models of integrated passive components (e.g., inductor, transformer), which are critical in radio frequency (RF) ICs, are not valid for high frequencies. They are not sufficient for mm-wave circuit simulation where the distributed effects of the passive components have to be taken into account. As a result, the designers are forced to rely on experience, intuition, and time-consuming electromagnetic (EM) simulators to predict the circuit performance and revise the design parameters. The design procedure involves quite a number of iterations and is time-consuming even for experienced designers.

- The traditional mm-wave IC design method relies on a systematic step-by-step design procedure, but it is sometimes difficult to optimize the desired circuit performance. Consider, for example, the design of a power amplifier (PA). Most mm-wave PA designs optimize the saturated output power ( $P_{\text{sat}}$ ) and consequently the maximum power-added efficiency ( $PAE@P_{\text{sat}}$ ). However, optimizing the  $PAE$  at the 1-dB compression point ( $P_{1\text{dB}}$ ) is more important to have a high average efficiency when transmitting modulated signals (e.g., 16QAM) [26]. Nevertheless, it is not easy to find the optimal load impedance (by load-pull simulation) and the optimal bias point to optimize  $PAE@P_{1\text{dB}}$  by the traditional PA design method.

To address this challenge, CI-based synthesis methods have been investigated. In the literature, the high-frequency IC synthesis research can be divided into two subareas with different requirements on effectiveness and efficiency:

- Low-GHz RF IC synthesis  
Some successful research exists in this area [27–29], and the main focus is the effectiveness, or the optimality. Computationally cheap parasitic-aware models for passive components are generated and are used for simulation. Like many analog circuit sizing methods, EAs are used to obtain optimized design solutions. Mathematically, they solve a constrained optimization problem, assuming that the number of simulations is not a limitation. Several hundreds to thousands of simulations can be used for obtaining optimized solutions.
- mm-wave IC synthesis  
There are two typical kinds of problems in mm-wave IC synthesis: synthesis focusing on small-signal performance optimization and general mm-wave IC synthesis.  
The first synthesis method for mm-wave ICs working at 100 GHz or above, called efficient machine learning-based differential evolution (EMLDE), was proposed in [20]. Due to the  $f_T$  of many technologies (e.g., 65 nm CMOS), maximizing the power gain (small-signal performance) is often the main consideration for ICs working at 100 GHz or above. In this area, besides using an EA to achieve optimized design solutions, efficiency becomes the main challenge, since at such frequencies computationally expensive EM simulation is unavoidable. When directly embedding the EM simulation into the EAs, an impractically long optimization time will result [20]. An SAEA was introduced into mm-wave IC synthesis in EMLDE. To address the dozens of design variables in mm-wave IC synthesis, a decomposition method exploiting the properties of the targeted problem was proposed for dimension reduction in EMLDE. It is difficult to apply the decomposition method for dimension reduction used in EMLDE for general mm-wave IC synthesis. Indeed, EMLDE relies on the stage-by-stage design method for mm-wave amplifiers focusing on small-signal performance optimization. Maximizing the power gain ( $G_p$ ) can be considered separately for each stage. However, both large-signal and small-signal performances need to be considered for general mm-wave IC synthesis. For example, for a 60 GHz PA, the

$PAE$ ,  $P_{1dB}$ , and  $G_p$  all need to be maximized. A stage designed for gain maximization may not be a good design for efficiency maximization. When using the decomposition method from EMLDE, appropriate specifications of all performance metrics for each stage are a must, but this is not easy to specify even for well-experienced designers. In addition, because of the multiple (high-performance) specifications, good constraint-handling techniques are needed, instead of the static penalty function method in EMLDE, which is only suitable for loose S-parameter constraints.

The GASPAD method, based on the introduction of SMAS, was proposed in [22] and aims to:

- develop a general mm-wave IC synthesis method starting from a given circuit topology, performance specifications, and some hints on layout (e.g., the metal layer to be used, the transistor layout template with different numbers of fingers), without any initial design nor the individual specifications of each stage;
- provide highly optimized results (including both objective function optimization and the satisfaction of multiple tight constraints) comparable to the results obtained by directly using a widely used EA-based constrained optimization method with embedded EM simulations, which is often the best synthesis method with respect to the solution quality;
- use much less computational effort compared with the above reference method, and, as such, make the computation time of the synthesis practical.

### 9.5.2 The GASPAD Method

The SMAS framework is designed for unconstrained expensive optimization. For the mm-wave IC synthesis problem with tight constraints, handling constraints and an appropriate method for selecting training data needs to be investigated.

Some SAEAs use the penalty function method to transform a constrained optimization problem to an unconstrained one in order to directly apply the SAEA for unconstrained optimization. The penalized cost function is given by:

$$f'(x) = f(x) + \sum_{i=1}^{i=c} w_i \langle g_i(x) \rangle \quad (9.11)$$

where  $f(x)$  is the objective function,  $g_i(x)$  is the  $i$ th constraint function, and the parameters  $w_i$  are the penalty weighting coefficients.  $\langle g_i(x) \rangle$  returns the absolute value of  $g_i(x)$  if it is negative, and zero otherwise, considering the constraints  $g_i(x) \geq 0, i = 1, 2, \dots, c$ . Although an SAEA for unconstrained optimization can be directly used when optimizing the penalized function  $f'(x)$ , the performance of the SAEA will be reduced. The reason is that a continuous and smooth hypersurface is important for generating high-quality surrogate models, but instead  $\langle g_i(x) \rangle$  are

piecewise functions. Moreover, various research works show that the static penalty function method is difficult to handle tight constraints [30].

A new constraint handling method is designed to be compatible with SMAS. GASPAD integrates constraint handling into the rules for ranking the newly generated candidates (see Step 2, Step 3, and Step 7 of SMAS). In other words, constraint satisfaction is considered to define the “best” candidate design in each iteration. The following ranking rules are presented:

1. The feasible design solutions (if any) are ranked higher than the infeasible design solutions.
2. The feasible design solutions (if any) are ranked based on the sorting of the objective function values in ascending order (assuming a minimization problem).
3. The infeasible design solutions are ranked based on the sorting of the sum of the constraint violation values in ascending order.

It can be seen that the ranking rules use the basic idea of a tournament selection method for constrained optimization [30], which is widely used in the EA field. Nevertheless, tournament selection based on a standard EA population is not used, but is modified to focus on the current best candidate design in order to match the proposed SAEA. Assuming that the prescreened best candidate design is a top ranked one in the generated candidate designs, the evolution can be divided into three phases. From the beginning to the appearance of the first feasible solution, GASPAD aims at minimizing the constraint violations (e.g., satisfying the  $G_p$ ,  $PAE$  specifications). From the appearance of the first feasible solution to where a considerable number of solutions are feasible in the current parent population, GASPAD searches for both objective function optimization (e.g., optimizing  $P_{1dB}$ ) and constraint satisfaction. Subsequently and until the end of the synthesis, GASPAD concentrates on optimizing the objective function. Note that independent surrogate models are constructed for each constraint, and this does not affect the smoothness and continuity of the hypersurface of the objective function and the constraint functions.

Appropriate training data points need to be selected to describe the current promising area, considering both objective function optimization and constraint satisfaction (Step 5 of SMAS). The promising area-based training data selection (PAS) method is used, which works as follows:

1. Calculate the median of the  $\lambda$  child solutions to obtain the vector  $mx$ .
2. Take the nearest  $c_1 \times d$  solutions to  $mx$  in the database (based on Euclidean distance).

The coefficient  $c_1$  is often selected from [5, 7]. According to the general idea of SMAS, both the  $\lambda$  child solutions and the training data points are around the current promising area. Thus, a set of training data points in proportion to the number of variables are selected to model the general trend of the targeted area. Because the nearest neighboring points are more useful than points far from the targeted area [8], the training data points are sorted based on their distance to  $mx$ .

### 9.5.3 Experimental Results

PA design is selected as an illustrative example of mm-wave IC design in this subsection. PA design is very difficult in the mm-wave IC design area because there are tedious tuning iterations between load–pull simulations and the design of impedance matching networks. Moreover, at mm-wave frequencies, not only the output matching network, but also the input and interstage matching networks need to be optimized to ensure sufficient output power and high efficiency.

A 60 GHz PA in a 65 nm CMOS technology is used as follows. (More examples and comparisons are in [22].) ADS Momentum is used as the EM simulator. Cadence SpectreRF is used as the circuit simulator. The bounds of the design variables are set both by the design rules of the technology and by the experience of the designer. The objective function is the power-added efficiency (PAE) at  $P_{1dB}$ , and the constraints are 1 dB compression point ( $P_{1dB}$ ) and the power gain ( $G_p$ ). GASPAD stops when the performance cannot be improved for 50 consecutive iterations. The examples are run on an Intel 2.66 GHz Dual Xeon PC under the Linux operating system and the MATLAB environment. The time measurements mentioned in the experiments correspond to wall clock time.

According to the parameter setting rules, the number of training data points ( $\tau$ ) is set to  $5 \times d$  and the population size ( $\lambda$ ) is set to 40. The number of initial samples,  $\alpha$ , is set to 70 for the PA with 18 design variables. For the GP modeling, the ooDACE toolbox [31] is used.

A 60 GHz two-stage PA with cascode differential pairs is synthesized. The layouts of the transistors with different numbers of fingers are designed beforehand in the form of layout templates, and the number of fingers of a transistor ( $nf$ ) in the driver stage is a design variable. The output load impedance is  $50 \Omega$ . The schematic is shown in Fig. 9.3. The design variables for the passive components are inner diameters of the primary and secondary inductors ( $d_{ins}$ ,  $d_{inp}$ ) and the metal width of these two inductors ( $ws$ ,  $wp$ ) for each of the three transformers. There are 5 biasing voltages:  $V_{DD}$ ,  $V_{cas1}$ ,  $V_{cas2}$ ,  $V_{b1}$ , and  $V_{b2}$ . The ranges for the design variables are summarized in Table 9.1. There are, in total, 18 design parameters.

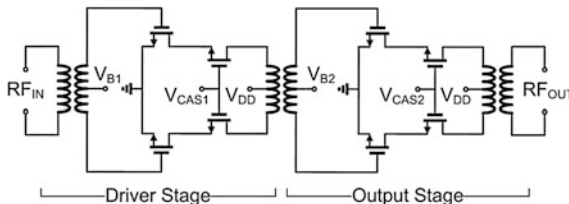


Fig. 9.3 Schematic of the 60 GHz power amplifier

**Table 9.1** Design parameters and their ranges for the 60 GHz power amplifier

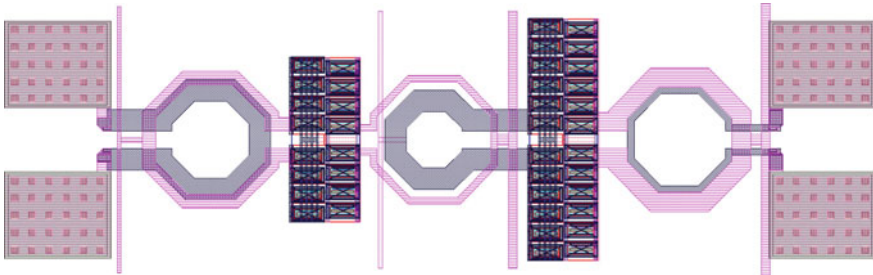
Parameters	Lower bound	Upper bound
dinp, dins ( $\mu\text{m}$ )	20	100
wp, ws ( $\mu\text{m}$ )	3	10
$V_{\text{DD}}$ (V)	1.5	2
$V_{\text{cas1}}$ (V)	1.2	2
$V_{\text{cas2}}$ (V)	1.2	2
$V_{\text{b1}}$ (V)	0.55	0.95
$V_{\text{b2}}$ (V)	0.55	0.95
nf (integer)	2	5

The synthesis problem is as follows:

$$\begin{aligned}
 &\text{maximize} && PAE (@P_{1\text{dB}}) \\
 &\text{s.t.} && P_{1\text{dB}} \geq 13 \text{ dBm} \\
 &&& G_p \geq 10 \text{ dB}
 \end{aligned} \tag{9.12}$$

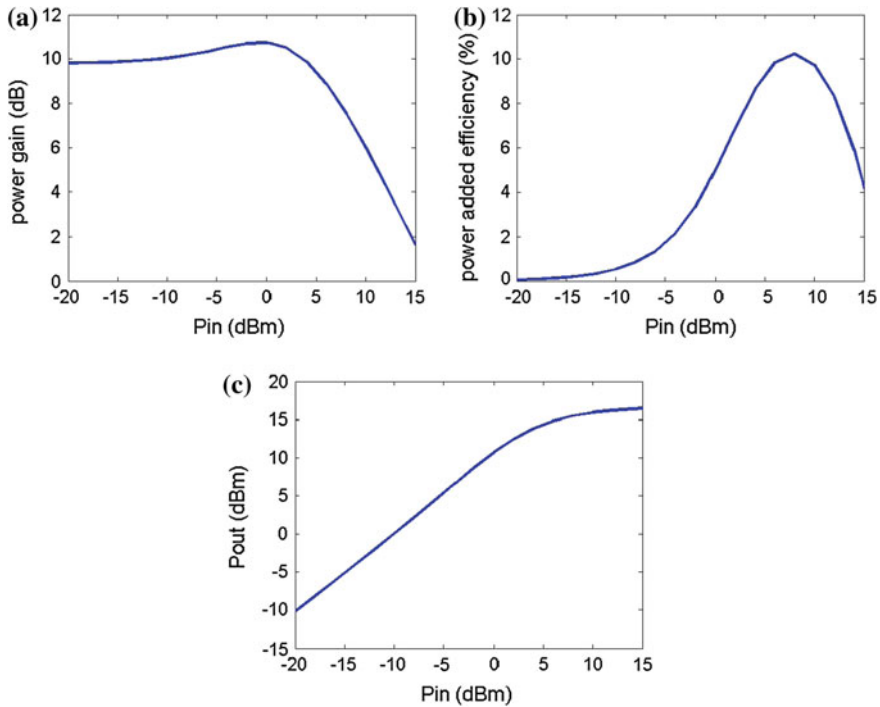
After 204 evaluations, GASPAD obtained the optimized design. The layout of the synthesized PA is shown in Fig. 9.4. The 1-dB compression point is 14.87 dBm, the power-added efficiency at  $P_{1\text{dB}}$  is 9.85 %, and the power gain is 10.73 dB. S-parameter simulation shows that the lowest of the Rollet stability factors ( $K$  factors) is 10.68, which is larger than 1, and  $|\Delta|$  is smaller than 1, so the obtained circuit design is unconditionally stable. The simulation results are shown in Fig. 9.5. The time consumption for GASPAD to synthesize this PA is 42 h.

A manual design using the same circuit structure and 65 nm technology has been reported in [32]. Its 1 dB compression point is 10.8 dBm, the power-added efficiency at  $P_{\text{sat}}$  is 7.2 %, and the power gain is 10.2 dB. Clearly, the synthesized design is better than the manual one on all of the three performances.



**Fig. 9.4** Layout of the PA synthesized by GASPAD





**Fig. 9.5** The simulated performances of the 60 GHz PA synthesized by GASPAD. **a** Power gain. **b** Power added efficiency. **c** Output power

## 9.6 NoC Parameter Design Optimization

### 9.6.1 NoC Design and Optimization

Today, there is a dramatic increase of intellectual property (IP) cores integrated on systems-on-chip (SoCs). Hence, network-on-chip is being adopted by the research community and industry as the underlying communication structure [33, 34]. An NoC consists of a network constructed of multiple point-to-point data channels (links) interconnected by routers. The routers are connected to a set of distributed IPs, and the communication among them usually utilizes a packet-switching method. In the packet-switching method, messages are divided into suitably sized blocks, which are called packets.

An important application of NoCs is chip multiprocessors (CMPs), which were introduced to provide near-linear improvements of performance over complexity (Pollack's rule [35]), while maintaining lower power and frequency budget. In a CMP, the number of cores is projected to increase rapidly, and good utilization of such cores is becoming an apparent challenge. CMP performance and power

consumption depend both on NoC and on cache coherence protocols. These protocols rely heavily on an underlying communication fabric to provide one-to-many (1-to-M) communication. A hybrid network architecture could retain the broadcasting capability of the buses and reduce the internode average hop count while maintaining high interconnect scalability when high-performance interconnect is adopted as the bus system (e.g., surface wave interconnects (SWIs)). The surface wave technology has been presented in [19], and the proposed hybrid wire–surface wave interconnects (W-SWIs) architecture utilizing this technology has shown excellent scalability and performance features (e.g., energy consumption and delay) [36, 37]. The W-SWI architecture is used in the NoC design optimization examples of this chapter.

The performance of an NoC is largely determined by the NoC architecture used and its design parameters. When the load is small, different architectures do not show much performance difference. With increasing load, performance difference becomes obvious for different architectures, but, for the same architecture, different design parameters do not show much performance difference. When the load further increases, obvious performance difference can be observed using different design parameters. This motivates optimization of architecture and parameters of the NoC.

However, due to the complexity of the problem, many designers prefer to adopt regular predefined architectures and design parameters when designing NoCs [38]. Clearly, this may fail to achieve optimal performance in various network traffic cases. Much performance improvement can be achieved if the NoC is optimized. Hence, there is some work on optimizing NoC topologies [39], but there are very few methods dealing with optimization of the design parameters of NoCs. Some case-specific methods to optimize one or a few key design parameters have been proposed (e.g., the placement of repeaters in global communication links [40]), and improved designs have been obtained. However, there is a lack of generality in most available methods, and many design parameters cannot be optimized, including some critical ones. This section therefore aims to provide a general method for NoC optimization considering all design parameters with a given architecture.

### **9.6.2 *The NDPAD Method***

This subsection introduces an SMAS-based NoC parameter design optimization method called NoC design optimization based on Gaussian process model-assisted differential evolution (NDPAD) [23].

Besides computationally expensive simulation, the NoC parameter design optimization problem also encounters constraints. Typically, there are one to three constraint(s), such as area, energy, and throughput. The tournament selection-based method from GASPAD can be directly applied. However, a more important challenge is that all of the NoC design parameters must be integers. Discrete variables pose challenges to both evolutionary search and surrogate modeling. In terms of

surrogate modeling, a discontinuous landscape needs to be modeled, but clearly a continuous and smooth landscape is good for surrogate modeling methods. In terms of search, research on standard EA shows that when directly using integers for encoding, the population diversity will decrease and it is much easier to be trapped in a local optimum. Therefore, the quantization method in Sect. 9.3 is often applied [16]. However, it is an open question whether simply applying the quantization method is enough for complex problems or not, especially for SAEA. Our pilot experiments on NoC problems show that satisfactory results can be obtained, but the robustness needs to be improved when only the quantization method in several SAEA frameworks, including SMAS, is applied.

When the reduction of the population diversity is considered for expensive optimization problems with integer variables, the appropriate DE mutation strategy must be investigated. In Sect. 9.3, we introduced DE/best/1 and DE/current-to-best/1 mutation. There is another widely used DE mutation strategy called DE/rand/1. Compared to DE/best/1, the best candidate in the current population is replaced by a randomly selected one. DE/best/1, DE/rand/1, and DE/current-to-best/1 trade off the convergence speed and the population diversity in different manners and are widely used in standard DE, especially the first two. The key idea of SMAS is to concentrate the search and the surrogate modeling in the current promising subregion, which is achieved by two factors: (1) the population update and (2) the mutation and crossover. Therefore, it is necessary to move the child population toward the current best solution in the DE mutation. The DE/current-to-best/1 and DE/best/1 strategies are thus appropriate to be used in SMAS, and the former can lead to higher diversity. Although DE/rand/1 is widely used in standard DE, it may not be appropriate for SMAS, because child solutions spreading in different subregions of the decision space may be generated and a high-quality surrogate model is often difficult to construct using such training data points. This conclusion is verified by experiments in [18].

DE/best/1 has faster convergence speed, while DE/current-to-best/1 shows more population diversity. Whether the additional population diversity of DE/current-to-best/1 compared to DE/best/1 has substantial help or not needs to be verified by real-world NoC test problems. However, a general experience is that DE/best/1 should be able to obtain a reasonably good design. The DE/best/1 strategy is especially useful for NoC with large dimensions when each simulation is very time-consuming because of its high convergence speed. DE/current-to-best/1 should have a higher ability to obtain even better results and has higher robustness, but more simulations may be needed. Therefore, DE/current-to-best/1 is more suitable for small-dimensional NoC optimization.

For integer variables, because of the rounding, the training data points around the promising subregion are fewer than those of continuous optimization problems. Also, the landscape to be modeled is discontinuous, which is more difficult to approximate. When using the PAS method from GASPAD, either insufficient training data points or training data points far from the current promising subregion may be selected, which affects the quality of the surrogate model negatively. To address this problem, a new simple empirical method called individual solution-based training data selection (ISS) method is used, which works as follows:

1. For each solution in the  $\lambda$  child solutions, take the nearest  $c_2 \times d$  solutions in the database (based on Euclidean distance) as temporary training data points.
2. Combine all the temporary training data points and remove the duplicate ones.

To trade off the model quality and the training cost, empirical results suggest  $c_2 \in [0.5, 1]$ . This method emphasizes the modeling of the area surrounding each child solution and also builds a single surrogate model for the whole population to improve the model quality, instead of building a separate model for each child solution [41].

### 9.6.3 Experimental Results

In this section, two NoC parameter design problems will be shown. As has been said, NoC with high mesh dimensions may take a long time to run a single simulation, and NDPAD is designed for such problems. However, when using those problems for testing, it is very difficult to compare NDPAD with standard EAs, because standard EAs may take an intractable time to run the optimization. Owing to this, NoC design of a  $6 \times 6$  mesh dimension is chosen as the first example to make the optimization time taken by standard EAs tractable, as each simulation takes about 10 s. However, this favors standard EA because of the small search space. References [21, 22] show that speed improvement increases when the number of design variables and the complexity of the problem landscape increase. In the second example, we use a  $15 \times 15$  NoC, where each simulation takes 15–20 min.

The hybrid architecture is used for both examples. The NoC simulator is programmed in the SystemC language. The reference method we used is the selection-based differential evolution algorithm (SBDE) [42], which uses the same tournament selection method [30] with the standard DE algorithm. SBDE with DE/current-to-best/1 is applied. SBDE has been used as the reference method in many applications and shows highly optimized results although computationally expensive [1]. The examples are run on a PC with Intel 2.66 GHz Dual Xeon CPU and 70 GB RAM on the Linux operating system. No parallel computation is applied yet in these experiments. All time measurements in the experiments correspond to wall clock time.

#### 9.6.3.1 Example 1

In this example, we minimize the average delay of packets navigating via the NoC fabric from their source to their final destination(s) of a  $6 \times 6$  NoC. The load environment is  $PIR = 0.009$ ,  $PS = 12$  flits in multicast 10 % uniform traffic, where  $PIR$  is the packet injection rate and  $PS$  is the packet size. The problem is formulated as follows:

$$\begin{aligned}
 &\text{minimize} && A_D(N_c, S_p, X_1, Y_1, \dots, X_4, Y_4) \\
 &\text{s.t.} && E(N_c, S_p, X_1, Y_1, \dots, X_4, Y_4) \leq 0.00335 J
 \end{aligned} \tag{9.13}$$

where  $N_c$  is the number of virtual surface wave channels,  $S_p$  is the number of global SWI arbiter grant periods, and  $(X_i, Y_i), i = 1, 2, \dots, 4$  are the locations of the master nodes. The ranges of the design variables are  $N_c \in [1, 16], S_p \in [1, 12]$  and all others  $\in [1, 6]$ . We use the parameter setting rules in Sect. 9.4.3 except that the population size  $\lambda$  is set to  $5 \times d$  for higher diversity and  $c_2$  in ISS is set to 0.5.

To observe the performance of NDPAD, NDPAD is compared with SBDE. In SBDE,  $F$  and  $CR$  are the same as NDPAD and the population size is set to 40, which is a normal setting considering both efficiency and population diversity. The experimental results are presented in Table 9.2. It is shown in Table 9.2 that NDPAD provides comparable result to SBDE. The median of the 5 runs for both methods is extracted. It is found that NDPAD converges when using 890 simulations (without improvement in consecutive iterations). To obtain this performance, SBDE uses 1920 simulations. It can be seen that NDPAD uses less than 50 % of the computational effort of SBDE to obtain comparable results. To obtain a satisfactory average delay below 24 clock cycles, only 250 simulations are needed for NDPAD. Another  $6 \times 6$  NoC optimization example using a different traffic in [23] shows that NDPAD uses about 13% of the computational effort of SBDE and with better results.

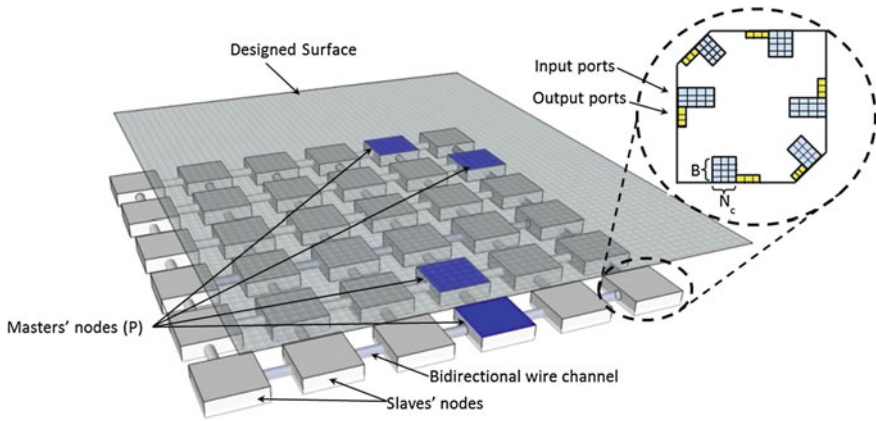
The optimized NoC is shown in Fig. 9.6. More detailed comparisons showing the effect of DE mutation strategy and the training data selection methods can be found in [23].

### 9.6.3.2 Example 2

The second example is a  $15 \times 15$  NoC with a load environment of  $PIR = 0.018, PS_{\min} = 2$  flits, and  $PS_{\max} = 12$  flits in multicast 0 % random traffic with hotspot traffic nodes at [6, 3], [13, 4], [4, 8], [6, 11], and [11, 12] with 3, 3, 2, 2, and 3 % rates, respectively. The problem is formulated as follows:

**Table 9.2** Comparison of NDPAD with SBDE

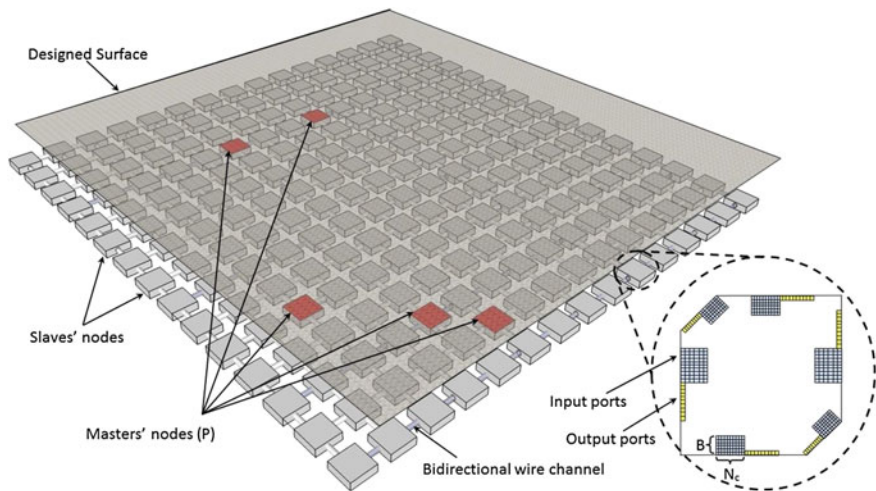
No. of runs	NDPAD		SBDE	
	$A_D$ /cycle	Constraint	$A_D$ /cycle	Constraint
1	23.0399	met	22.3288	met
2	23.0424	met	22.4656	met
3	22.2593	met	22.4656	met
4	22.5811	met	22.3722	met
5	23.4306	met	22.4656	met



**Fig. 9.6** An optimized design of a  $6 \times 6$  NoC

$$\begin{aligned}
 &\text{minimize} && A_D(N_c, S_p, B, X_1, Y_1, \dots, X_5, Y_5) \\
 &\text{s.t.} && E(N_c, S_p, B, X_1, Y_1, \dots, X_5, Y_5) \leq 0.16J \quad (9.14) \\
 &&& C_{\text{area}} \leq 3.78
 \end{aligned}$$

where  $B$  is the buffer depth of each channel of the router.  $C_{\text{area}}$  is the area cost metric. The area for the NoC is expected to be within  $850 \text{ mm}^2$ . When normalized by the NoC size  $15 \times 15$ ,  $C_{\text{area}} = 3.78$  is obtained. The ranges of the design variables are  $N_c \in [1, 10]$ ,  $S_p \in [1, 12]$ ,  $B \in [1, 8]$ , and all others  $\in [1, 15]$ .



**Fig. 9.7** An optimized design of a  $15 \times 15$  NoC

After 37 hours, convergence has been achieved and the result is  $A_D = 55.78$  cycles,  $E = 0.153$  J, and  $C_{\text{area}} = 3.58$ . The optimized NoC is shown in Fig. 9.7.

## 9.7 Conclusions

This chapter has presented the SMAS framework and its applications to mm-wave IC synthesis and NoC parameter design optimization. The SMAS framework unifies “evaluation for optimization” and “evaluation for surrogate modeling,” so that the search can focus on a small promising area and is appropriately supported by the carefully constructed surrogate model. Owing to this, the SMAS framework achieves significant improvement in terms of efficiency compared to SAEAs using the standard EA structure (especially for problems with dozens of variables) and makes the solution of many computationally expensive EDA problems possible. This framework was adopted and developed in terms of constraint handling for the application to mm-wave IC synthesis. Search strategy, parameter setting rules, and training data selection methods were investigated for the application of NoC parameter design optimization, where all the design variables are discrete. Experimental results have shown good performance for both applications. The SMAS framework is a general method and can also be applied and developed for applications of computationally expensive antenna synthesis, MEMS synthesis, optical device and system synthesis and process variation-aware IC synthesis, etc.

**Acknowledgments** We sincerely thank Mr. Dixian Zhao, Katholieke Universiteit Leuven, Belgium, and Mr. Mengyuan Wu, GroepT, Belgium, for valuable discussions and supports. F.V. Fernández thanks the support of Spanish MINECO and ERDF (TEC2013-45638-C3-3-R) and the Junta de Andalucía (P12-TIC-1481).

## References

1. Liu, B., Gielen, G., Fernández, F.V.: Automated Design of Analog and High-frequency Circuits: A Computational Intelligence Approach. Springer, Berlin (2013)
2. McConaghy, T.: Variation-Aware Analog Structural Synthesis: A Computational Intelligence Approach. Springer, Berlin (2009)
3. Dietrich, M., Haase, J.: Process Variations and Probabilistic Integrated Circuit Design. Springer, Berlin (2012)
4. Liu, B., Zhao, D., Reynaert, P., Gielen, G.G.: Synthesis of integrated passive components for high-frequency RF ICs based on evolutionary computation and machine learning techniques. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **30**(10), 1458–1468 (2011)
5. Nieuwoudt, A., Massoud, Y.: Variability-aware multilevel integrated spiral inductor synthesis. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **25**(12), 2613–2625 (2006)
6. Jin, Y., Olhofer, M., Sendhoff, B.: A framework for evolutionary optimization with approximate fitness functions. *IEEE Trans. Evol. Comput.* **6**(5), 481–494 (2002)
7. Jones, D., Schonlau, M., Welch, W.: Efficient global optimization of expensive black-box functions. *J. Global Optim.* **13**(4), 455–492 (1998)

8. Emmerich, M., Giannakoglou, K., Naujoks, B.: Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Trans. Evol. Comput.* **10**(4), 421–439 (2006)
9. Lim, D., Jin, Y., Ong, Y., Sendhoff, B.: Generalizing surrogate-assisted evolutionary computation. *IEEE Trans. Evol. Comput.* **14**(3), 329–355 (2010)
10. Zhou, Z., Ong, Y., Nair, P., Keane, A., Lum, K.: Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **37**(1), 66–76 (2007)
11. Goel, T., Haftka, R.T., Shyy, W., Queipo, N.V.: Ensemble of surrogates. *Struct. Multi. Optim.* **33**(3), 199–216 (2007)
12. Fogel, D.B.: *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, vol. 1. Wiley, New York (2006)
13. Rasmussen, C.: Gaussian processes in machine learning. *Advanced Lectures on Machine Learning*, pp. 63–71 (2004)
14. Dennis, J., Torczon, V.: Managing approximation models in optimization. *Multi. Des. Optim. State Art*, 330–347 (1997)
15. Couckuyt, I., Dhaene, T., Demeester, P.: ooDACE toolbox. *Adv. Eng. Softw.* **49**(3), 1–13 (2012). (Elsevier)
16. Price, K., Storn, R., Lampinen, J.: *Differential Evolution: A Practical Approach to Global Optimization*. Springer, Berlin (2005)
17. Rasheed, K., Hirsh, H.: Informed operators: speeding up genetic-algorithm-based design optimization using reduced models. In: *The Genetic and Evolutionary Computation Conference*, pp. 628–635 (2000)
18. Liu, B., Chen, Q., Zhang, Q., Gielen, G., Grout, V.: Behavioral study of the surrogate model-aware evolutionary search framework. In: *IEEE Congress on Evolutionary Computation (CEC)*, pp. 715–722. IEEE (2014)
19. Stein, M.: Large sample properties of simulations using Latin hypercube sampling. *Technometrics*, 143–151 (1987)
20. Liu, B., Deferm, N., Zhao, D., Reynaert, P., Gielen, G.: An efficient high-frequency linear RF amplifier synthesis method based on evolutionary computation and machine learning techniques. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **31**(7), 981–993 (2012)
21. Liu, B., Zhang, Q., Gielen, G.: A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive black box optimization problems. *IEEE Trans. Evol. Comput.* **18**(2), 180–192 (2013)
22. Liu, B., Zhao, D., Reynaert, P., Gielen, G.G.: GASPAD: a general and efficient mm-wave integrated circuit synthesis method based on surrogate model assisted evolutionary algorithm. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **33**(2), 169–182 (2014)
23. Wu, M., Karkar, A., Liu, B., Yakovlev, A., Gielen, G., Grout, V.: Network on chip optimization based on surrogate model assisted evolutionary algorithms. In: *IEEE Congress on Evolutionary Computation (CEC)*, pp. 3266–3271. IEEE (2014)
24. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.* **13**(2), 398–417 (2009)
25. Niknejad, A.: Siliconization of 60 GHz. *Microw. Mag. IEEE* **11**(1), 78–85 (2010)
26. Zhao, D., Kulkarni, S., Reynaert, P.: A 60-GHz outphasing transmitter in 40-nm CMOS. *IEEE J. Solid State Circuits* **47**(12), 3172–3183 (2012)
27. Agarwal, A., Vemuri, R.: Layout-aware RF circuit synthesis driven by worst case parasitic corners. In: *IEEE International Conference on Computer Design: VLSI in Computers and Processors*, pp. 444–449. IEEE (2005)
28. Allstot, D.J., Park, J., Choi, K.: *Parasitic-Aware Optimization of CMOS RF Circuits*. Springer, Berlin (2003)
29. Tulunay, G., Balkir, S.: A synthesis tool for cmos rf low-noise amplifiers. *Comput. Aided Des. Integr. Circuits Syst. IEEE Trans.* **27**(5), 977–982 (2008)



30. Deb, K.: An efficient constraint handling method for genetic algorithms. *Comput. Methods Appl. Mech. Eng.* **186**(2), 311–338 (2000)
31. Couckuyt, I., Forrester, A., Gorissen, D., De Turck, F., Dhaene, T.: Blind kriging: implementation and performance analysis. *Adv. Eng. Softw.* **49**, 1–13 (2012)
32. Zhao, D., He, Y., Li, L., Joos, D., Philibert, W., Reynaert, P.: A 60 GHz 14 dBm power amplifier with a transformer-based power combiner in 65 nm CMOS. *Int. J. Microw. Wireless Technol.* **3**(02), 99–105 (2011)
33. Salihundam, P., Jain, S., Jacob, T., Kumar, S., Erraguntla, V., Hoskote, Y., Vangal, S., Ruhl, G., Borkar, N.: A 2 tb/s 6 by 4 mesh network for a single-chip cloud computer with DVFS in 45 nm CMOS. *IEEE J. Solid State Circuits* **46**(4), 757–766 (2011)
34. Dally, W.J., Towles, B.: Route packets, not wires: On-chip interconnection networks. In: *Design Automation Conference*, pp. 684–689. IEEE (2001)
35. Borkar, S., Chien, A.A.: The future of microprocessors. *Commun. ACM* **54**(5), 67–77 (2011)
36. Karkar, A., Dahir, N., Al-Dujaily, R., Tong, K., Mak, T., Yakovlev, A.: Hybrid wire-surface wave architecture for one-to-many communication in networks-on-chip. In: *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2014, pp. 1–4. IEEE (2014)
37. Karkar, A.J., Turner, J.E., Tong, K., Ra’ed, A.D., Mak, T., Yakovlev, A., Xia, F.: Hybrid wire-surface wave interconnects for next-generation networks-on-chip. *IET Comput. Digital Tech.* **7**(6), 294–303 (2013)
38. Benini, L., Bertozzi, D.: Network-on-chip architectures and design methods. In: *IEE Computers and Digital Techniques*, vol. 152, pp. 261–272. IET (2005)
39. Tino, A., Khan, G.N.: High performance NoC synthesis using analytical modeling and simulation with optimal power and minimal IC area. *J. Syst. Architect.* **59**(10), 1348–1363 (2013)
40. Banerjee, K., Mehrotra, A.: A power-optimal repeater insertion methodology for global interconnects in nanometer designs. *IEEE Trans. Electron. Devices* **49**(11), 2001–2007 (2002)
41. Li, R., Emmerich, M.T., Eggermont, J., Bovenkamp, E.G., Back, T., Dijkstra, J., Reiber, J.H.: Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis. In: *IEEE Congress on Evolutionary Computation*, pp. 2764–2771 (2008)
42. Zielinski, K., Laur, R.: Constrained single-objective optimization using differential evolution. In: *IEEE Congress on Evolutionary Computation*, pp. 223–230 (2006)

# Chapter 10

## Computational Intelligence Techniques for Determining Optimal Performance Trade-Offs for RF Inductors

Elisenda Roca, Rafael Castro-López, Francisco V. Fernández, Reinier González-Echevarría, Javier Sieiro, Neus Vidal and José M. López-Villegas

**Abstract** The automatic synthesis of integrated inductors for radio frequency (RF) integrated circuits is one of the most challenging problems that RF designers have to face. In this chapter, computational intelligence techniques are applied to automatically obtain the optimal performance trade-offs of integrated inductors. A methodology is presented that combines a multi-objective evolutionary algorithm with electromagnetic simulation to get highly accurate results. A set of sized inductors is obtained showing the best performance trade-offs for a given technology. The methodology is illustrated with a complete set of examples where different inductor trade-offs are obtained.

### 10.1 Introduction

Computational intelligence techniques have been successfully incorporated in automated design methodologies for analog integrated circuits (ICs). However, a reduced number of approaches are reported in the design of radio frequency (RF) ICs, the main reason for this being the important and unresolved challenges that designers of RF ICs have to face. Among these challenges, the design of integrated inductors is probably one of the most difficult to overcome. Inductors are used in RF ICs for input/output matching networks, passive filters, low-noise amplifiers, oscillators, etc. In all of these circuits, inductors have to be carefully designed or selected, since both, circuit performances and area, are extremely

---

E. Roca (✉) · R. Castro-López · F.V. Fernández · R. González-Echevarría  
Instituto de Microelectrónica de Sevilla, IMSE-CNM, CSIC and Universidad de Sevilla,  
Seville, Spain  
e-mail: eli@imse-cnm.csic.es

J. Sieiro · N. Vidal · J.M. López-Villegas  
Departament d'Electrònica, Universitat de Barcelona, Barcelona, Spain

dependent on them. The main inductor performances, namely equivalent inductance ( $L_{eq}$ ), quality factor ( $Q$ ), self-resonance frequency ( $SRF$ ), and area, are closely related to the inductor geometric parameters and some are frequency dependent, but they are also conflicting, i.e., one cannot be improved without worsening other. Therefore, finding the inductor with the appropriate performance trade-off for a specific application is a complicated task. This task is even more complex since accurately evaluating the inductor performances requires long and computationally expensive electromagnetic (EM) simulations. In automated design methodologies, this problem is especially important due to the large number of inductors that has to be iteratively evaluated, making the incorporation of EM simulation in the design flows almost impossible.

Previous approaches reported in the literature deal with the inductor sizing problem either (a) indirectly, by optimizing the performances of the RF circuits in which they are embedded [1–6], or (b) directly, by optimizing inductor performances such as  $L_{eq}$ ,  $Q$ ,  $SRF$ , and area [7–16]. All of them share a common goal: They are intended for what it is called online optimization, where, as illustrated in Fig. 10.1a, the inductors are designed for some given specifications at the moment they are needed during the design of an RF circuit. This brings a critical efficiency versus accuracy trade-off. Since accurate EM simulation takes too long, it is not possible to embed it into an optimization loop; therefore, some kind of approximate analytical or surrogate-based model of the inductor performances is used, sometimes with iterative improvement of the model accuracy as the optimization progresses [9, 17]. The different approaches reported in the literature vary in the optimization technique, the kind of approximate model used for performance evaluation or how the performance evaluator is integrated with the optimization technique. Also, most of these approaches have in common that they are intended for the design of a single inductor, e.g., they are aimed to obtain a certain inductance and quality factor at a given frequency with minimum area occupation with some constraints on the self-resonance frequency. Trade-offs between inductor performances are actually not of interest for these approaches.

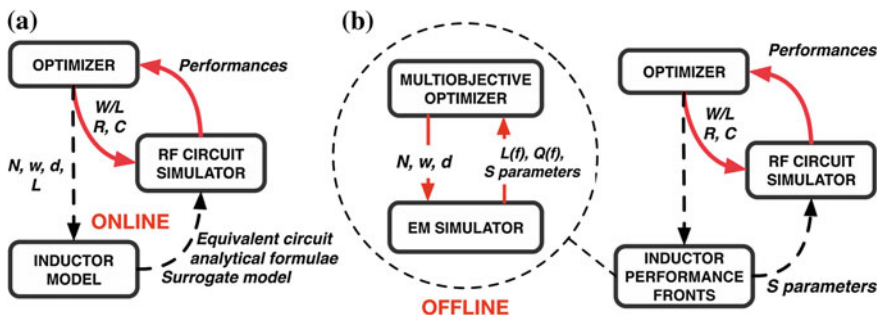


Fig. 10.1 RF circuit design flow with a online and b off-line optimization of inductors

A new RF design paradigm based on off-line optimization of inductor performances is proposed by the authors, where computational intelligence techniques are incorporated [18]. Instead of designing a specific inductor when it is needed, the set of inductors showing the best performance trade-offs for a given technology is generated by formulating a multi-objective optimization problem. As no specific performance value is pursued in this approach, the generation can be performed off-line, i.e., much before some specific inductance or quality factor values are required for an RF circuit design problem. This implies that the critical efficiency versus accuracy trade-off of online optimization approaches is avoided. Considerably higher computation times are acceptable, and accurate EM simulation becomes an attractive option for performance evaluation in the multi-objective optimization algorithm. Figure 10.1 illustrates the off-line optimization concept together with the online optimization concept. Although long simulation times are acceptable in the off-line approach, efficiency enhancement techniques have been applied, such as parallelization of the optimization process, adaptive meshing and smart selection of sampling frequencies in the EM simulation, command line programming control of commercial tools, and adaptive stopping criteria that will halt the optimization algorithm when no further improvement can be obtained.

This approach can be used for more efficient RF circuit design. In addition, performance fronts are generated which enable easy exploration of trade-offs between the inductor performances.

This chapter is organized as follows. Section 10.2 presents related work reported in the literature from a double point of view: inductor performance optimization and optimization of RF circuits embedding integrated inductors. The multi-objective optimization problem for the inductor performances is defined in Sect. 10.3, and then, its implementation is presented in Sect. 10.4. Experimental results are shown in Sect. 10.5.

## 10.2 Related Work

Most automated design approaches for RF ICs are based on optimization procedures coupled to some performance evaluation of the circuit at hand. Inductors are incorporated into the optimization flow in different ways. Some approaches rely on foundry-provided inductors [19–21]. However, these libraries are typically limited to a few tens of devices characterized for some typical working frequencies, representing a much reduced part of the inductor design space. Therefore, using these inductors frequently results in suboptimal circuit performances.

Since accurate electromagnetic evaluation of the inductor performances is computationally expensive, most optimization-based approaches have tried to avoid or drastically limit its use. Equivalent lumped circuit models are used instead in [3–5] with analytical [11, 13] or posynomial [22] models for each circuit element. The advantage of these latter methods is the fast evaluation of the inductor performances, although they can be considered only as a first-order approximation:

Errors above 10 % have been reported for inductance estimation and much larger errors for the inductor's resistance [2]. Moreover, these errors dramatically increase if wide ranges of inductor geometries, higher frequencies, or different topologies are considered.

Trying to reduce these errors, more accurate simulations of a limited number of inductor geometries for a given operating frequency together with the interpolation of intermediate values have been used for the design of power amplifiers [1, 6]. EM simulation for the selected inductances was used in [1], whereas ASITIC [12] or FastHenry [23] was used in [6]. The drawback of these approaches is that the limited number of devices simulated introduces large errors in the interpolation. Besides, the inductors selected for interpolation have not been previously optimized.

Analytical formulae for inductance and parasitics are used in [2], whereas a more accurate simulation is performed for the inductor losses: FastHenry or a full-wave electromagnetic simulator depending on the substrate characteristics. Simulation complexity is reduced by approximating to a circular inductor. The accuracy is certainly improved at the expense of 24-h computation time for an optimization process with 700 iterations, usually considered too low of a number unless the search space is relatively small.

In all these works, inductors are optimized indirectly by optimizing the performances of the circuits in which they are included. Other reported approaches are specifically devoted to optimization of inductor performances. Exhaustive enumeration and binary search algorithms have been implemented in ASITIC and COILS [14], allowing the selection of the inductor with the highest quality factor for a given value of inductance, which is found by sampling the search space. Both techniques, however, need a high number of evaluations and are inefficient in selecting the optimum inductor.

A more efficient approach than enumeration is geometric programming [8], but this method implies that the design problem must be formulated in terms of posynomial functions, which is not always possible with acceptable accuracy. Sequential quadratic programming has been used in [16], improving the optimization time of exhaustive enumeration by an order of magnitude. Furthermore, the possibility of using any physical model circumvents the problems of geometric programming. However, the accuracy of the results is limited by the use of those approximate models.

The approaches in [11, 13] also use equivalent circuit models and approximate analytical models of its components, hence suffering from the same accuracy problems. Unlike the other works reviewed in this section, that presented in [11] generates a set of Pareto designs, i.e., points that show some trade-offs among different performances. These points are obtained by solving a sequence of single-objective optimization problems in which one performance is minimized or maximized and several sets of constraint values are assigned to the other performances. Eventually, a surrogate model, i.e., a compact analytical model that approximates data behavior using a limited set of points, is generated using the

Pareto designs. This surrogate model is then used as a feasibility model for the optimization problem of a single inductor.

Other approaches replace the equivalent circuit model by a surrogate model approximating the inductor behavior using a limited set of accurate EM simulations. Artificial neural networks with 500 EM-simulated inductor samples are used in [10]. Optimization seeks only a given inductance with no specification on quality factor and with operating frequencies much below the self-resonance frequency, where the errors in the surrogate model are expected to be much smaller.

The work reported in [15] uses a formulation similar to that in [8] as a coarse (and, therefore, very inaccurate) surrogate model. A few EM simulations are used to adjust the model parameters and improve its accuracy.

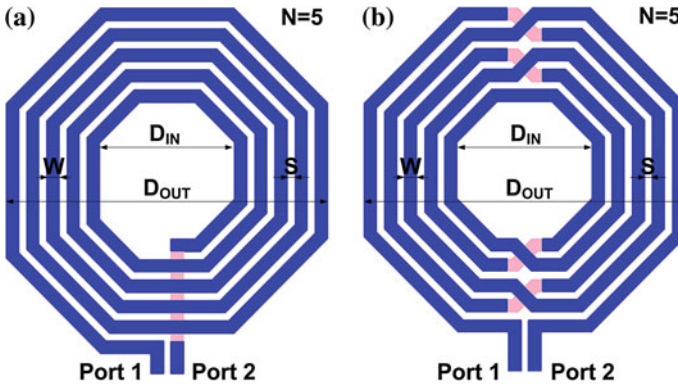
A model for the inductance and quality factor values as a function of the geometric parameters is generated using EM-simulated points and the regularization theory in [7]. The optimum inductor is obtained by first locating an optimal region with global optimization algorithms and, after that, iteratively improving the model in that region with additional EM simulations until convergence is reached. The optimization starts from a coarse model, and for this reason, the search process can be biased to a non-optimal region.

A similar approach is followed in [9], where a small set of samples, selected using Latin hypercube sampling techniques, are used to train an initial surrogate model. Then, this model is used in an optimization process, and it is improved at each iteration with the EM simulation of the candidate solution with the best potential. The best potential accounts not only for the value predicted by machine learning techniques but also for the prediction uncertainty. This may yield exploration of many solutions until a sufficiently low prediction uncertainty is achieved. Moreover, only single-objective optimization of inductors is addressed.

All works reviewed above rely on approximate equivalent circuit models or surrogate models that approximate the inductor behavior using a limited set of accurate simulations. They are intended for online optimization of inductors for some specific performances, i.e., their required performances are known when the RF circuit in which they are embedded is designed. As the optimization implies a repetitive evaluation, practical time constraints require that performance evaluation is efficient enough, hence sacrificing accuracy. Such a tight trade-off is avoided by the generation of optimal performance trade-offs presented in the following sections.

### 10.3 Definition of the Inductor Optimization Problem

The most frequently used inductors in CMOS technologies, and the focus of this chapter, are planar spiral inductors. Several geometric topologies of inductors (e.g., square or octagonal) are typically used in CMOS technologies. The geometry of a planar spiral inductor is usually defined by four geometric parameters: the number of turns ( $N$ ), the diameter of the inner hole ( $D_{IN}$ ), the turn width ( $W$ ), and the



**Fig. 10.2** Geometric parameters for **a** an octagonal asymmetric spiral inductor and **b** a symmetric inductor, for the same values of the design variables

spacing between turns ( $S$ ). The outer diameter ( $D_{OUT}$ ) can be obtained from the other geometric parameters. Inductors can also be classified depending on whether their structure is asymmetric or symmetric. Figure 10.2 illustrates these parameters on a planar octagonal inductor, where both topologies, asymmetric and symmetric, are presented. Though the total occupied area is not an electrical characteristic, it is, however, an essential parameter as it is directly related to the fabrication cost. It can be directly estimated from the geometric characteristics in Fig. 10.2.

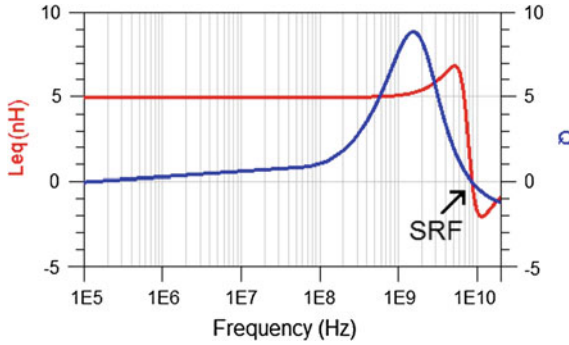
The most relevant performances of inductors are  $L_{eq}$  and  $Q$ , which are defined as follows:

$$L_{eq}(f) = \frac{\text{Im}[Z_{eq}(f)]}{2\pi f} \quad (10.1)$$

$$Q(f) = \frac{\text{Im}[Z_{eq}(f)]}{\text{Re}[Z_{eq}(f)]} \quad (10.2)$$

where  $f$  is the operating frequency and  $Z_{eq}$  is the equivalent input impedance. An important parameter is the self-resonance frequency,  $SRF$ , which is defined as the frequency at which the imaginary part of  $Z_{eq}$  is zero, or the frequency at which the behavior of the inductor changes from inductive to capacitive, as shown in Fig. 10.3.

The value of  $Z_{eq}$  depends on the excitation conditions of the circuit where the inductor is included. Two different situations are possible: (a) The inductor is driven by a non-differential excitation, as shown in Fig. 10.4a; (b) the inductor is driven by a differential excitation, as shown in Fig. 10.4b. In order to correctly define all these parameters, both situations have to be studied separately [24]. Let us first consider



**Fig. 10.3** Equivalent inductance and quality factor as a function of the operating frequency for the inductor in Fig. 10.2a in a 0.35- $\mu\text{m}$  CMOS technology

an inductor as a two-port structure. For a non-differential excitation, the equivalent input impedance is given by

$$Z_{\text{eq}}(f) = Z_{\text{in}}(f) = Z_o \frac{1 + \Gamma_{\text{in}}(f)}{1 - \Gamma_{\text{in}}(f)} \tag{10.3}$$

where  $Z_o$  is the characteristic impedance or surge impedance at the inductor input and  $\Gamma_{\text{in}}(f)$  is the input reflection coefficient. The value of  $\Gamma_{\text{in}}(f)$  is given by

$$\Gamma_{\text{in}}(f) = \frac{S_{11} - |S|\Gamma_L}{1 - S_{22}\Gamma_L} \tag{10.4}$$

where  $|S| = S_{11}S_{22} - S_{12}S_{21}$ ,  $S_{11}$ ,  $S_{12}$ ,  $S_{21}$  and  $S_{22}$  being the scattering parameters of the two-port structure, and  $\Gamma_L$  the reflection coefficient associated with the impedance load at the output of the two-port structure. For a differential excitation, the value of the equivalent input impedance is given by

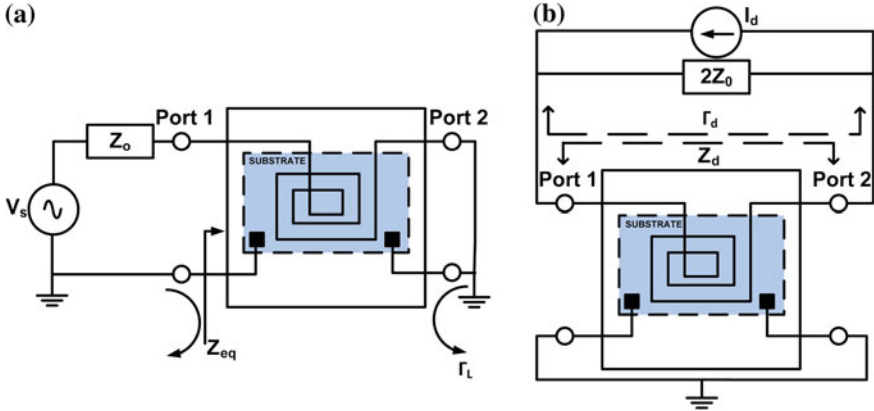
$$Z_{\text{eq}}(f) = Z_d(f) = 2Z_o \frac{1 + \Gamma_d(f)}{1 - \Gamma_d(f)} \tag{10.5}$$

where the differential input reflection coefficient,  $\Gamma_d$ , is given by

$$\Gamma_d(f) = \frac{S_{dd} - |S|\Gamma_L^c}{1 - S_{cc}\Gamma_L^c} \tag{10.6}$$

where  $|S| = S_{dd}S_{cc} - S_{cd}S_{dc}$ ,  $S_{dd}$ ,  $S_{cc}$ ,  $S_{cd}$  and  $S_{dc}$  being the mixed-mode scattering parameters of the inductor two-port structure, which are directly related to the  $S$ -parameters [25], and  $\Gamma_L^c$  is the reflection coefficient associated with the common-mode load impedance. Typically, symmetric inductors are used in differentially driven circuits, whereas asymmetric inductors are used in non-differential circuits and mostly in single-ended configuration. Therefore, the performance of each type of inductor must be evaluated according to these situations.  $Z_{\text{eq}}$  can be calculated





**Fig. 10.4** Two-port inductor driven **a** in a non-differential excitation mode and **b** in a differential excitation mode

from Eqs. (10.4) and (10.6) taking into account that for symmetric inductors,  $S_{cd} = S_{dc} = 0$ , and for asymmetric inductors,  $\Gamma_L = -1$  (short-circuit condition at the output). From these values, the equivalent inductance and quality factor can be obtained.

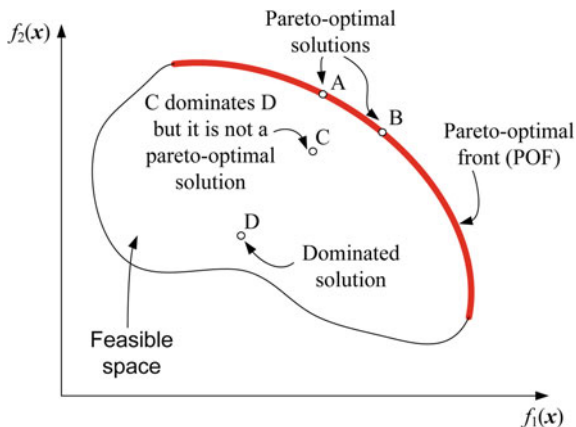
Finding the optimal trade-offs between the inductor performances (inductance, quality factor at one or more frequencies and occupied area) is formulated as a multi-objective optimization problem:

$$\left\{ \begin{array}{l} \text{Maximize } \mathbf{F}(\mathbf{x}); \quad \mathbf{F}(\mathbf{x}) = \{f_1(\mathbf{x}), \dots, f_n(\mathbf{x})\} \in \mathbb{R}^n \\ \text{such that: } \mathbf{G}(\mathbf{x}) \geq 0; \quad \mathbf{G}(\mathbf{x}) = \{g_1(\mathbf{x}), \dots, g_m(\mathbf{x})\} \in \mathbb{R}^m \\ \text{where } x_{Li} \leq x_i \leq x_{Ui}, \quad i \in [1, p] \end{array} \right. \quad (10.7)$$

where  $\mathbf{x}$  is a vector with  $p$  design variables, each design variable value being restricted between a lower limit ( $x_{Li}$ ) and an upper limit ( $x_{Ui}$ ). Functions  $f_j(\mathbf{x})$ , with  $1 \leq j \leq n$ , are the objectives that will be maximized, and  $g_k(\mathbf{x})$ , with  $1 \leq k \leq m$ , are the constraint functions. Objectives that should be minimized can be easily transformed into a maximization problem by just inverting their sign. The space that contains all possible solutions is known as the feasible space. The goal of multi-objective optimization is to provide the best trade-offs among solutions in the feasible objective space.

The multi-objective optimization evolutionary algorithm (MOEA) NSGA-II [26] was selected as optimization algorithm in our approach, although its role could be replaced by any of the tens of multi-objective optimization algorithms reported in the literature. NSGA-II is based on the evolution of a population of solutions (also called individuals) guided by the concept of Pareto dominance (see Fig. 10.5). That is, given the maximization problem in (10.7), an individual,  $\mathbf{x}_A$ , is said to dominate

**Fig. 10.5** Representation of the feasible objective space, Pareto-optimal front, and dominance concepts



another individual,  $\mathbf{x}_C$ , if  $f_j(\mathbf{x}_A) \geq f_j(\mathbf{x}_C)$  for all “ $j$ ” values and the “ $>$ ” relation is verified for at least one function. Solution  $\mathbf{x}_A$  is said to be non-dominated if no other individual dominates it. The set of non-dominated solutions provided by the iterative evolution of this population is usually known as the Pareto-optimal front (POF), and it is the solution to the optimization problem [26]. Constraints in our constrained optimization problem are handled by applying the constraint-domination condition introduced in [27].

For the inductor optimization, the vector of design variables may be given by the number of turns, the inner diameter, the width of the turns, and the spacing among turns. The range of possible values for each variable has also to be set. Although wide ranges of variables are generally allowed to enable a wide exploration of the design space, the technology process may impose some bounds. For instance, it imposes a lower limit on metal widths in the inductor turns or a minimum size of the inner inductor diameter. An upper limit can also be obtained from a reasonable maximum area of the inductors. A variation grid is also needed, which changes for each type of variable, due to technology process limitations (e.g., the minimum grid for turn widths is limited by the minimum change of metal widths allowed) or layout restrictions (e.g., the number of turns of the inductors can be an integer number, or change in quarters of a turn only).

The vector of the objectives to optimize depends on the design scenario. For example, if the trade-off between the inductance, quality factor, and area is explored, the objective vector will be  $\mathbf{F}(\mathbf{x}) = \{L_{eq}, Q, A\}$ , but other trade-offs can also be studied that include, for instance, the *SRF*. Constraints are imposed through the vector  $\mathbf{G}(\mathbf{x})$  to ensure that the inductor’s behavior agrees with operating specifications (e.g., location of *SRF* with respect to operating frequency, maximum inductor area, etc.). The final POF obtained includes a set of inductors that represents the best trade-offs among those performances selected as objectives and that meets the constraints.

### 10.4 Optimization Flow

The proposed flow is shown in Fig. 10.6. After setting the necessary input data, i.e., inductor topology, design variables, performance objectives, constraints, and range of possible values for each variable, the flow starts by randomly generating the initial population of individuals whose design variables are set within the ranges previously defined. In order to evaluate the individuals of this and subsequent populations, the layout of each inductor (according to the values of its design variables) has to be first generated automatically. Then, each inductor of the population is electromagnetically simulated, obtaining the *S*-parameter matrix, which is used to calculate the values for inductance and quality factor at certain frequencies. Postprocessing of these data provides the objective and constraint values for all the inductors of the population. These values are feedback to the optimization algorithm, and the individuals of this population are merged with the population of the previous generation. Then, individuals of the merged population are ranked, and the best ones are selected as the population for the next generation. If this population meets some stopping criteria, the optimization flow stops and the non-dominated set of inductors (the

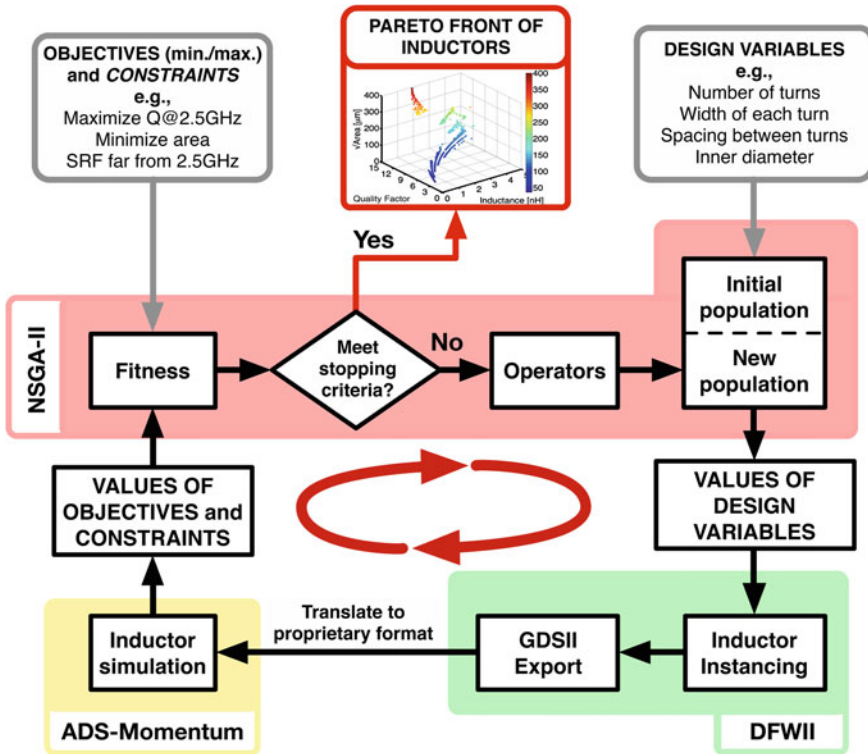


Fig. 10.6 Block diagram of the proposed optimization flow

Pareto-optimal front), which represents the best trade-offs among the inductor performances, is returned. If the stopping criteria are not fulfilled, a new population of inductors is created by applying to the previous population different operators, such as crossover and mutation. The complete process is then repeated. In the following subsections, the most relevant parts of this flow are described.

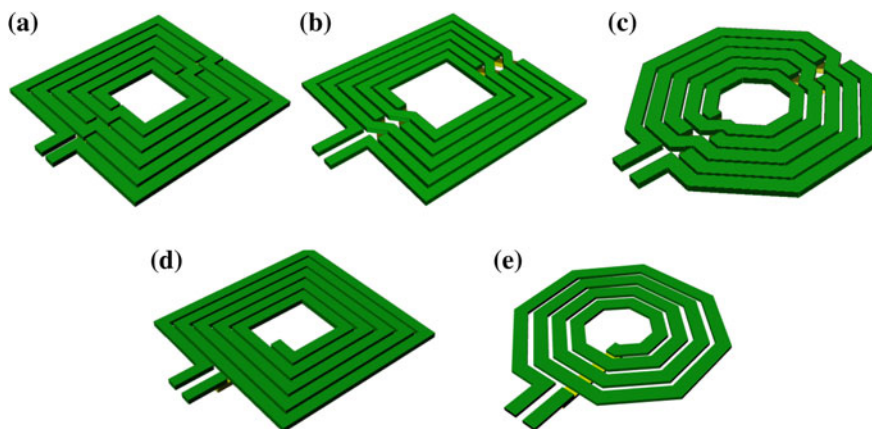
### 10.4.1 Layout Generation

Previously to the evaluation of the performances of the individuals of the population, the layout of each inductor has to be generated according to the values of its design variables. Parameterized cells (Pcells) are used to instance the inductor layout in Cadence Virtuoso platform for the corresponding inductor topology, then exported to GDSII format, and transformed into a layout format compatible with the EM simulator used for evaluation. The entire task is controlled automatically to increase efficiency.

Figure 10.7 illustrates the library of parameterized inductor cells implemented, with five different inductor topologies. The Pcells are technology independent, so migrating to a new technological process is straightforward.

### 10.4.2 Evaluation of the Inductor Performances

As stated above, the off-line approach allows to bet for a high-accuracy evaluation rather than for a very fast one. Therefore, the electromagnetic simulator ADS



**Fig. 10.7** Implemented P cell inductor topologies: **a** symmetric square with bridges at  $90^\circ$ ; **b** symmetric square with bridges at  $45^\circ$ ; **c** symmetric octagonal; **d** spiral square; **e** spiral octagonal

Momentum has been used in our implementation. Green functions associated with the substrate are calculated before any EM simulation is started. Although computationally expensive, once calculated for a certain frequency range, the information can be reused for any simulation within that frequency range. Therefore, the substrate is precomputed over a wider frequency range than any simulation is expected to need. The simulation process basically has two phases:

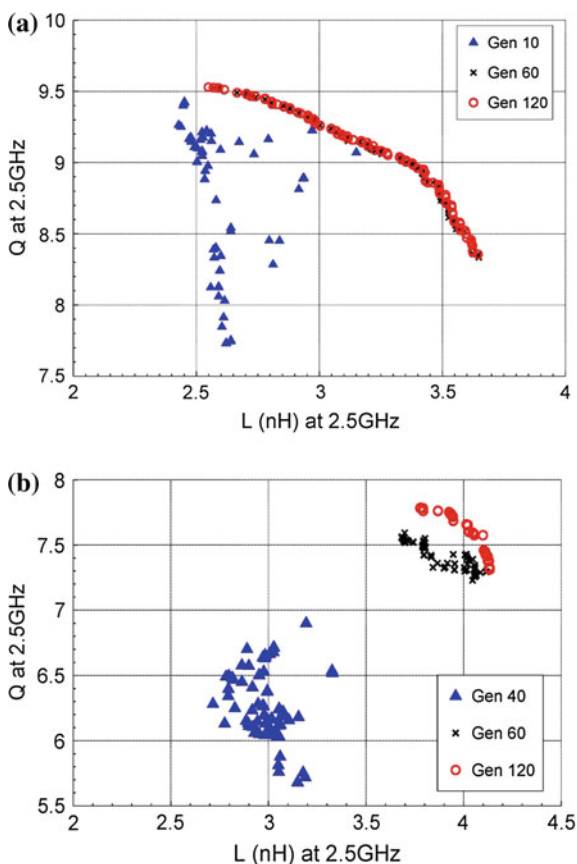
- **Configuration of Momentum:** Several parameters have to be set before starting the simulation, such as location of ports, the type of ports (that determine the simulation type), layers and substrate definition, the precomputed substrate files, simulation frequencies, and mesh configuration. Mesh definition is one of the most important steps because it has a major impact on both the accuracy of the simulated results and the simulation time. During the mesh definition, the layout is divided by creating a grid pattern of cells of a certain size, depending, among other parameters, on the simulation frequency and width and length of the metal lines. The generated mesh, which is specific to each inductor layout and it is generated before each simulation, changes according to the width of the inductor's turn, selecting a denser mesh of the outer cells for wider turns. This mesh adaptation mechanism provides an optimum trade-off between accuracy and simulation time.
- **Evaluation:** The inductor layout is then simulated with momentum, and the  $S$ -parameter matrix is extracted for the desired frequencies. Equations (10.1)–(10.6) are used to calculate the equivalent inductance and quality factor for each inductor from its  $S$ -parameters. The simulation time grows linearly with the number of frequency points. Therefore, a smart frequency sampling technique is developed in each optimization so that the necessary information is obtained with a minimum number of simulated frequency points. For instance, determining the location of the  $SRF$  may require simulation over numerous frequency points. However, in practice, it is not usually required to know the exact location of the  $SRF$ , but just to know that the frequency at which the inductor behavior becomes capacitive is sufficiently above the frequencies of interest. This can be guaranteed by just ensuring that the frequency at which the maximum quality factor is obtained is at or beyond the maximum operating frequency. This can be checked by calculating the quality factor at just two frequency points, as will be shown later in the “Experimental Results” section.

### 10.4.3 Stopping Criteria for the Multi-objective Algorithm

The MOEA NSGA-II has been described in numerous papers and books [27] where the interested reader can find further details. However, this being a computationally expensive optimization problem, the stopping criteria shown in Fig. 10.6 play an especially important role due to its impact on the computational cost. Stopping criteria essentially determine when the optimization process can be stopped. This

problem has not received much attention in the evolutionary computation literature because evaluation of objectives and constraints in mathematical benchmark functions is typically inexpensive. Hence, most research efforts have been focused on just improving the optimization results. Typically, evolutionary algorithms are stopped after a predefined number of generations. However, setting this number is not a trivial problem. If the number is too low, the obtained POF will be far from the ideal one, and the usefulness of the results will dramatically decrease. If the number is too high, it will imply a large waste of computational resources as each extra generation in our optimization problem may take a number of hours to compute. Figure 10.8 illustrates the difference in convergence of the optimization algorithm for two examples with two objectives (inductance and quality factor). As it is shown in Fig. 10.8a, there are very small improvements in the set of solutions from generation 60 to generation 120; therefore, the evolution could be stopped

**Fig. 10.8** Sets of solutions at different generation counts for two different experiments



much before generation 120. However, in Fig. 10.8b, the set of solutions has evolved very significantly from generation 60–120. Unfortunately, there is not a universal rule about the appropriate setting, as it depends on the number of individuals in each generation, the number of objectives, the number and variation range of the design variables, and the complexity of the optimization problem.

Thus, smart stopping criteria aim at stopping the optimization algorithm when no significant improvement in the obtained POF is expected from allowing additional iterations. Two different POF properties needed to be measured to evaluate this improvement: convergence and diversity. Convergence refers to how close the obtained POF and the ideal POF are. Diversity encompasses extent (i.e., maximum range of objective values within the established constraints) and uniformity (i.e., distribution of points along the POF). As an improvement in convergence can be masked by a degradation of uniformity, three separate metrics for convergence and diversity are monitored [28]. To measure uniformity, the spacing metric [27] is used. To monitor extent, an extent ratio metric is introduced. The extent ratio is defined as the ratio of the extent metric of the populations of two consecutive generations. The extent metric is defined as the volume of the hypercube given by the minimum and maximum values of each objective in the current front. A possible metric to monitor convergence is the generational distance (GD) defined as follows:

$$GD = \frac{(\sum_{i=1}^n d_i^p)^{1/p}}{|\text{POF}_{\text{approx}}|} \quad (10.8)$$

where  $d_i$  is the Euclidean distance of each member in the current front ( $\text{POF}_{\text{approx}}$ ) to the closest point in the ideal POF, and  $p$  is set to 2 [29]. However, the ideal POF is not known in our case. Therefore, a new metric, the dominance-based generational distance [28], is used where the generational distance between the current front and the ideal front is replaced by the generational distance between the current front and that of the previous generation and such distance calculation is restricted to the solutions of the second front that are dominated by each solution of the first front.

## 10.5 Experimental Results

The objective of this section is to illustrate how to use this methodology to model different trade-offs between inductor performances as POFs. The information provided by these POFs can be very useful to RF circuit designers or design automation tools. Several experiments were performed in a 0.35- $\mu\text{m}$  CMOS technology, although the methodology presented here is valid for any technology process.

### 10.5.1 *L* versus *Q* Trade-off

One of the more interesting trade-offs to study when designing inductors is the achievable inductance and quality factor at a certain operating frequency within a restricted area. Therefore, this experiment considers maximization of both inductance and quality factor at a certain operating frequency, with a constraint in the maximum area allowed. The optimization is performed for an operating frequency  $f_o = 2.5$  GHz, whereas the inductor area is limited to  $200 \mu\text{m} \times 200 \mu\text{m}$ . The three symmetric topologies with differential excitation are considered, where the ranges of variation of the design variables are as follows:  $1 \leq N \leq 10$ ,  $10 \mu\text{m} \leq D_{IN} \leq 190 \mu\text{m}$ ,  $5 \mu\text{m} \leq W \leq 100 \mu\text{m}$ , and the spacing has been fixed at  $2.5 \mu\text{m}$ , as no improvement is expected from a larger spacing. The upper bound of the inner diameter is motivated by the area limitation.

Additional constraints are imposed to ensure that the inductance is sufficiently flat from DC to slightly above the operating frequency and that the self-resonance frequency is sufficiently beyond this frequency. As mentioned before, in order to keep the number of simulation points to a minimum value, a sampling strategy is planned that allows to verify whether the constraints are met without simulating at a large number of frequency points. This is extremely important as determining the *SRF* could imply simulating a large number of frequency points for each inductor, and therefore, long computation times needed. Instead, the alternative approach in Fig. 10.9 is proposed. To ensure that the inductance is flat at the operating frequency, it is measured at four frequency values: 100 kHz,  $f_o = 2.5$  GHz and two additional frequencies, one slightly above  $f_o$  and one slightly below  $f_o$ . Then, the differences between the value of the inductance at  $f_o$  and at 100 kHz are constrained to be smaller than 5 %, and the differences between  $L(f_o + \Delta f)$  and  $L(f_o - \Delta f)$  must be smaller than 1 %. On the other hand, if we want to ensure that the *SRF* lies sufficiently above  $f_o$ , we only have to ensure that the value of  $Q$  is near its maximum at  $f_o$  and always with a positive slope around  $f_o$ , i.e.,  $Q(f_o) < Q(f_o + \Delta f)$ . As it

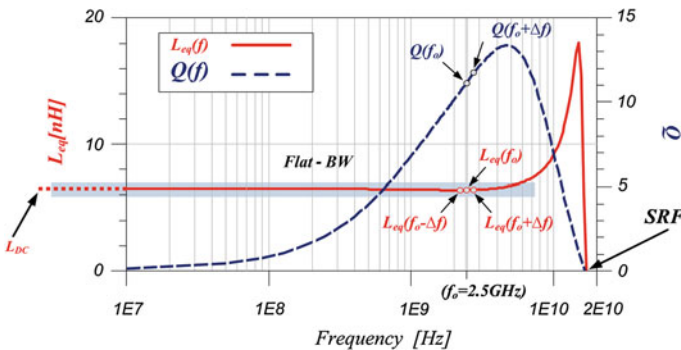
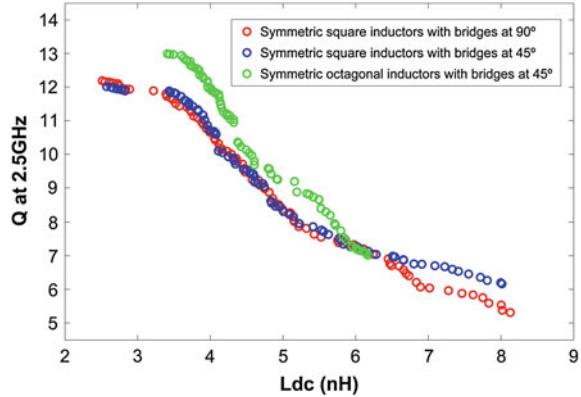


Fig. 10.9 Illustrating the sampling strategy for the optimization of *L* versus *Q*



**Fig. 10.10** Trade-offs for the three symmetric topologies



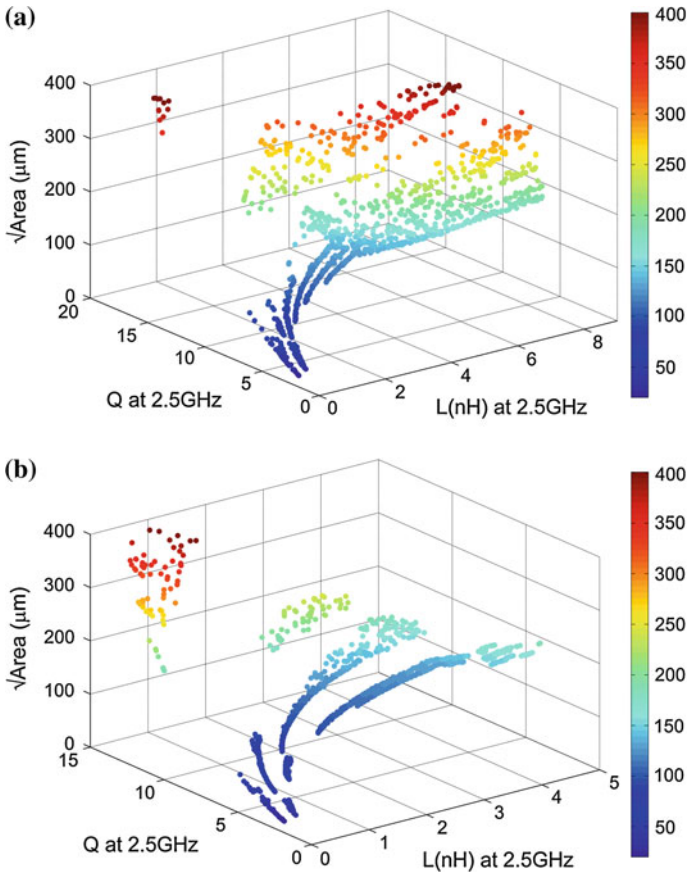
can be seen, only four frequency points are required to correctly determine the value of all the constraints.

Figure 10.10 shows the results for the three symmetric topologies. When comparing square topologies, it can be observed how the trade-off obtained is almost identical up to values slightly above 6 nH, where the reduced overlapping of 45° bridges results in higher  $Q$  values for the same inductance value. On the other hand, octagonal topologies, as expected, show higher  $Q$  values compared to square topologies; however, square topologies achieve higher inductance values within a given area.

### 10.5.2 $L$ versus $Q$ versus Area Trade-off

The most important parameters when inductors are included in RF circuits are inductance, quality factor, and area. Therefore, the aim of this experiment is to obtain a full set of inductors that model the best trade-offs between these performances. The optimization algorithm, NSGA-II, is configured to maximize the inductance and quality factor at  $f_o = 2.5$  GHz, whereas the inductor area is minimized and limited to a maximum value of  $400 \mu\text{m} \times 400 \mu\text{m}$ . The ranges of variation of the design variables are as follows:  $1 \leq N \leq 10$ ,  $10 \mu\text{m} \leq D_{\text{IN}} \leq 390 \mu\text{m}$ ,  $5 \mu\text{m} \leq W \leq 100 \mu\text{m}$ , and the spacing has been fixed at  $2.5 \mu\text{m}$ . Additionally, constraints imposed in the previous section on  $L$  and  $Q$ , and illustrated in Fig. 10.9, are also included here.

The experiments are performed for octagonal inductor topologies, for both symmetric and asymmetric inductors. The excitation mode is different for each topology: differential excitation for the former topology (symmetric) and non-differential for the latter (asymmetric). In the case of the asymmetric inductor, the fact that the input impedance is different at each port of the device due to its lack of symmetry was taken into account in the methodology. For this reason, the quality



**Fig. 10.11** Inductance versus quality factor versus area trade-offs for **a** symmetric octagonal with bridges at  $45^\circ$  and **b** asymmetric octagonal inductors

factor for each port was obtained and the best value was selected to guide the optimization flow. Results are shown in Fig. 10.11.

The POFs obtained provide valuable information to circuit designers because they bring in accurate information of the price to pay if one of the inductor performances is to be improved. For example, for a given inductance, we can know the possible values for the quality factor that can be obtained and the silicon area occupied by each one of them. It is important to remark that the inductor POF is actually a library of completely sized and accurately characterized inductors with optimum performance trade-offs that can be used for circuit-level sizing. A total of 1250 symmetric inductors and 1150 asymmetric inductors are obtained in these examples. The time needed to generate these fronts is about 2 weeks on a Linux node with 2 AMD64 processors and 6 cores at 2.6 GHz. This computational time is not a critical aspect as it must be run only once and the POF is useful for any performance specifications of the circuits in which they will be included.

## 10.6 Conclusions

This chapter has presented a methodology for the generation of performance fronts exhibiting the best trade-offs of integrated inductors which provide a set of accurately simulated optimal inductors. The generated Pareto-optimal fronts enable RF IC designers to explore the design space as well as select those inductors that better fit their requirements. A new model of use is enabled in automated RF IC design methodologies in which Pareto-optimal performance fronts of inductors are used at higher hierarchical levels, e.g., in multi-objective bottom-up design methodologies [30]. This model of use requires a highly accurate performance evaluation, which is provided by full EM simulation. The unavoidable penalty in computation time is alleviated by the off-line generation. Moreover, several techniques have been introduced that decrease the computational effort very significantly.

**Acknowledgments** This work has been partially supported by the TEC2013-45638-C3-3-R, TEC2010-14825, TEC2013-40430-R, and TEC2010-21484 projects, funded by the Spanish Ministry of Economy and Competitiveness and ERDF, by the P12-TIC-1481 project, funded by Junta de Andalucía, and by CSIC project PIE 201350E058.

## References

1. Choi, K., Allstot, D.J.: Parasitic-aware design and optimization of a CMOS RF power amplifier. *IEEE Trans. Circuits Syst. I Regul. Pap.* **53**(1), 16–25 (2006). doi:[10.1109/tcsi.2005.854608](https://doi.org/10.1109/tcsi.2005.854608)
2. De Ranter, C.R.C., Van der Plas, G., Steyaert, M.S.J., Gielen, G.G.E., Sansen, W.M.C.: CYCLONE: automated design and layout of RF LC-oscillators. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **21**(10), 1161–1170 (2002). doi:[10.1109/tcad.2002.802267](https://doi.org/10.1109/tcad.2002.802267)
3. Hershenson, M.D.M., Hajimiri, A., Mohan, S.S., Boyd, S.P., Lee, T.H.: Design and optimization of LC oscillators. In: 1999 IEEE/ACM International Conference on Computer-Aided Design, pp. 65–69, 7–11 Nov 1999 (1999)
4. Nieuwoudt, A., Ragheb, T., Massoud, Y.: Narrow-band low-noise amplifier synthesis for high-performance system-on-chip design. *Microelectron. J.* **38**(12), 1123–1134 (2007)
5. Pereira, P., Fino, H., Ventim-Neves, M.: LC-VCO design methodology based on evolutionary algorithms. In: International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), pp. 189–192, 19–21 Sept 2012 (2012)
6. Ramos, J., Francken, K., Gielen, G.G.E., Steyaert, M.S.J.: An efficient, fully parasitic-aware power amplifier design optimization tool. *IEEE Trans. Circuits Syst. I Regul. Pap.* **52**(8), 1526–1534 (2005). doi:[10.1109/tcsi.2005.851677](https://doi.org/10.1109/tcsi.2005.851677)
7. Ballicchia, M., Orcioni, S.: Design and modeling of optimum quality spiral inductors with regularization and Debye approximation. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **29**(11), 1669–1681 (2010)
8. Hershenson, M.D.M., Mohan, S.S., Boyd, S.P., Lee, T.H.: Optimization of inductor circuits via geometric programming. In: Proceedings of 36th Design Automation Conference, pp. 994–998 (1999)

9. Liu, B., Zhao, D., Reynaert, P., Gielen, G.E.: Synthesis of integrated passive components for high-frequency RF ICs based on evolutionary computation and machine learning techniques. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **30**(10), 1458–1468 (2011)
10. Mandal, S.K., Sural, S., Patra, A.: ANN- and PSO-based synthesis of on-chip spiral inductors for RF ICs. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **27**(1), 188–192 (2008)
11. Nieuwoudt, A., Massoud, Y.: Variability-aware multilevel integrated spiral inductor synthesis. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **25**(12), 2613–2625 (2006)
12. Niknejad, A.M., Meyer, R.G.: Analysis, design, and optimization of spiral inductors and transformers for Si RF ICs. *IEEE J. Solid-State Circuits* **33**(10), 1470–1481 (1998)
13. Pereira, P., Helena Fino, M., Coito, F., Ventim-Neves, M.: RF integrated inductor modeling and its application to optimization-based design. *Analog Integr. Circ. Sig. Process* **73**(1), 47–55 (2012). doi:[10.1007/s10470-011-9682-x](https://doi.org/10.1007/s10470-011-9682-x)
14. Talwalkar, N.A., Yue, C.P., Wong, S.S.: Analysis and synthesis of on-chip spiral inductors. *IEEE Trans. Electron Dev.* **52**(2), 176–182 (2005)
15. Yu, W., Bandler, J.W.: Optimization of spiral inductor on silicon using space mapping. In: *IEEE MTT-S International Microwave Symposium Digest*, pp. 1085–1088, 11–16 June 2006
16. Zhan, Y., Sapatnekar, S.S.: Optimization of integrated spiral inductors using sequential quadratic programming. In: *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, pp. 622–627, 16–20 Feb 2004
17. Jeong, S., Obayashi, S.: Efficient global optimization (EGO) for multi-objective problem and data mining. In: *IEEE Congress on Evolutionary Computation*, pp. 2138–2145, 2–5 Sept 2005
18. González-Echevarría, R., Castro-López, R., Roca, E., Fernández, F.V., Sieiro, J., Vidal, N., López-Villegas, J.M.: Automated generation of the optimal performance trade-offs of integrated inductors. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **33**(8), 1269–1273 (2014). doi:[10.1109/TCAD.2014.2316092](https://doi.org/10.1109/TCAD.2014.2316092)
19. Fiorelli, R., Peralias, E.J., Silveira, F.: LC-VCO design optimization methodology based on the  $g_m/I_D$  ratio for nanometer CMOS technologies. *IEEE Trans. Microw. Theory Tech.* **59**(7), 1822–1831 (2011). doi:[10.1109/tmtt.2011.2132735](https://doi.org/10.1109/tmtt.2011.2132735)
20. Tulunay, G., Balkir, S.: A synthesis tool for CMOS RF low-noise amplifiers. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **27**(5), 977–982 (2008)
21. Zhang, G., Dengi, A., Carley, L.R.: Automatic synthesis of a 2.1 GHz SiGe low noise amplifier. In: *IEEE Radio Frequency Integrated Circuits (RFIC) Symposium*, pp. 125–128 (2002)
22. Mohan, S.S., Hershenson, M.D.M., Boyd, S.P., Lee, T.H.: Simple accurate expressions for planar spiral inductances. *IEEE J. Solid-State Circuits* **34**(10), 1419–1424 (1999)
23. Kamon, M., Tsuk, M.J., White, J.K.: FASTHENRY: a multipole-accelerated 3-D inductance extraction program. *IEEE Trans. Microw. Theory Tech.* **42**(9), 1750–1758 (1994). doi:[10.1109/22.310584](https://doi.org/10.1109/22.310584)
24. Carrasco, T., Sieiro, J., López-Villegas, J.M., Vidal, N., González-Echevarría, R., Roca, E.: Mixed-mode impedance and reflection coefficient of two-port devices. *Prog. Electromagnet. Res.* **130**, 411–428 (2012). doi:[10.2528/PIER12052906](https://doi.org/10.2528/PIER12052906)
25. Bockelman, D.E., Eisenstadt, W.R.: Combined differential and common-mode scattering parameters: theory and simulation. *IEEE Trans. Microw. Theory Tech.* **43**(7), 1530–1539 (1995). doi:[10.1109/22.392911](https://doi.org/10.1109/22.392911)
26. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
27. Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, New York (2001)

28. Fernández, F.V., Esteban-Muller, J., Roca, E., Castro-López, R.: Stopping criteria in evolutionary algorithms for multi-objective performance optimization of integrated inductors. In: IEEE Congress on Evolutionary Computation (CEC), pp. 1–8, 18–23 July 2010
29. Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary computation and convergence to a Pareto front. In: Late breaking papers at the genetic programming 1998 conference, pp. 221–228 (1998)
30. Sanchez-Lopez, C., Castro-Lopez, R., Roca, E., Fernandez, F.V., Gonzalez-Echevarria, R., Esteban-Muller, J., Lopez-Villegas, J.M., Sieiro, J., Vidal, N.: A bottom-up approach to the systematic design of LNAs using evolutionary optimization. In: 2010 XIth International Workshop on Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD), pp. 1–5, 4–6 Oct 2010

# Chapter 11

## RF IC Performance Optimization by Synthesizing Optimum Inductors

Mladen Božanić and Saurabh Sinha

**Abstract** The chapter reviews inductor theory and describes various integrated inductor options. It also explains why integrated planar spiral inductors are so useful when it comes to integrated RF circuits. Furthermore, the chapter discusses the theory of spiral inductor design, inductor modeling, and how this theory can be used in inductor synthesis. In the central part of the chapter, the authors present a methodology for synthesis of planar spiral inductors, where numerous geometries are searched through in order to fit various initial conditions.

### 11.1 Introduction

With technology scaling, it has become possible to integrate an ever-increasing number of devices into the same integrated circuit (IC), thus making systems on chip more compact and affordable. Specific integrated radio-frequency (RF) circuits, particularly transmitters, are often power hungry, and therefore, it is paramount to design these circuits so that they operate at the maximum attainable efficiency to conserve battery power and reduce heat emissions. Suboptimal design is still one of the major problems in ICs. Even with optimal system design and

---

M. Božanić

Department of Electrical and Electronic Engineering Science, Faculty of Engineering and the Built Environment, University of Johannesburg, Auckland Park Kingsway Campus, Johannesburg 2006, South Africa  
e-mail: mbozanic@ieee.org

S. Sinha (✉)

Faculty of Engineering and the Built Environment, University of Johannesburg, Auckland Park Kingsway Campus, Johannesburg 2006, South Africa  
e-mail: ssinha@ieee.org

careful choice of topology for the particular application, large amounts of energy are often wasted due to low-quality passives, especially inductors.

Inductors have traditionally been difficult to integrate due to their inherent low-quality factors and modeling complexity. Furthermore, although many different inductor configurations are available for an RF designer to explore, support for integrated inductors in electronic design automation (EDA) tools and process design kits has been very limited in the past. Some vendors provide a library of several qualified integrated inductors for each RF-capable process. Each of these inductors operates at its peak efficiency only at a certain frequency, making the library impractical for many applications. Other vendors provide p-cells of spiral inductors, and although technology parameters are taken into account to calculate the resulting quality factors for a specific frequency, there is still a distinct lack of technology-aware optimization. It is more practical, yet tedious, to use such p-cells, owing to the cut-and-try nature of this approach to inductor selection and the lack of automated design flow. In this chapter, a recent advance in technology-aware integrated inductor design is presented, where designers are supported by an equation-based inductor synthesis algorithm. The computation technique aims to allow RF designers to optimize integrated inductors, given the inductor center frequency dictated by the device application, and the geometry constraints. This does not only lay down a foundation for system-level RF circuit performance optimization, but, because inductors are often the largest parts of an RF system, it also allows for optimal usage of chip real estate.

The chapter first introduces inductor theory and describes various integrated inductor options. The second part of the chapter introduces the theory of spiral inductor design, inductor modeling, and ways in which this theory can be used in inductor synthesis. In the central part of the chapter, a methodology for design, computation, and optimization of planar spiral inductors is presented. The methodology provides for an intelligent search through inductor configurations fitting the initial choices. Based on the selected model, the algorithm will compute optimal inductors, together with inductance, quality factors, schematics, and layouts and intelligently select the configuration with the best performance. The algorithm is only introduced as an illustration of an inductor synthesis methodology, the theory of which can be expanded to any integrated inductor configuration.

## 11.2 Inductor Theory

A real inductor is usually modeled as an ideal inductor  $L_S$  in series with a resistor  $R_S$ , both in parallel with capacitor  $C_S$ , as shown in Fig. 11.1 [1]. Inclusion of the series resistor and parallel capacitor is necessary to model the losses of the inductor even at frequencies below RF because the quality or Q-factor is generally much lower for the inductor than for other passive components. The Q-factor of the inductor is defined as  $2\pi$  times the ratio of energy stored in the device and energy

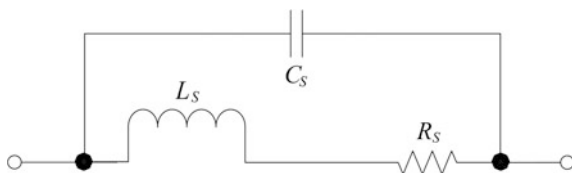


Fig. 11.1 General high-frequency model of an inductor [1]

lost in one oscillation cycle. If  $Z$  is defined as the impedance of an inductor, then the Q-factor is given by

$$Q = \frac{\text{Im}(Z)}{\text{Re}(Z)}. \tag{11.1}$$

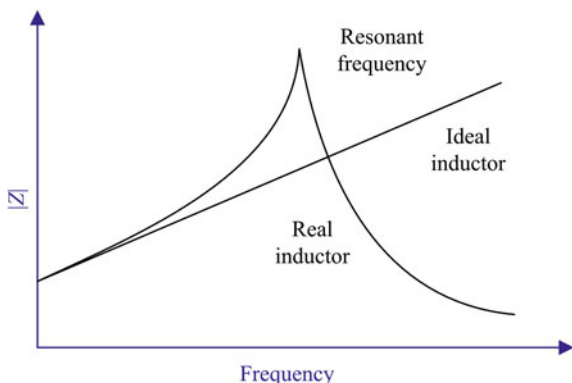
For the simple circuit in Fig. 11.1, (11.1) reduces to

$$Q = \frac{X}{R_S}, \tag{11.2}$$

where  $X$  is the total reactance of the inductor. The Q-factor is heavily dependent on the frequency and exhibits a peak  $Q_{\text{max}}$ .

While an ideal inductor exhibits a constant impedance slope value for all frequencies, every non-ideal inductor exhibits a slope dependent on frequency, as shown in Fig. 11.2. The frequency where magnitude of impedance ( $|Z|$ ) peaks is called the resonant frequency of an inductor. The resonant frequency,  $f_r = \frac{1}{2\pi\sqrt{L_S C_S}}$ , should ideally peak at infinity, but the finite value of the peak is due to the resistance  $R_S$ . Similarly, capacitance  $C_S$  is the reason the inductor exhibits capacitive instead of inductive behavior at frequencies above the resonance.

Fig. 11.2 Frequency response of the impedance of ideal and real inductors [1]





### ***11.2.1 Inductor Implementation Options***

As will become apparent later in this chapter, the geometry of choice for the topic of this chapter is the integrated planar spiral inductor topology. Various factors, such as inductor size and lower Q-factor of integrated passive inductors, often result in one of the following inductor alternative implementations:

- External inductors,
- Active integrated inductors,
- Microelectromechanical systems (MEMS) inductors,
- Bond wires, or
- Other on-chip or on-package/in-package implementations.

Each of the above possibilities is discussed in more detail in the sections that follow.

#### **11.2.1.1 External Inductors**

External or off-chip inductors are connected to a system outside of the IC package. They are usually implemented as a solenoidal coil or a toroid. Their usage at high frequencies also implies careful printed-circuit board (PCB) modeling and design. Although high-quality inductors are widely available from suppliers, their inductance values are usually limited to standard values of 10 nH and higher. The frequency of the Q-factor peak (typically in the range of hundreds) is also predefined and is usually located in either the high-megahertz or the low-gigahertz range. Another drawback for integrated design is the fact that the value obtained upon PCB placement will differ from the rated value due to parasitics involving PCB tracks, IC bonding, and other factors.

#### **11.2.1.2 Integrated Active Inductors**

Integrated active inductors are a good alternative to their passive counterparts because of their higher Q-factor. Typical Q-factors that can be obtained for active configurations are between 10 and 100, which is up to ten times those of spiral inductors [2]. Active inductors can also take up a smaller area on the chip than spiral inductors. The main disadvantages of active inductors include increased power consumption, presence of electrical noise from active devices, and limited dynamic range. A design requiring only six transistors has been proposed in Ler et al. [3].

#### **11.2.1.3 MEMS Inductors**

MEMS is an IC fabrication technique that empowers conventional two-dimensional (2-D) circuits to expand into the third dimension (3-D) [4]. This principle becomes

particularly useful in inductor fabrication, because the influence of substrate parasitics on the Q-factor can be reduced significantly when silicon below the inductor is effectively replaced by air or another material that has lower relative permittivity. Typical obtainable Qs range from 10 to 30 for 1-nH inductors at multigigahertz frequencies. An example of a high-Q silicon-based inductor using polymer cavity can be found in Khoo et al. [5]. As an alternative to spiral MEMS inductors, solenoidal inductors suspended on chip can be used with various degrees of chip stability [6]. Several advantages over conventional spiral inductors can be identified, which include a lower stray capacitance due to the fact that only a part of the inductor is lying on the silicon substrate, a simple design equation and greater possibilities for flexible layout. Out-of-plane inductors [7] are similar to MEMS inductors, but their coils are fabricated using stress-engineered thin films. The stress gradient is induced by changing the ambient pressure during film deposition. When released, a stress-graded film curls up in a circular trajectory. The typical Q-factor of this configuration is over 70 at 1 GHz.

Although MEMS devices present an attractive alternative to conventional passive inductors, particularly because of the high Q-factors, their fabrication requires process changes or modifications to the wafer after fabrication. After these procedures, repeatability [8] is not assured.

#### 11.2.1.4 Bond Wires

Bond wires, which usually present a parasitic quantity for signals transmitted between systems inside and outside the packaged device, reflect inductive behavior [9] which can be used as an advantage in RF design. Electrical characteristics of bond wires depend on the material of which they are made and their cross section, the height above the die plane, the horizontal length, and the pitch between the adjacent wires [10]. Many of these characteristics are dependent on pad location and type of package, but if these parameters are known in advance of design, bond-wire models can be used accurately to determine bond-wire Q-factor and inductance. Although bond wires with Q-factors of 50 have been reported, their inductances will typically be less than 1 nH [10]. This limits their feasibility for gigahertz range where well-controlled inductances of 1 nH and more are often needed.

#### 11.2.1.5 Other On-Chip Implementations

Masu et al. [11] discusses two types of inductor not commonly found in the literature. The first type of inductor is a meander inductor. It is a flat passive inductor consisting of a long piece of metal that is not wound as in the case of the spiral inductor which will be described in detail later, but meanders similarly to rivers in their lower watercourses. This inductor occupies a small area and no underpass is needed, but its measured Q-factor is quite low (about 2.1 for

inductance of 1.3 nH). Such trade-off between the area and Q-factor is acceptable for matching network applications. The second type of inductor is a snake inductor that meanders into the third dimension.

Vroubel et al. [12] discusses electrically tunable solenoidal on-chip inductors. Other tunable inductors are commonly seen as implemented in active configuration, such as in the case of the inductor in Seo et al. [13].

Toroid inductors can also be implemented on chip by means of micromachining [14].

## 11.2.2 *Spiral Inductor Theory*

Although inductor implementations described in the previous section are widely used due to their advantages over passive integrated inductors, they are normally too complex to implement, due to process changes and post-fabrication requirements, which in turn increase total RF device manufacturing cost. Spiral integrated inductors present a viable option for practical RF implementations when designed with the aid of the inductor optimization technique described in this chapter. This is due to the deterministic models that can be used to accurately predict the inductance value and Q-factors of any inductive structure on chip, given the process parameters and geometry of that inductive structure.

### 11.2.2.1 **Common Spiral Inductor Geometries**

Several spiral inductor geometries are commonly used in RF circuits. These include square and circular inductors, as well as various polygons [15]. The square spiral has traditionally been more popular since some IC processes constrained all angles to 90° [16], but it generally has a lower Q-factor than the circular spiral, which most closely resembles the common off-chip solenoidal inductors but is difficult to layout. A polygon spiral is a compromise between the two. Drawings of square and circular inductors are shown in Fig. 11.3.

The geometries shown in Fig. 11.3 are asymmetric and require only a single metal layer for fabrication. Additional layers are only needed to bring the signal lines to the outside of an inductor and are universally known as underpasses. Symmetrical inductors are also possible, but they require more than one underpass, in this case known as metal-level interchange, shown in Fig. 11.4a [16]. Alternatively, the second metal layer can be used as part of the core of the inductors. An example of such multilayer geometry is a two-layer square inductor as shown in Fig. 11.4b [17]. The multilayer geometries can deliver higher quality factors than a single-layer inductor due to mutual inductance coupling of different spirals.

Another common geometry is a taper geometry, where inner spirals of inductors decrease in width in respect of the outer spirals [18] (Fig. 11.5). Tapering is done to

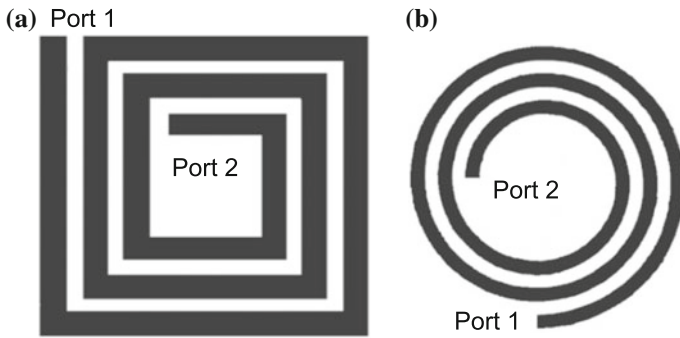


Fig. 11.3 The square (a) and circular (b) spiral inductors [16]

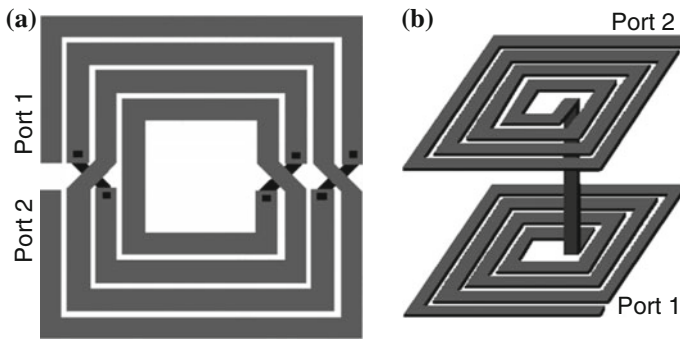
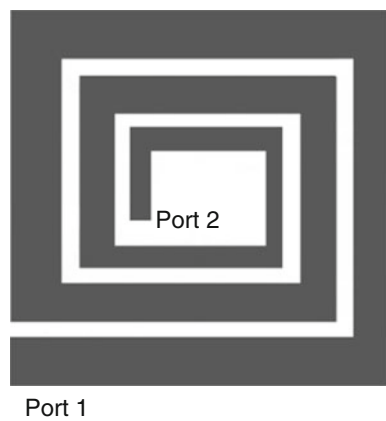


Fig. 11.4 The symmetrical [16] (a) and two-layer (b) [17] spiral inductor

Fig. 11.5 A taper spiral inductor [18]



suppress eddy current losses in the inner turns in order to increase the Q-factor, but it is most effective when substrate losses are negligible.

### 11.2.2.2 Spiral Inductor Geometry Parameters

For a given geometry, a spiral inductor is fully specified by the number of turns ( $n$ ), the turn width ( $w$ ), and two of the following: inner, outer, or average diameter ( $d_{in}$ ,  $d_{out}$  or  $d_{avg} = (d_{in} + d_{out})/2$ ), as shown in Fig. 11.6 for the square and circular inductors. Spacing between turns,  $s$ , can be calculated from other geometry parameters. Another geometry parameter commonly used in equations is the fill ratio, defined as

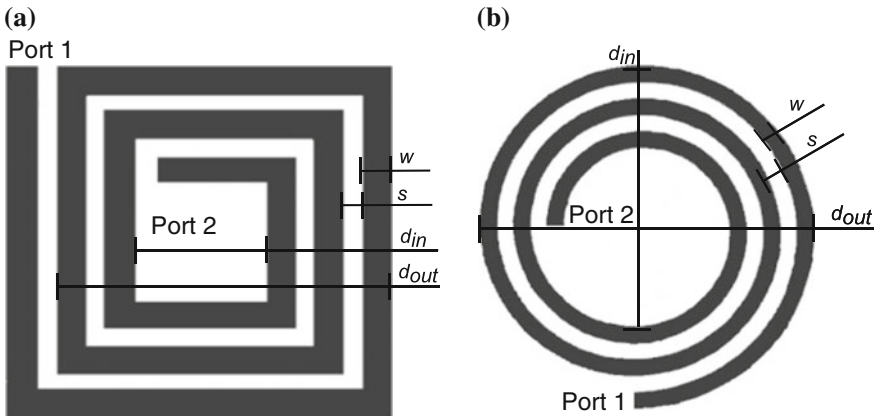
$$\rho_{fill} = \frac{d_{out} - d_{in}}{d_{out} + d_{in}}. \quad (11.3)$$

The total length of a spiral is also important for calculations. It is dependent on inductor geometry. For a square inductor, it can be calculated as

$$l = 4(d_{in} + w) + 2n(2n - 1)(s + w). \quad (11.4)$$

### 11.2.2.3 Spiral Inductor Models

Several spiral inductor models are widely used, depending on the required modeling complexity. In this section, single- $\pi$ , segmented, double- $\pi$ , and third-order models will be described.



**Fig. 11.6** Geometry parameters of the **a** square and **b** circular spiral inductors

**Single- $\pi$  Model** The most commonly used model is a lumped single- $\pi$  nine-component configuration as shown in Fig. 11.7 [15, 19]. In this model,  $L_S$  is the inductance at the given frequency,  $R_S$  is the parasitic resistance, and  $C_S$  is the parasitic capacitance of the spiral inductor structure.  $C_{ox}$  is the parasitic capacitance due to oxide layers directly under the metal inductor spiral. Finally,  $C_{Si}$  and  $R_{Si}$  represent the parasitic resistance and capacitance due to the silicon substrate. This topology does not model the distributive capacitive effects, but it models correctly for parasitic effects of the metal spiral and the oxide below the spiral, as well as for substrate effects.

**Segmented Model** A somewhat more complicated model is the model presented by Koutsoyannopoulos and Papananos [20]. Each segment of the inductor is modeled separately with a circuit as shown in Fig. 11.8. In this model, parasitics  $C_{ox}$ ,  $C_{Si}$ , and  $R_{Si}$  represent parasitics of only one inductor segment,  $L_S$  and  $R_S$  represent inductance and parasitic capacitance of one segment coupled to all segments, while capacitances  $C_{f1}$  and  $C_{f2}$  are added to represent coupling to adjacent segment nodes.

**Double- $\pi$  Distributed Model** The standard single- $\pi$  model can also be extended into a second-order, distributed double- $\pi$  model as shown in Fig. 11.9 [19, 21]. A second-order ladder (with third grounded branch) is used to model the distributive characteristics of metal windings. The interwinding capacitance ( $C_w$ ) is included to model the capacitive effects between metal windings of the inductor. The transformer loops ( $M_{S1}$  and  $M_{S2}$ ) represent the effects of frequency-dependent series loss.

**Third-Order Transmission-Line Model** The second-order model shown in Fig. 11.9 is valid for the inductor up to the first resonance frequency. If a third-order model is used, it is possible to predict inductor behavior accurately, even beyond the resonant frequency. One such model is presented by Lee et al. [22]. An equivalent

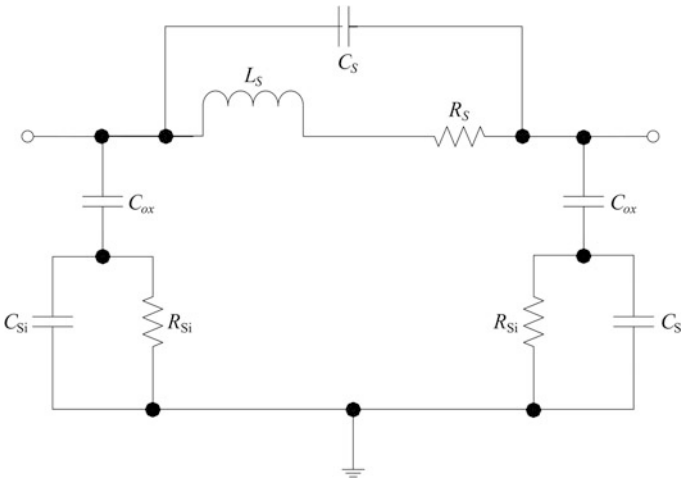


Fig. 11.7 A commonly used nine-component spiral inductor model [15]

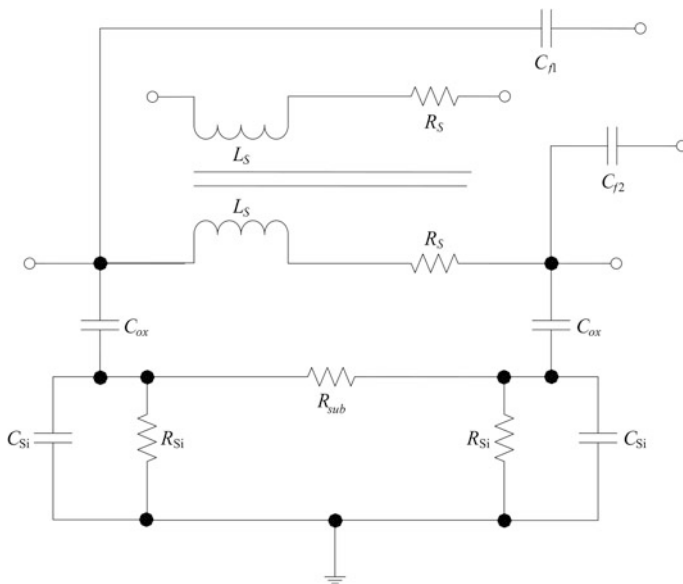


Fig. 11.8 An equivalent two-port model for one segment of a spiral inductor [20]

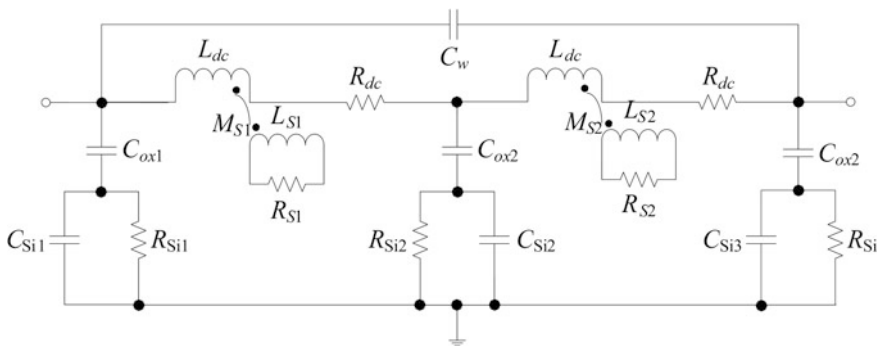


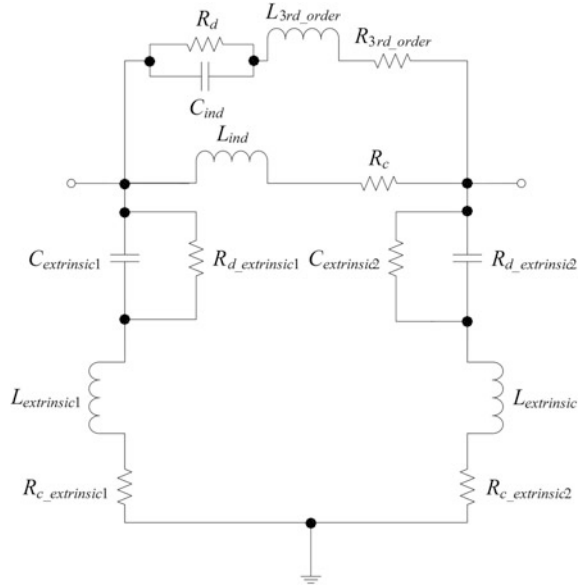
Fig. 11.9 A double- $\pi$  distributed inductor model [21]

circuit diagram for this configuration is shown in Fig. 11.10. Extrinsic admittances are used, and all circuit components are self-explanatory from this figure.

### 11.2.2.4 Computation of Series Inductance and Parasitics for Single- $\pi$ Model

The single- $\pi$  inductor model of Fig. 11.7 is sufficient to model spiral inductors accurately for frequencies below resonance [23]. This model can be used as proof of

**Fig. 11.10** A complete third-order inductor model [22]



concept when developing a routine for spiral inductor design and optimization. In sections that follow, series inductance  $L_S$ , as well as parasitic capacitances and resistances shown in this figure, together with their influence of inductor performance, is described and explained.

**Series Inductance ( $L_S$ )** Various equations are commonly used in the literature to represent the series inductance of spiral inductors with various levels of accuracy.

The modified Wheeler equation is based on the equation derived by Wheeler in 1928 [15]:

$$L_{mw} = K_1 \mu \frac{n^2 d_{avg}}{1 + K_2 \rho_{fill}}, \tag{11.5}$$

where  $K_1$  and  $K_2$  are geometry-dependent coefficients with values defined in Table 11.1 and  $\mu$  is magnetic permeability of the metal layer.

Another expression can be obtained by approximating the sides of the spiral by symmetrical current sheets of equivalent current densities as described in [15]:

**Table 11.1** Coefficients for the modified Wheeler expression [15]

Layout	$K_1$	$K_2$
Square	2.34	2.75
Octagonal	2.33	3.82
Hexagonal	2.25	3.55



**Table 11.2** Coefficients for the current sheet expression [15]

Layout	$c_1$	$c_2$	$c_3$	$c_4$
Square	1.27	2.07	0.18	0.13
Hexagonal	1.09	2.23	0.00	0.17
Octagonal	1.07	2.29	0.00	0.19
Circular	1.00	2.46	0.00	0.20

$$L_{\text{gmd}} = \mu \frac{n^2 d_{\text{avg}} c_1}{2} \left[ \ln \frac{c_2}{\rho} + c_3 \rho_{\text{fill}} + c_4 \rho_{\text{fill}}^2 \right]. \quad (11.6)$$

where  $c_1$ ,  $c_2$ ,  $c_3$ , and  $c_4$  are geometry-dependent coefficients with values defined in Table 11.2. This expression exhibits a maximum error of 8 % for  $s \leq 3w$ .

Bryan's equation is another popular expression for the square spiral inductance [24]:

$$L = 0.00241 \left( \frac{d_{\text{out}} + d_{\text{in}}}{4} \right) n^5 \ln \left( \frac{4}{\rho_{\text{fill}}} \right). \quad (11.7)$$

The data-fitted monomial expression results in an error smaller than seen in the expressions given above (typically less than 3 %). It is based on a data-fitting technique. Inductance in nanohenries (nH) is calculated as follows [15, 24]:

$$L_{\text{mon}} = \beta d_{\text{out}}^{\alpha_1} w^{\alpha_2} d_{\text{avg}}^{\alpha_3} n^{\alpha_4} s^{\alpha_5}, \quad (11.8)$$

where coefficients  $\beta$ ,  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$ ,  $\alpha_4$ , and  $\alpha_5$  are once again geometry dependent, as presented in Table 11.3.

The monomial expression has been developed by curve fitting over a family of 19,000 inductors [15]. It has better accuracy and higher simplicity than the equations described above and is the equation of choice.

**Parasitic Resistance ( $R_s$ )** Parasitic resistance is dependent on the frequency of operation. At DC, this value is mostly determined by the sheet resistance of the material of which the wire is made. At high frequencies, this is surpassed by the resistance that arises due to the formation of eddy currents. It is governed by the resistivity of the metal layer in which the inductor is laid out ( $\rho$ ), the total length of all inductor segments ( $l$ ), the width of the inductor ( $w$ ), and its effective thickness ( $t_{\text{eff}}$ ) [25]:

**Table 11.3** Coefficients for the spiral inductor inductance calculation [15]

Layout	$\beta$	$\alpha_1 (d_{\text{out}})$	$\alpha_2 (w)$	$\alpha_3 (d_{\text{avg}})$	$\alpha_4 (n)$	$\alpha_5 (s)$
Square	$1.62 \times 10^{-3}$	-1.21	-0.147	2.40	1.78	-0.030
Hexagonal	$1.28 \times 10^{-3}$	-1.24	-0.174	2.47	1.77	-0.049
Octagonal	$1.33 \times 10^{-3}$	-1.21	-0.163	2.43	1.75	-0.049

$$R_S = \frac{\rho l}{wt_{\text{eff}}}. \quad (11.9)$$

Effective thickness,  $t_{\text{eff}}$ , is dependent on the actual thickness of the metal layer,  $t$ :

$$t_{\text{eff}} = \delta(1 - e^{-t/\delta}), \quad (11.10)$$

where  $\delta$  is skin depth related to frequency  $f$  via relation

$$\delta = \sqrt{\frac{\rho}{\pi \mu f}}. \quad (11.11)$$

**Parasitic Capacitance ( $C_S$ )** Parasitic capacitance is the sum of all overlap capacitances created between the spiral and the underpass. If there is only one underpass and it has the same width as the spiral, then the capacitance is equal to [25]

$$C_S = nw^2 \frac{\epsilon_{\text{ox}}}{t_{\text{ox}M1-M2}}, \quad (11.12)$$

where  $t_{\text{ox}M1-M2}$  is the oxide thickness between the spiral and the underpass and  $\epsilon_{\text{ox}}$  is the dielectric constant of the oxide layer between the two metals.

**Oxide and Substrate Parasitics ( $C_{\text{ox}}$ ,  $C_{\text{Si}}$ , and  $R_{\text{Si}}$ )** The oxide and substrate parasitics are approximately proportional to the area of the inductor spiral ( $l \cdot w$ ), but are also highly dependent on the conductivity of the substrate and the operating frequency. In order to calculate the oxide capacitance  $C_{\text{ox}}$  and substrate capacitance  $C_{\text{Si}}$ , the effective thickness ( $t_{\text{eff}}$ ) and effective dielectric constant ( $\epsilon_{\text{eff}}$ ) of either oxide or substrate must be determined. The effective thickness is calculated as follows [26]:

$$t_{\text{eff}} = w \left[ \frac{w}{t} + 2.42 - 0.44 \frac{t}{w} + \left( 1 - \frac{t}{w} \right)^6 \right]^{-1}, \quad \text{for } \frac{t}{w} \leq 1, \quad (11.13)$$

or

$$t_{\text{eff}} = \frac{w}{2\pi} \ln \left( \frac{8t}{w} + \frac{4w}{t} \right), \quad \text{for } \frac{t}{w} \geq 1 \quad (11.14)$$

for both oxide and substrate. The effective dielectric constant is determined as follows:

$$\epsilon_{\text{eff}} = \frac{1 + \epsilon}{2} + \frac{\epsilon - 1}{2} \left( 1 + \frac{10t}{w} \right)^{-\frac{1}{2}}. \quad (11.15)$$

Then,

$$C_{\text{ox}} = \frac{wl\varepsilon_0\varepsilon_{\text{eff ox}}}{t_{\text{eff ox}}} \quad (11.16)$$

and

$$C_{\text{Si}} = \frac{wl\varepsilon_0\varepsilon_{\text{eff Si}}}{t_{\text{eff ox Si}}}. \quad (11.17)$$

In addition to the effective thickness ( $t_{\text{eff}}$ ) given in (11.14), to calculate  $R_{\text{Si}}$ , the effective conductivity ( $\sigma_{\text{eff}}$ ) of the substrate is needed. The effective conductivity can be obtained from

$$\sigma_{\text{eff}} = \sigma \left[ \frac{1}{2} + \frac{1}{2} \left( 1 + \frac{10t}{w} \right)^{-\frac{1}{2}} \right], \quad (11.18)$$

where  $\sigma = \frac{1}{\rho}$  represents the substrate conductivity.

Therefore,

$$R_{\text{Si}} = \frac{t_{\text{eff Si}}}{\sigma_{\text{eff}}wl}. \quad (11.19)$$

### 11.2.2.5 Quality Factor and Resonance Frequency for Single- $\pi$ Model

As discussed at the beginning of the chapter, the quality factor is the measure of performance of any inductor. For the single- $\pi$  model, if  $R_P$  and  $C_P$  are defined as

$$R_P = \frac{1}{\omega^2 C_{\text{ox}}^2 R_{\text{Si}}} + \frac{R_{\text{Si}}(C_{\text{ox}} + C_{\text{Si}})^2}{C_{\text{ox}}^2} \quad (11.20)$$

and

$$C_P = C_{\text{ox}} \cdot \frac{1 + \omega^2(C_{\text{ox}} + C_{\text{Si}})C_{\text{Si}}R_{\text{Si}}^2}{1 + \omega^2(C_{\text{ox}} + C_{\text{Si}})^2 R_{\text{Si}}^2}, \quad (11.21)$$

then the Q-factor can be calculated as [27]

$$Q = \frac{\omega L_S}{R_S} \cdot \frac{R_P}{R_P + \left[ \left( \frac{\omega L_S}{R_S} \right)^2 + 1 \right] R_S} \cdot \left[ 1 - (C_P + C_S) \cdot \left( \omega^2 L_S + \frac{R_S^2}{L_S} \right) \right], \quad (11.22)$$

where  $\omega = 2\pi f$ . Three different factors can be isolated in (11.20) [28]. The first factor,  $F_1 = \omega L_S/R_S$ , is the intrinsic (nominal) Q-factor of the overall inductance. The second factor,  $F_2 = \frac{R_p}{R_p + [(\omega L_S/R_S)^2 + 1]R_S}$ , models the substrate loss in the semiconducting silicon substrate. The last factor,  $F_3 = 1 - (C_P + C_S) \cdot (\omega^2 L_S + R_S^2/L_S)$ , models the self-resonance loss due to total capacitance  $C_P + C_S$ . This resonant frequency can be isolated by equating the last factor to zero, and solving for  $\omega$ . This results in the formula for self-resonance frequency of the spiral inductor:

$$f_r = \frac{\omega_o}{2\pi} = \frac{1}{2\pi} \sqrt{\frac{1}{L_S \cdot (C_P + C_S)} - \left(\frac{R_S}{L_S}\right)^2}. \quad (11.23)$$

At low frequencies, the loss of metal line ( $F_1$ ) restricts the performance of inductors [29]. In high-frequency ranges, the loss of substrate ( $F_2$ ) prevails as the restricting factor.  $F_2$  is greatly dependent on the conductivity of the substrate. As conductivity increases at a fixed frequency, the skin depth of the substrate also increases, leading to an increase of eddy currents in the substrate resulting in a decrease of the Q-factor of the inductor. Heavily doped substrates are usually used in a submicron process, with substrate resistivity usually lying in the range of 10–30  $\Omega$  cm. As a result, in the traditional (Bi)CMOS process, the performance of spiral inductors is limited by the substrate. Inductors laid out in MEMS processes, as mentioned earlier in this chapter, strive to minimize the effects of this limitation.

Figure 11.11 shows the analysis of factors  $F_1$ ,  $F_2$ , and  $F_3$  defined in (11.20) for 1 and 5-nH sample spiral inductors optimized at different frequencies for their highest quality operation. It can be observed that, although the nominal Q-factor ( $F_1$ ) increases with frequency,  $F_2$  and  $F_3$  decrease in the same range, resulting in the decrease of the overall Q-factor ( $Q$ ) at frequencies above 1 GHz.

Close to resonant frequency, the frequency has some effect on the apparent inductance value, which can be calculated from [30]

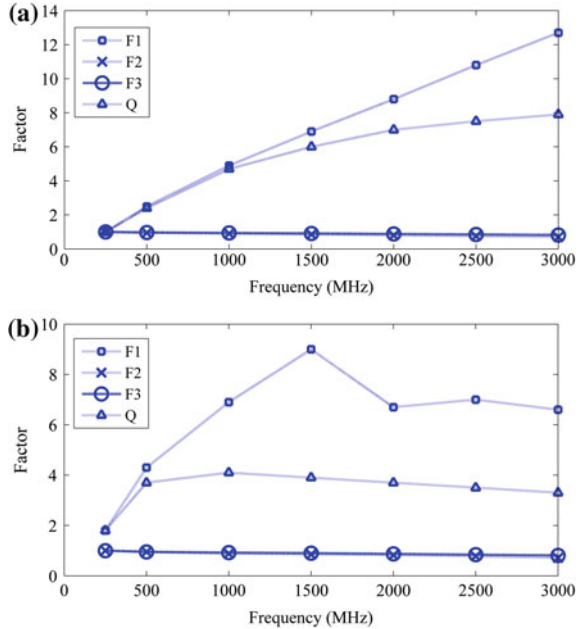
$$L_{\text{eff}} = \frac{\text{Im}(Z)}{2\pi f_r}, \quad (11.24)$$

where  $Z$  is the total impedance of the single- $\pi$ -modeled inductor with its one port grounded.

### 11.2.2.6 Current Approach to Spiral Inductor Design

When designing an integrated capacitor, a designer may simply increase or decrease the area of the component until the required capacitance is obtained. Although capacitance of the parallel plate capacitor does not solely depend on the area of its plates, but also on other factors such as fringing effects, a nearly linear relationship

**Fig. 11.11** Analysis of the determining factors of the Q-factor equation for **a** 1-nH inductor and **b** 5-nH inductor

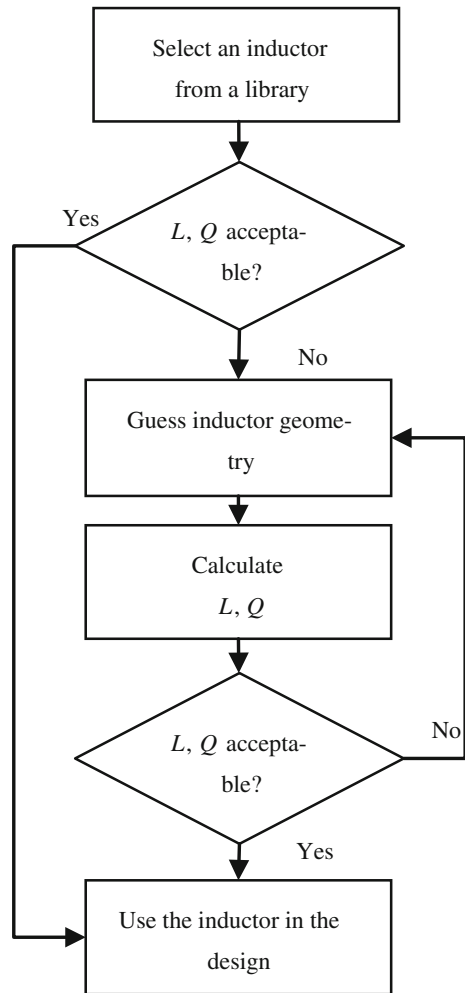


between the two is retained. A similar relationship between the length and total resistance holds for resistors. By modifying the length of a part of the process layer used for fabrication of the resistor, a designer can obtain the desired value of its resistance. However, this does not apply to the spiral inductors. Contrary to common sense, one cannot just simply increase the number of turns or the width of a single turn to change the inductance. The complicated inductance relationship given in (11.22) can illustrate this interdependency. This complexity of spiral inductor models is one of the reasons why various cut-and-try approaches are used in practice, such as the one illustrated by the flowchart in Fig. 11.12.

In this typical approach, designer chooses an inductor from a library if it contains one with acceptable inductance and Q-factor. Most likely this inductor will not be available, in which case he or she has to guess inductor geometry, then calculate its  $L$  and  $Q$ , decide on whether these parameters are acceptable, and if not, repeat the guessing process until a satisfactory inductor is found. This process, even if calculations are performed by means of software such as MATLAB<sup>1</sup> or inductors are simulated in electromagnetic (EM) simulation software, could take substantial amount of time.

<sup>1</sup>MATLAB is a technical computing language from MathWorks: <http://www.mathworks.com/>.

**Fig. 11.12** A flowchart of conventional spiral inductor design procedure



### 11.2.2.7 Guidelines for Integrating Spiral Inductors

As detailed in the introductory section of this chapter, although spiral inductors are a good choice for exclusively on-chip RF circuits, their implementation is not as straightforward. The inductors occupy large areas on the chip, suffer from low-quality factors, and are difficult to design for low tolerance. Hastings [31] isolates some general guidelines that can assist in increasing the quality of an inductor, irrespective of its geometry and its model. These guidelines were adhered to throughout this chapter:

1. Where possible, one should use the highest resistivity substrate available. This will reduce the eddy losses that reduce the Q-factor.
2. Inductors should be placed on the highest possible metal layers. In this way, substrate parasitics will have a less prominent role because the inductor will be further away from the silicon.
3. If necessary, parallel metal layers for the body of the inductor may be used to reduce the sheet resistance.
4. Unconnected metal should be placed at least five turn widths away from inductors. This is another technique that helps to reduce eddy current losses.
5. Excessively wide or narrow turn widths should be avoided. Narrow turns have high resistances, and wide turns are vulnerable to current crowding.
6. The narrowest possible spacing between the turns should be maintained. Narrow spacing enhances magnetic coupling between the turns, resulting in higher inductance and Q-factor values.
7. Filling the entire inductor with turns should be avoided. Inner turns are prone to the magnetic field, again resulting in eddy current losses.
8. Placing of unrelated metal plates above or under inductors should be avoided. Ungrounded metal plates will also aid the eddy currents to build up.
9. Placing of junctions beneath the inductor should be avoided. The presence of a junction close to the inductor can produce unwanted coupling of AC signals.
10. Short and narrow inductor leads should be used. The leads will inevitably produce parasitics of their own.

### 11.3 Method for Designing Spiral Inductors

In this section, an improvement to the common iterative procedure described previously is proposed. The proposed software routine can find an inductor close to the specified value, with the highest possible Q-factor, occupying a limited area, and using predetermined technology layers (synthesis of the inductor structure). For completeness and verification purposes, inductances and Q-factor values of various spiral inductors can be calculated if the geometry parameters of such inductors are given (analysis of inductor structure). Analysis and synthesis concepts are developed for the single-square spiral inductor of Fig. 11.6a, using equations for single- $\pi$  model, but they can be extended to other geometries and more complex models.

In the text that follows, input and output parameters of the routine are given, together with its flow.

#### 11.3.1 *Input Parameters*

Parameters for accurate inductor modeling can be divided into two groups: geometry and process parameters. Process parameters are related to the fabrication

process (technology) in which the IC is to be prototyped and the designer has very little, if any, control over them. Geometry parameters can be understood as user parameters because they are related to the specific application required by the designer. In addition, the frequency of operation of the inductor also needs to be known for applicable Q-factor calculation. When providing user parameters, general guidelines for the inductor design presented in the previous section need to be followed where possible.

The following subsections give a detailed description of the parameters needed for the spiral inductor design.

### 11.3.1.1 Geometry Parameters

For the analysis of an inductor structure, the following input geometry parameters are necessary:

- Outer diameter,  $d_{\text{out}}$  ( $\mu\text{m}$ );
- Inner diameter,  $d_{\text{in}}$  ( $\mu\text{m}$ );
- Turn width,  $w$  ( $\mu\text{m}$ ); and
- Number of turns,  $n$ .

For the synthesis of the inductor structure, only constraints for the geometry should be specified (all in micrometer):

- Minimum value of inner diameter,  $d_{\text{in (min)}}$ ;
- Maximum value of outer diameter,  $d_{\text{out (max)}}$ ;
- Minimum value for turn spacing,  $s_{\text{min}}$ ; and
- Minimum turn width,  $w_{\text{min}}$ .

Tolerance (in percentage) for the acceptable inductance values, as well as grid resolution (in micrometer), is also required for the synthesis part of the routine. For the inductor to pass design rule checks (DRC), design rules document provided by the foundry has to be consulted, with emphasis on maximum allowed grid resolution and minimum allowed metal spacings (including  $s_{\text{min}}$ ) for all used metal layers (both in  $\mu\text{m}$ ).

Table 11.4 summarizes the geometry input parameters.

### 11.3.1.2 Process (Technology) Parameters

The inductance value of a high-Q structure is predominantly determined by its geometry. However, the silicon substrate introduces process-dependent parasitics, which are dependent on the process parameters. They decrease the Q-factor and add shift to the inductance value. The following substrate parameters need to be specified:



**Table 11.4** Geometry parameters for the spiral inductor design

Parameter	Units	Geometry/inductance known
Outer diameter ( $d_{out}$ )	$\mu\text{m}$	Geometry
Inner diameter ( $d_{in}$ )	$\mu\text{m}$	Geometry
Turn width ( $w$ )	$\mu\text{m}$	Geometry
Number of turns ( $n$ )	–	Geometry
Minimum value of the inner diameter	$\mu\text{m}$	Inductance
Maximum value of the outer diameter	$\mu\text{m}$	Inductance
Minimum value for turn spacing ( $s$ )	$\mu\text{m}$	Inductance
Minimum turn width	$\mu\text{m}$	Inductance
Inductance value tolerance	%	Inductance
Grid resolution	$\mu\text{m}$	Inductance

- Thickness of the metal in which the inductor spiral is laid out,  $t$  (nm);
- Resistivity of the metal used for the spiral,  $\rho$  ( $\Omega\text{ m}$ );
- Permeability of the metal used for the spiral,  $\mu$  (H/m);
- Thickness of the oxide between the two top metals,  $t_m$  (nm);
- Relative permittivity of the oxide between the two top metals,  $\epsilon_{rm}$ ;
- Thickness of the oxide between the substrate and the top metal,  $t_{sm}$  (nm);
- Relative permittivity of the oxide between the substrate and the top metal,  $\epsilon_{rs}$ ;
- Thickness of the silicon substrate,  $t_{Si}$  ( $\mu\text{m}$ );
- Relative permittivity of the silicon substrate,  $\epsilon_{rSi}$ ; and
- Resistivity of the silicon substrate,  $\rho_{Si}$  ( $\Omega\text{ m}$ ).

The process parameters can normally be obtained or calculated from parameters obtained in the datasheets supplied by the process foundry. Table 11.5 summarizes the technology input parameters.

**Table 11.5** Process parameters for the spiral inductor design

Parameter	Unit
Thickness of metal in which the inductor spiral is laid out	nm
Resistivity of metal used for the spiral ( $\rho$ )	$\Omega\text{m}$
Permeability of metal used for the spiral ( $\mu$ )	H/m
Thickness of oxide between the two top metals ( $t_m$ )	nm
Relative permittivity of oxide between the two top metals ( $\epsilon_{rm}$ )	–
Thickness of oxide between substrate and top metal ( $t_{sm}$ )	nm
Relative permittivity of oxide between substrate and top metal ( $\epsilon_{rm}$ )	–
Thickness of the silicon substrate ( $t_{Si}$ )	$\mu\text{m}$
Relative permittivity of the silicon substrate ( $\epsilon_{rSi}$ )	–
Resistivity of the silicon substrate ( $\rho_{Si}$ )	$\Omega\text{m}$

### 11.3.1.3 Operating Frequency ( $f_o$ )

Operating frequency may be understood as the frequency at which the Q-factor will be highest for a particular geometry. For devices such as power amplifiers (PAs) or low-noise amplifiers (LNAs), the operating frequency is the center frequency of the channel.

## 11.3.2 Description and Flow Diagrams of Inductor Design Routine

The inductor design software routine consists of analysis and synthesis parts. Complete flow diagram of this routine is given in Fig. 11.13 [37, 38].

Analysis part of the routine is selected when user decides to provide inductor geometry parameters. Following this choice, a set of calculations that utilizes equations for the single- $\pi$  inductor model is performed. This model is simple yet accurate enough for the proof of concept. Nominal inductance is calculated by means of the data-fitted monomial equation as specified by (11.8), where coefficients are

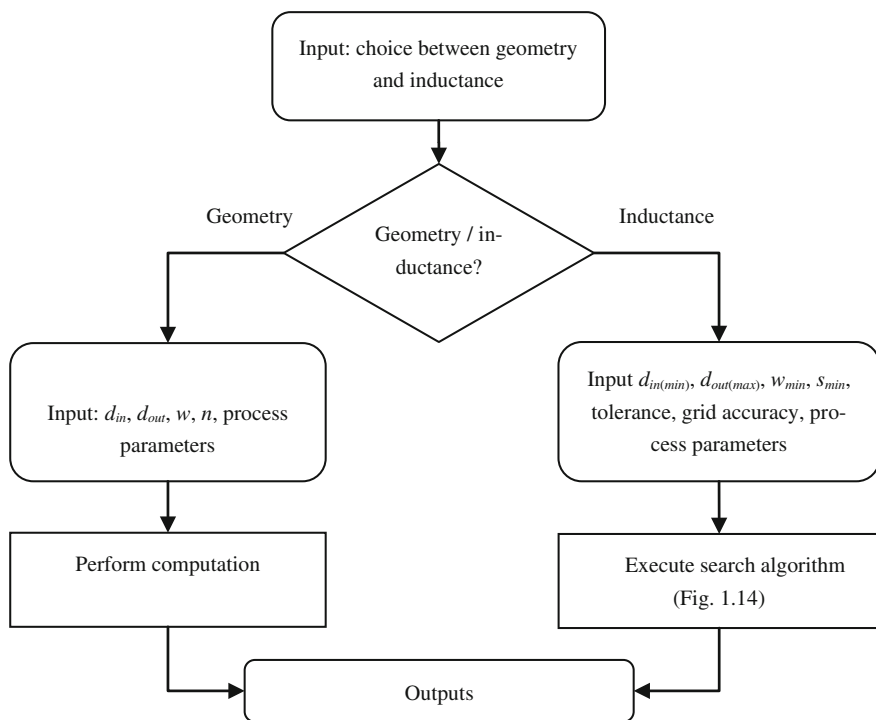


Fig. 11.13 Flow diagram of the inductor design routine

specified in Table 11.3. Parasitics are calculated by utilizing (11.9)–(11.19). Q-factor and resonance frequency are calculated by (11.20)–(11.23), and the apparent inductance at a particular frequency is calculated by (11.24).

Synthesis part of the routine is selected when user decides to provide inductance and required tolerance, constraining geometry detail as well as grid accuracy. In this case, an intelligent search algorithm shown in Fig. 11.13 is invoked. The search algorithm looks into a range of possible geometries and identifies a geometry that will result in the required inductance with high Q-factor within a certain tolerance (Fig. 11.14).

Synthesis algorithm in Fig. 11.13 commences by first computing constraints based on the geometry inputs, such as minimum and maximum number of turns ( $n$ ), minimum and maximum inner ( $d_{in}$ ) and outer ( $d_{out}$ ) diameter values, and spiral width ( $w$ ) in order to minimize the search space. The same equations used in the analysis part of the routine, (11.9)–(11.24), are used to compute inductance and quality factors of the minimum inductor geometry. Spacing between the turns  $s$  is then set to the minimum spacing that is feasible because densely spaced spirals are known to have the highest inductance. This in turn decreases the number of degrees of freedom and therefore the number of loops in the algorithm. Grid resolution is set and search commences. Each of  $n$ ,  $w$ , and  $d_{in}$  is then increased in a specific order, and  $L$  and  $Q$  are calculated for each step. Steps are chosen such that the whole allowed search space is covered but no unnecessary calculations are performed.

While more than one geometry will result in the tolerant inductance at a given frequency, each of these geometries will have a different Q-factor. The geometry that gives the highest Q-factor is chosen by the algorithm as its output. Accuracy of the algorithm depends on the tolerance for the required inductance values and on the search grid resolution. Although resolution is specified by the user, it cannot be chosen to be higher than allowed by the process design rules. Higher tolerance of the inductance value will result in less accurate inductance values, but there will be a greater probability that high-Q (or any) inductor geometry resulting in the particular inductance will be found with a lower grid resolution. This probability can again be increased by increasing the grid resolution, but with this increase, the time of execution and memory requirements of the search algorithm will also increase. It is up to users to decide which combination of inductance tolerance and grid resolution will be appropriate for a specific application. Time analysis of the calculation effort on two different systems for the synthesis of a typical 2-nH inductor in the ams AG (formerly austriamicrosystems) 0.35- $\mu\text{m}$  BiCMOS S35 process, for various tolerances and grid resolutions, is given in Table 11.6. This table also illustrates other trade-offs of different settings. It is clear from this analysis that higher grid resolution (in this case resolution higher than 1  $\mu\text{m}$ ) does not add to the quality of synthesized inductors, and therefore, time consumed for the inductor synthesis is acceptable even for the older system.

To illustrate how a programming or scripting language can be used to automate the process, the MATLAB code for the inductance search algorithm is provided in Fig. 11.15. MATLAB is only used as an example because the authors believe that

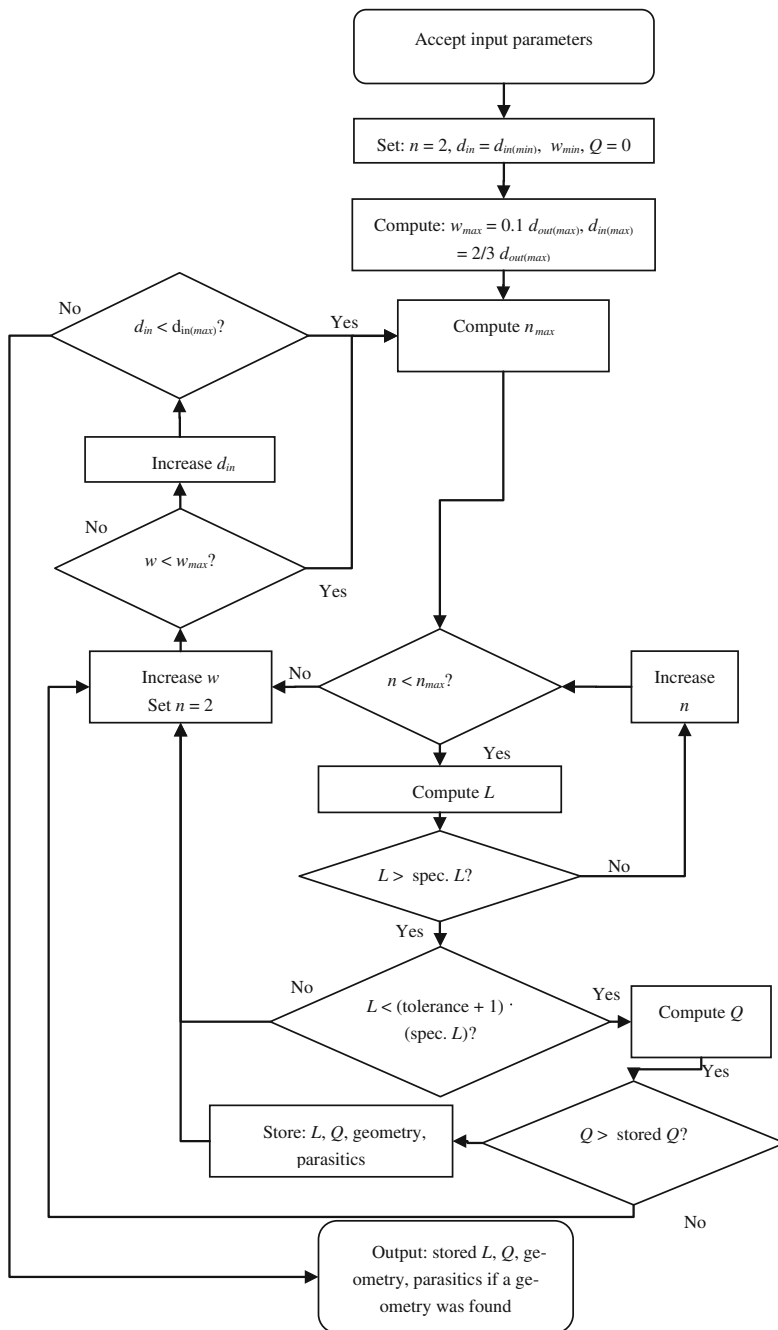


Fig. 11.14 Flow diagram of the inductance search algorithm

**Table 11.6** Analysis of computational efforts and trade-offs of different grid resolution and tolerance settings for the synthesis of a 2-nH inductor

Grid ( $\mu\text{m}$ )	Tolerance	0.1 %		0.5 %		1 %		5 %	
	System	Time (s)	Q-factor	Time (s)	Q-factor	Time (s)	Q-factor	Time (s)	Q-factor
0.1	Core2duo	147	6.82	147	6.82	147	6.82	151	6.82
	i7	55.6		55.7		55.5		56.7	
0.2	Core2duo	36.7	6.82	36.7	6.82	36.9	6.82	36.8	6.82
	i7	14.4		14.7		14.6		14.8	
0.5	Core2duo	6.01	6.82	5.98	6.82	5.99	6.82	5.97	6.82
	i7	3.62		3.08		2.81		2.79	
1	Core2duo	1.54	6.78	1.57	6.81	1.58	6.81	1.54	6.81
	i7	1.17		1.09		1.19		1.19	
2	Core2duo	Not found	–	0.435	6.81	0.435	6.81	0.438	6.81
	i7	Not found		0.483		0.478		0.521	
5	Core2duo	Not found	–	0.116	4.82	0.121	6.78	0.112	6.78
	i7	Not found		0.181		0.178		1.175	

many readers of this text would have at least a basic knowledge of the language. Alternatively, any programming or scripting language may be used for this purpose.

### 11.3.3 Design Outputs

The following quantities are numerical outputs of the inductor design routine that will be valuable for the RF designer:

1. Effective inductance value of the inductor at the operating frequency,  $L_S$  (nH);
2. Nominal inductance value of the inductor ( $Q \rightarrow \infty$ ),  $L_{\text{inf}}$  (nH);
3. Q-factor of the inductor at the operating frequency,  $Q$ ;
4. Resonant frequency of the inductor,  $f_r$  (GHz);
5. Width of the spiral ( $\mu\text{m}$ );
6. Spacing between the turns of the spiral ( $\mu\text{m}$ );
7. Input diameter of the spiral ( $\mu\text{m}$ );
8. Output diameter of the spiral ( $\mu\text{m}$ ); and
9. Number of turns of the spiral.

```

%This procedure searches for the inductance geometry with the
%highest quality factor given the inductance
%Initialize all storage variables to zero
Qstored = 0; fostored = 0; Lcstored = 0; Rsstored = 0; RSistored =
0; CSistored = 0; Coxstored = 0; Csstored = 0; wstored = 0;
sstored = 0; dinstored = 0; doutstored = 0; nstored = 0;
fprintf('\nLooking for geometry with highest Q-factor...\n\n');
%Initialize geometry parameters to default minimum/maximum values
Lc = 0;
dout = 0;
s = smin;
din = dinmin;
w = wmin;
n = 2;
%Inductance search algorithm
while (din < 2*doutmax/3)
    s = smin;
    w = wmin;
    while (w <= doutmax/10)
        n = 2;
        dout = 0;
        while (dout < doutmax)
            dout = din + 2*n*w + 2*(n-1)*s;
            if (dout > doutmax)
                break
            end%if
            davg = (din + dout) / 2;
            Lc = b * dout^a1 * w^a2 * davg^a3 * n^a4 * s^a5;
            calcParasitics; %Procedure to calc parasitics
            Lcc = Lc/1e9;
            Lzz = Lz*1e9;
            if (Lzz > Ls)
                if (Lzz < (1 + tolerance) * Ls)
                    %Calculate Q-factor
                    Rp = 1/(omega^2*Cox^2*RSi) + RSi*(Cox + CSi)^2/Cox^2;
                    Cp = Cox*(1 + omega^2*(Cox + CSi)*CSi*RSi^2)/(1 +
omega^2*(Cox + CSi)^2*RSi^2);
                    Q = omega*Lcc/Rs*Rp/(Rp + ((omega*Lcc/Rs)^2 +
1)*Rs)*(1 - (Cp + Cs)*(omega^2*Lcc + Rs^2/Lcc));
                    fo = 1/(2*pi)*sqrt(1/(Lcc*(Cp + Cs)) - (Rs/Lcc)^2);
                    if (Q > Qstored)
                        Qstored = Q; fostored = fo; Lclfstored = Lc;
                        Lcstored = Lzz; Rsstored = Rs; RSistored = RSi;
                        CSistored = CSi; Coxstored = Cox; Csstored = Cs;
                        wstored = w; sstored = s; dinstored = din;
                        doutstored = dout; nstored = n;
                    end%if
                end%if
                Lc = 0;
                n = 1;
                break
            end%if
            n = n + 1;
        end%while
    w = w + resolution;

```

Fig. 11.15 MATLAB code for the inductance search algorithm

```

    din = din + resolution;
end%while
%==== OUTPUT PARAMS ====%
if Qstored < 1
    fprintf('Could not find a geometry for %.2f nH\nlimited to
dinmin and doutmax with Q greater than 1 at %.2f MHz.\n', Ls,
f/1e6)
end%if
if Qstored >=1
    fprintf('Ls = %.2f nH \n', Lcstored);
    fprintf('Lslf = %.2f nH \n', Lclfstored);
    fprintf('Q = %.2f \n', floor(100 * Qstored + 0.5) / 100);
    fprintf('fo = %.2f GHz\n', floor(fostored/1e7 + 0.5) / 100);
    fprintf('w = %.2f um\n', floor(100*wstored + 0.5) / 100);
    fprintf('s = %.2f um\n', floor(100*sstored + 0.5) / 100);
    fprintf('din = %.2f um\n', floor(100*dinstored + 0.5) / 100);
    fprintf('dout = %.2f um\n', floor(100*doutstored + 0.5) / 100);
    fprintf('n = %d\n', floor(100*nstored + 0.5) / 100);
end%if

```

Fig. 11.15 (continued)

## 11.4 Verification of the Spiral Inductor Model and the Inductance Search Algorithm

The inductance search algorithm was used to design ten metal-three (3M) and ten thick-metal (TM) inductors fabricated over a standard resistivity substrate at common frequencies of 1, 2, 2.4, and 5 GHz in the ams AG S35 process. The smallest inductor value designed for was 0.5 nH, followed by nine inductors in increments of 0.5 nH. Tables 11.7 and 11.8 show geometric parameters, low-frequency inductance values, and the Q-factor of each designed 3M and TM inductor, respectively. To verify the predicted values, EM simulation on the designed inductors was performed in Virtuoso Spiral Inductor Modeler [32]. The solver for the Spiral Inductor Modeler employs partial element equivalent circuit (PEEC) algorithm in the generation of macromodels for the spiral components. Electrostatic and magnetostatic EM solvers are invoked separately to extract the capacitive and inductive parameters of the spiral inductor structure. A process file with information on metal and dielectric layers was required by the modeler and it needed to be manually created.

Aforementioned tables show that inductance values obtained using the inductor design routine correspond with simulated inductance values. Good correspondence between predicted and simulated values in terms of Q-factor values exists for 3M inductors as well, whereas in the case of TM inductors, simulated Q-factors are larger than those of the calculated Q-factors. This discrepancy can be explained: As the impedance of parasitic elements in the  $RL$  model of the spiral (with oxide and substrate effects ignored) approaches that of inductive reactance near the peak frequency, the model yields a pessimistic estimate of the actual Q-factor of the

**Table 11.7** Metal-3 inductors designed with inductance search algorithm

Frequency (GHz)	Nominal inductance (nH)	Calculated LF inductance (nH)	Calculated Q	EM inductance (nH)	EM Q	$d_{out}$ ( $\mu\text{m}$ )	$d_{in}$ ( $\mu\text{m}$ )	$w$ ( $\mu\text{m}$ )	$s$ ( $\mu\text{m}$ )	$n$
1	0.5	0.50	3.37	0.50	3.00	220	30	47	1	2
1	1	1.00	4.63	0.94	4.15	291	93	49	1	2
1	1.5	1.48	5.22	1.40	4.75	347	149	49	1	2
1	2	1.95	5.47	1.90	5.11	397	199	49	1	2
1	2.5	2.39	5.50	2.33	2.33	441	243	49	1	2
1	3	2.83	5.39	2.75	5.27	483	285	49	1	2
1	3.5	3.29	5.10	3.22	3.22	500	326	43	1	2
1	4	3.67	4.80	3.52	4.52	426	164	43	1	3
1	4.5	4.12	4.66	3.97	4.56	427	189	39	1	3
1	5	4.55	4.60	4.52	4.41	431	211	36	1	3
2	0.5	0.49	5.79	0.45	4.29	210	32	55	1	2
2	1	0.97	7.16	0.91	5.92	288	90	49	1	2
2	1.5	1.41	6.89	1.32	6.11	339	141	49	1	2
2	2	1.86	6.24	1.73	6.23	349	191	39	1	2
2	2.5	2.32	5.64	2.26	5.68	363	233	32	1	2
2	3	2.76	5.12	2.72	5.27	384	270	28	1	2
2	3.5	3.21	4.81	3.01	5.33	276	152	20	1	3
2	4	3.65	4.53	3.45	5.19	283	171	18	1	3
2	4.5	4.06	4.27	3.86	4.98	294	188	17	1	3
2	5	4.51	4.05	4.35	4.76	241	123	14	1	4
2.4	0.5	0.49	6.72	0.45	4.94	216	30	46	1	2
2.4	1	0.95	7.63	0.90	6.38	286	88	49	1	2

(continued)



Table 11.7 (continued)

Frequency (GHz)	Nominal inductance (nH)	Calculated LF inductance (nH)	Calculated Q	EM inductance (nH)	EM Q	$d_{out}$ ( $\mu\text{m}$ )	$d_{in}$ ( $\mu\text{m}$ )	$w$ ( $\mu\text{m}$ )	$s$ ( $\mu\text{m}$ )	$n$
2.4	1.5	1.39	6.95	1.34	6.34	316	142	43	1	2
2.4	2	1.86	6.18	1.82	6.25	320	190	32	1	2
2.4	2.5	2.30	5.51	2.26	5.47	339	229	27	1	2
2.4	3	2.76	5.04	2.71	5.32	249	131	19	1	3
2.4	3.5	3.17	4.69	3.12	4.89	262	150	18	1	3
2.4	4	3.61	4.39	3.59	4.74	262	168	15	1	3
2.4	4.5	4.03	4.12	4.03	4.60	220	110	13	1	4
2.4	5	4.47	3.91	4.50	4.55	216	122	11	1	4
5	0.5	0.47	9.48	0.41	6.05	200	30	42	1	2
5	1	0.92	8.22	0.88	6.93	209	95	28	1	2
5	1.5	1.36	6.76	1.34	5.88	222	140	20	1	2
5	2	1.81	5.71	1.80	5.48	169	87	13	1	3
5	2.5	2.22	5.03	2.20	5.15	176	106	11	1	3
5	3	2.60	4.46	2.59	4.45	186	122	10	1	3
5	3.5	2.97	4.02	2.96	4.03	160	82	9	1	4
5	4	3.46	3.70	3.52	4.43	148	94	6	1	4
5	4.5	3.98	3.43	4.07	4.30	141	103	4	1	4
5	5	4.29	3.27	4.42	4.24	106	38	4	1	7

**Table 11.8** Thick-metal inductors designed with inductance search algorithm

Frequency (GHz)	Nominal inductance (nH)	Calculated LF inductance (nH)	Calculated Q	EM inductance (nH)	EM Q	$d_{out}$ ( $\mu\text{m}$ )	$d_{in}$ ( $\mu\text{m}$ )	$w$ ( $\mu\text{m}$ )	$s$ ( $\mu\text{m}$ )	$n$
1	0.5	0.50	7.43	0.38	4.71	216	30	48	2	2
1	1	0.99	10.1	0.93	8.13	299	95	50	2	2
1	1.5	1.47	11.1	1.39	9.96	355	151	50	2	2
1	2	1.95	11.5	1.84	11.1	406	202	50	2	2
1	2.5	2.39	11.4	2.30	11.7	451	247	50	2	2
1	3	2.81	10.9	2.70	12.0	493	289	50	2	2
1	3.5	3.29	10.3	3.20	11.6	499	331	41	2	2
1	4	3.71	9.78	3.49	9.62	403	173	37	2	3
1	4.5	4.17	9.49	3.95	9.89	409	197	34	2	3
1	5	4.60	9.21	4.40	9.99	418	218	32	2	3
2	0.5	0.49	11.4	0.43	9.17	222	30	47	2	2
2	1	0.96	13.0	0.88	13.4	295	91	50	2	2
2	1.5	1.42	11.9	1.35	14.6	316	148	41	2	2
2	2	1.88	10.8	1.82	14.2	324	196	31	2	2
2	2.5	2.33	9.80	2.27	13.8	348	236	37	2	2
2	3	2.79	9.08	2.66	11.6	268	134	21	2	3
2	3.5	3.21	8.55	3.10	11.6	276	154	19	2	3
2	4	3.64	8.05	3.54	11.6	283	173	17	2	3
2	4.5	4.09	7.59	3.96	11.4	294	190	16	2	3
2	5	4.49	7.25	4.39	10.3	240	124	13	2	4
2.4	0.5	0.49	12.4	0.43	10.7	219	31	46	2	2
2.4	1	0.94	13.0	0.86	14.4	286	90	48	2	2

(continued)

Table 11.8 (continued)

Frequency (GHz)	Nominal inductance (nH)	Calculated LF inductance (nH)	Calculated Q	EM inductance (nH)	EM Q	$d_{out}$ ( $\mu\text{m}$ )	$d_{in}$ ( $\mu\text{m}$ )	$w$ ( $\mu\text{m}$ )	$s$ ( $\mu\text{m}$ )	$n$
2.4	1.5	1.41	11.6	1.36	15.3	289	149	34	2	2
2.4	2	1.88	10.3	1.83	14.7	302	194	26	2	2
2.4	2.5	2.34	9.25	2.24	11.8	229	113	18	2	3
2.4	3	2.78	8.61	2.69	12.0	238	134	16	2	3
2.4	3.5	3.18	8.01	3.08	12.0	256	152	16	2	3
2.4	4	3.63	7.47	3.56	11.7	256	170	13	2	3
2.4	4.5	4.54	7.04	4.00	10.6	213	113	11	2	4
2.4	5	4.46	6.67	4.36	10.6	223	123	11	2	4
5	0.5	0.47	14.4	0.41	17.3	200	32	41	2	2
5	1	0.93	12.1	0.89	18.9	199	99	24	2	2
5	1.5	1.38	9.83	1.35	17.0	215	143	17	2	2
5	2	1.82	9.37	1.74	14.0	163	89	11	2	3
5	2.5	2.26	7.31	2.18	13.5	170	108	9	2	3
5	3	2.70	6.74	2.62	10.6	123	47	6	2	5
5	3.5	3.09	6.23	3.04	12.0	182	138	6	2	3
5	4	3.42	5.88	3.32	10.4	137	61	6	2	5
5	4.5	3.79	5.44	3.70	10.8	163	103	6	2	4
5	5	4.07	5.02	3.98	9.65	149	73	6	2	5

spiral [33]. 3M inductors lie closer to the substrate and have larger resistances than TM inductors, so this effect is less prominent. The fact that the Q-factor is underestimated rather than overestimated is an advantage, since the TM inductors designed by the inductor design routine will perform better than predicted, which will be acceptable in many cases. Where higher accuracy is needed, the use of one of the more detailed models may be explored.

Furthermore, inductance routine was used to predict inductances and Q-factors of several spiral inductor geometries provided and measured by ams AG. The measurement results showed that inductance values are correctly predicted (within 3.7 %) by the inductor models used for the inductance search algorithm with Q-factors exhibiting the same behavior as shown by EM simulations. Details of this study can be found in [37].

## 11.5 IC Design Flow Integration

Simple programming techniques may be used to interpret numerical design outputs described previously to export the SPICE<sup>2</sup> netlist and layout (GDS<sup>3</sup> format) of the designed inductor structure. The SPICE netlist of the inductor structure, complete with the inductance value and parasitics calculated for the chosen inductor model, may be used in SPICE simulations to avoid drawing of the schematic of the inductor with its parasitics in the schematic editor. Layout of the inductor may be imported into layout software to eliminate the need to draw any inductor layout structures. With this in place, minimum effort is needed to deploy inductors designed using this methodology in full system design.

To demonstrate complete design flow integration, several 2.4-GHz Class-E and Class-F PAs were designed and fabricated in the IBM 7WL (180 nm) process. Another set of developed software routines was used to first perform each amplifier design. The designs required several spiral inductors for both amplifier design and the design of the matching networks. All inductors were designed using the software routine presented together with netlist and layout extraction. This allowed for the complete system to be simulated in SPICE before layouts were completed and systems were sent for prototyping.

Detailed presentation of simulation and prototyping results is beyond the scope of this chapter and the reader is referred to [34].

---

<sup>2</sup>SPICE stands for Simulation Program with Integrated Circuit Emphasis.

<sup>3</sup>GDS stands for Graphic Database System.

## 11.6 Going Beyond RF Frequencies

As frequency of operation increases beyond about 20 GHz (micro-/millimeter-wave as opposed to RF frequencies), it becomes possible to utilize transmission lines instead of passive components. Transmission line theory may be applied in order to expand the algorithms presented in this chapter for use in millimeter-wave applications [35, 36].

## 11.7 Conclusion

The aim of this chapter was to introduce the reader to the concept of spiral inductor design and to show how optimum inductor design can aid performance optimization of RF devices. It was pointed out that due to the indeterministic behavior of inductance and parasitics of inductors, design using simple equations should be replaced by a more streamlined methodology even for very simple inductor geometries. A methodology for synthesis-based design of planar spiral inductors where numerous geometries are searched through in order to fit the start conditions was conceptualized, but it was concluded that it becomes too tedious to do this by hand and that software-aided design is recommended. The readers were given an example of the algorithm implemented by using a MATLAB script for the simpler, single- $\pi$ , model, and provided with sufficient information to probe further. Computational intelligence could be applied to the resulting algorithm, including the IC layout, and in this way lead to further computer-aided design and optimization. As proof of the concept, several inductors were synthesized using this methodology and their inductances and quality factors were presented and evaluated against simulation and measurement results. Finally, the reader was referred to texts where optimum inductors have aided practical RF design.

**Acknowledgment** The authors thank Azoteq (Pty) Ltd, South Africa, for their support.

## References

1. Ludwig, R., Bretchko, P.: RF Circuit Design: Theory and Applications. Prentice Hall, Upper Saddle River (2000)
2. Uyanik, H., Tarim, N.: Compact low voltage high-Q CMOS active inductor suitable for RF applications. *Analog Integr. Circ. Sig. Process.* **51**, 191–194 (2007)
3. Ler, C.L., A'ain, A.K.B., Kordesh, A.V.: CMOS source degenerated differential active inductor. *Electron. Lett.* **44**(3), 196–197 (2008)
4. De Los Santos, H.J.: MEMS for RF/microwave wireless applications. *Microwave J.* **44**(3), 20–24, 28, 32–41 (2001)
5. Khoo, Y.M., Lim, T.G., Ho, S.W., Li, R., Xiong, Y.Z., Zhang, X.: Enhancement of silicon-based inductor Q-factor using polymer cavity. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2**(12), 1973–1979 (2012)

6. Gu, L., Li, X.: High-Q solenoid inductors with a CMOS-compatible concave-suspending MEMS process. *J. Microelectromech. Syst.* **16**(5), 1162–1172 (2007)
7. Chua, L.C., Fork, D.K., Van Schuylenbergh, K., Lu, J.P.: Out-of-plane high-Q inductors on low-resistance silicon. *J. Microelectromech. Syst.* **12**(6), 989–995 (2003)
8. Foty, D.: Prospects for nanoscale electron devices: some little-recognized problems. In: *Proceedings of the International Semiconductor Conference (CAS)*, Sinaia, 13–15 Oct 2008
9. Murad, S.A.Z., Pokharel, R.K., Kanaya, H., Yoshida, K., Nizhnik, O.: A 2.4-GHz 0.18- $\mu$ m CMOS class E single-ended switching power amplifier with a self-biased cascode. *Int. J. Electron. Comm.* **64**(9), 813–818 (2010)
10. Khatri, H., Gudem, P.S., Larson, L.E.: Integrated RF interference suppression filter design using bond-wire inductors. *IEEE Trans. Microw. Theory Tech.* **56**(5), 1024–1034 (2008)
11. Masu, K., Okada, K., Ito, H.: RF passive components using metal line on Si CMOS. *Trans. Electron.* **E89–C**(6), 681–691 (2006)
12. Vroubel, M., Zhuang, Y., Rejaei, B., Burghartz, J.N.: Integrated tunable magnetic RF inductor. *IEEE Electron. Device Lett.* **25**(12), 787–789 (2004)
13. Seo, S., Ryu, N., Choi, H., Jeong, Y.: Novel high-Q inductor using active inductor structure and feedback parallel resonance circuit. In: *Proceedings of IEEE Radio frequency integrated circuits symposium*, Honolulu, 3–5 June 2007
14. Zine-El-Abidine, I., Okoniewski, M.: High quality factor micromachined toroid and solenoid inductors. In *Proceedings of the 37th European Microwave Conference*, Munich, 9–12 Oct 2007
15. Mohan, S.S., del Mar Hershenson, M., Boyd, S.P., Lee, T.H.: Simple accurate expressions for planar spiral inductances. *IEEE J. Solid State Circuits* **34**(10), 1419–1424 (1999)
16. Niknejad, A.M., Meyer, R.G.: *Design, Simulation and Application of Inductors and Transformers for Si RF ICs*. Springer, New York (2000)
17. Xu, X., Li, P., Cai, M., Han, B.: Design of novel high-Q-factor multipath stacked on-chip spiral inductors. *IEEE Trans. Electron Devices* **59**(8), 2011–2018 (2012)
18. Pei, S., Wanrong, Z., Lu, H., Dongyue, J., Hongyun, X.: Improving the quality factor of an RF spiral inductor with non-uniform metal width and non-uniform coil spacing. *CIE J. Semiconductors* **32**(6), 1–5 (2011)
19. Wang, T.P., Li, Z.W., Tsai, H.Y.: Performance improvement of a 0.18- $\mu$ m CMOS microwave amplifier using micromachined suspended inductors: theory and experiment. *IEEE Trans. Electron Devices* **60**(5), 1738–1744 (2013)
20. Koutsyannopoulos, Y.K., Papananos, Y.: Systematic analysis and modeling of integrated inductors and transformers in RF IC design. *IEEE Trans. Circuits Sys. II Analog and Digital Signal Proc.* **47**(8), 699–713 (2000)
21. Watson, A.C., Melendy, D., Francis, P., Hwang, K., Weisshaar, A.: A comprehensive compact-modeling methodology for spiral inductors in silicon-based RFICs. *IEEE Trans. Microw. Theory Tech.* **52**(3), 849–857 (2004)
22. Lee, K.Y., Mohammadi, S., Bhattacharya, P.K., Katehi, L.P.B.: Compact models based on transmission-line concept for integrated capacitors and inductors. *IEEE Trans. Microw. Theory Tech.* **54**(12), 4141–4148 (2006)
23. Wang, H., Sun, L., Yu, Z., Gao, J.: Analysis of modeling approaches for on-chip spiral inductors. *Int. J. RF and Microw. Comput. Aided Eng.* **22** (3), 377–386 (2012)
24. Musunuri, S., Chapman, P.L., Zou, J., Liu, C.: Design issues for monolithic DC–DC converters. *IEEE Trans. Power Electron.* **20**(3), 639–649 (2005)
25. Yue, C.P., Wong, S.S.: Physical modeling of spiral inductors on silicon. *IEEE Trans. Electron Devices* **47**(3), 560–568 (2000)
26. Huo, X., Chan, P.C.H., Chen, K.J., Luong, H.C.: A physical model for on-chip spiral inductors with accurate substrate modeling. *IEEE Trans. Electron Devices* **53**(12), 2942–2949 (2006)
27. Lee, C.Y., Chen, T.S., Deng, J.D.S., Kao, C.H.: A simple systematic spiral inductor design with perfected Q improvement for CMOS RFIC application. *Trans. Microw. Theory Tech.* **53** (2), 523–528 (2005)

28. Sun, H., Liu, Z., Zhao, J., Wang, L., Zhu, J.: The enhancement of Q-factor of planar spiral inductor with low-temperature annealing. *IEEE Trans. Electron Devices* **55**(3), 931–936 (2008)
29. Xue, C., Yao, F., Cheng, B., Wang, Q.: Effect of the silicon substrate structure on chip spiral inductor. *Front. Electr. Electron. Eng. China* **3**(1), 110–115 (2008)
30. Austriamicrosystems: 0.35  $\mu\text{m}$  HBT BiCMOS RF SPICE models, Unterpremstätten (2005)
31. Hastings, A.: *The Art of Analog Layout*. Prentice Hall, Upper Saddle River (2006)
32. Zhu, H.: *Modeling and Simulation of On-Chip Spiral Inductors and Transformers*. Cadence Design Systems, San Jose (2000)
33. IBM Corporation: *BiCMOS-7WL Model Reference Guide*. Armonk (2008)
34. Božanić, M., Sinha, S., Müller, A.: Streamlined design of SiGe based power amplifiers. *Rom. J. Inf. Sci. Technol.* **13**(1), 22–32 (2010)
35. Božanić, M., Sinha, S.: Switch-mode power amplifier design method. *Microwave and Opt. Technol. Lett.* **53**(12), 2724–2728 (2011)
36. Foty, D., Sinha, S., Weststrate, M., Coetzee, C., Uys, A.H., Sibanda, E.: mm-Wave radio communications systems: the quest continues. In: *Proceedings of the 3rd International Radio Electronics Forum (IREF) on Applied Radio Electronics: The State and Prospects of Development*, Kharkov, 22–24 Oct 2008
37. Božanić, M., Sinha, S.: Design methodology for SiGe-based class-E power amplifier. In: *Proceedings of the 1st South African Conference on Semi and Superconductor Technology*, Stellenbosch, 8–9 Apr 2009/I
38. Hsu, H.M.: Investigation on the layout parameters of on-chip inductor. *Microelectron. J.* **37**(8), 800–803 (2006)

# Chapter 12

## Optimization of RF On-Chip Inductors Using Genetic Algorithms

Eman Omar Farhat, Kristian Zarb Adami, Owen Casha  
and John Abela

**Abstract** This chapter discusses the optimization of the geometry of RF on-chip inductors by means of a genetic algorithm in order to achieve adequate performance. Necessary background theory together with the modeling of these inductors is included in order to aid the discussion. A set of guidelines for the design of such inductors with a good quality factor in a standard CMOS process is also provided. The optimization process is initialized by using a set of empirical formulae in order to estimate the physical parameters of the required structure as constrained by the technology. Then, automated design optimization is executed to further improve its performance by means of dedicated software packages. The authors explain how to use state-of-the-art computer-aided design tools in the optimization process and how to efficiently simulate the inductor performance using electromagnetic simulators.

### 12.1 Introduction

The design and fabrication of on-chip radio frequency (RF) inductors have constantly demonstrated wide interest due to the need of providing single-chip solutions for integrated transceivers. They are required in matching networks, resonator tanks, baluns, and as inductive loads. The performance of radio frequency integrated circuits (RFIC), including low-noise amplifiers, mixers, and oscillators, is greatly limited by the quality factor of such passive elements. This is particularly

---

E.O. Farhat · O. Casha (✉) · J. Abela  
University of Malta, Msida, Malta  
e-mail: owen.casha@um.edu.mt

E.O. Farhat  
e-mail: efar0026@um.edu.mt

K.Z. Adami  
University of Oxford, Oxford, UK  
e-mail: kza@astro.ox.ac.uk

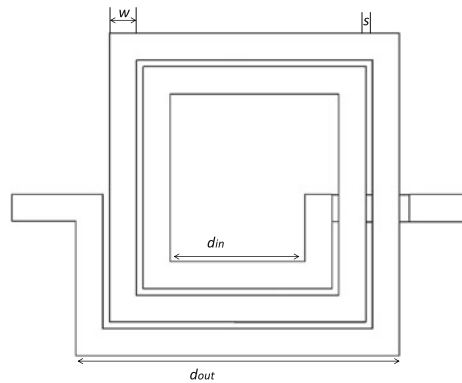


true in standard silicon processes (e.g., CMOS), whose characteristics (such as substrate coupling) contribute to the relatively poor performance of the available passive components. CMOS processes are often chosen to implement RF circuit blocks due to their low cost, high level of integration, and availability. RFIC designers generally demand for on-chip inductors to have a desirable value with a high self-resonant frequency and high quality factor and occupy a small layout area. In general, passive inductors fabricated in a standard CMOS fabrication process have small inductance in the range of nanohenries.

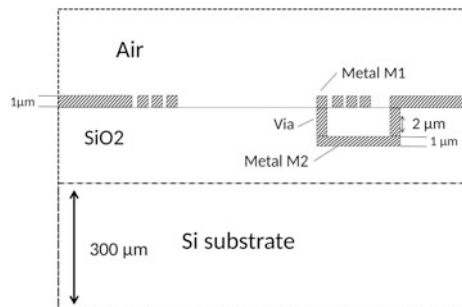
Inductors are circuit elements which store energy in the form of a magnetic field. In RFIC, spiral inductors are fabricated on the topmost metals available in the process. For instance, Figs. 12.1 and 12.2 illustrate the top and cross-sectional views of a square inductor fabricated in a generic CMOS process [1]. The top metal layer (M1) is used for the spiral, while the lower metal layer (M2) is used for the underpass as depicted in Fig. 12.2.

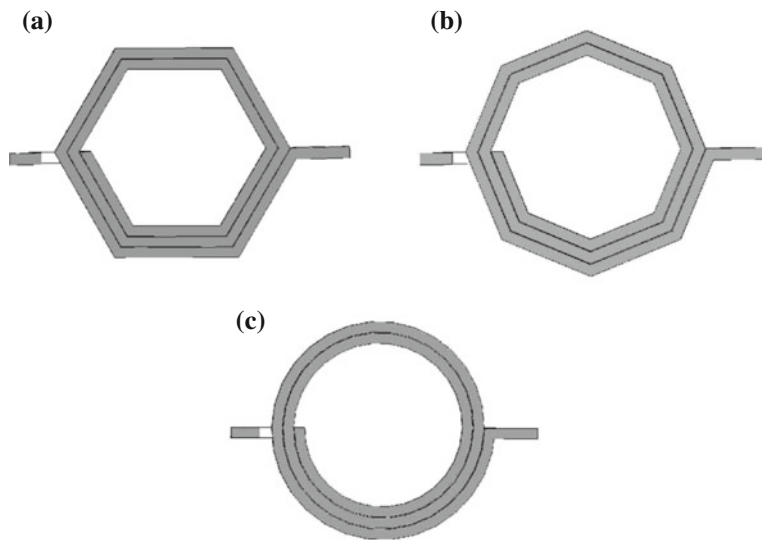
Spiral inductors are mainly defined by a number of geometrical parameters: the number of turns  $n$ , the width of the metal trace  $w$ , the turn spacing  $s$ , the inner diameter  $d_{in}$ , and the outer diameter  $d_{out}$ . They can be implemented in different

**Fig. 12.1** Layout of a square spiral inductor (*top view*)



**Fig. 12.2** Cross-sectional view of the implementation of a square inductor in a generic CMOS process



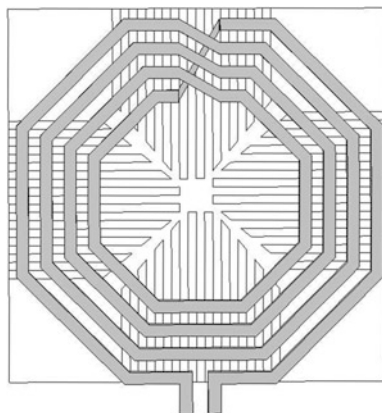


**Fig. 12.3** Spiral inductor topologies [1]. **a** A hexagonal spiral. **b** An octagonal spiral. **c** A circular spiral

shapes such as hexagonal, octagonal, and circular configurations as shown in Fig. 12.3. The symmetrical forms of these inductors are often used in differential circuits such as voltage-controlled oscillators and low-noise amplifiers [3].

In order to reduce the substrate losses and enhance the inductor quality factor, a patterned ground shield (PSG) fabricated via a metal layer which is located between the spiral inductor and the substrate can be employed [4]. This is shown in Fig. 12.4.

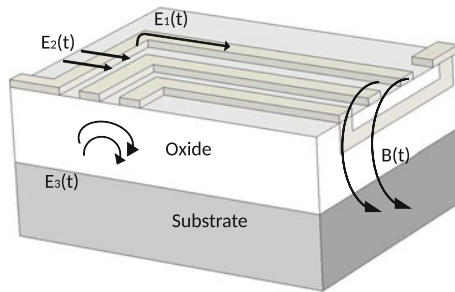
**Fig. 12.4** A spiral inductor with the patterned ground shield [3]



## 12.2 Losses in CMOS Inductors

Inductors implemented in a standard CMOS technology experience a number of electric and magnetic effects, which limit their performance. When a potential difference is applied to the terminals of the integrated inductor, magnetic and three electric fields appear as illustrated in Fig. 12.5 [5]. A magnetic field  $B(t)$  is generated as the ac current flows through the tracks of the spiral. This induces an inductive behavior, while parasitic currents flow in the tracks and the substrate. According to Faraday's law, a time-varying magnetic field induces an electric field in the substrate which forces an image current to flow in the substrate opposite in direction to the current in the winding directly above it. Thus, this adds a loss to the CMOS inductor, since the substrate acts as an undesired secondary winding which loads the coil. In the case of larger inductors, the magnetic field penetrates deeper into the substrate causing higher substrate losses. To minimize the effect of such substrate losses, some technologies provide the possibility to either use nonstandard high-resistivity silicon substrate or have a post-processing micromachining step in order to etch the substrate underneath the inductor [6]. Additionally,  $f$  induces eddy currents in the center of winding which affect the inner turns of the inductor. This is also known as current crowding [7].

The electric field  $E_1(t)$  appears as the potential difference is applied between the terminals of the spiral. Because of the finite metal resistivity, ohmic losses occur as the current flows through the track. In typical CMOS processes, aluminum (and sometimes copper) is used as the interconnecting metal. Its sheet resistivity lies between 30 and 70 m $\Omega$ /sq, depending on the metallization thickness and the type of aluminum alloy. Therefore, the dc resistance of the spiral can be easily calculated by the product of the sheet resistance and the number of spiral turns. An improvement in the quality factor can be achieved by an introduction of a copper metallization track with a thicker upper-level interconnect metal. Also, strapping multiple metallization levels to create a multilayer spiral effectively lowers the dc winding resistance [7].



**Fig. 12.5** Electric and magnetic fields associated with a square spiral inductor implemented in a generic CMOS process [5]

The potential difference between the turns in the metal that forms the spiral causes an electric field  $E_2(t)$ . Thus, capacitive coupling is induced between the tracks because of the dielectric material. Usually, the winding of the spirals in a CMOS technology is separated from the substrate by a thin layer of silicon dioxide. The silicon substrate is neither a perfect conductor nor an insulator. Therefore, there are losses in the reactive fields that surround the windings of the spiral. The substrate is a heavily doped  $p$ -type material and it is tied to ground such that a potential difference appears between the spiral and the substrate. Therefore, capacitive coupling is created between the inductive structure and the substrate. The induced electric field  $E_3(t)$  penetrates into the conductive substrate, causing an ohmic loss. This allows RF currents to interact with the substrate, lowering the inductance value. Additionally, it increases the parasitic capacitance and lowers the self-resonant frequency. Reducing the trace width decreases the effect of this parasitic capacitance but in turn increases the series resistance. Hence, this implies that using wide traces helps to overcome the low thin-film conductivity of the metalization. On the other hand, this limits the possibility of creating large-value inductors. As a conclusion, the major losses in a standard CMOS technology are due to the effect of the substrate. This is still an important limiting factor, even when the conductivity of the spiral windings is not an issue.

### 12.3 Quality Factor

The quality factor  $Q$  is a fundamental parameter associated with energy storing elements and it is the measure of the storage efficiency. Since inductors store magnetic energy, they have an associated quality factor which offers an insight on their performance. It is defined as the ratio of the energy stored per cycle to the energy dissipated per cycle, as given in (12.1).

$$Q = 2\pi \frac{\text{maximum energy stored}}{\text{energy dissipated}} \quad (12.1)$$

For inductors, the only form of required energy is that stored in the magnetic field, while any energy stored in the electric field is a loss. In addition, inductors have an associated self-resonant frequency  $f_{sr}$  beyond which it starts to behave capacitively. At  $f_{sr}$ , the peak magnetic and electric energies are equal, such that  $Q$  becomes zero.  $Q$  is proportional to the net magnetic energy stored, which is equal to the difference between the peak magnetic and electric energies. Based on this definition and the lumped element  $\pi$ -model (refer to Sect. 12.5),  $Q$  is calculated using (12.2) [8].

$$Q = \frac{\omega L_s}{R_s} \cdot \frac{R_p \left( 1 - \frac{R_s^2 C_p}{L_s} - \omega^2 L_s C_p \right)}{R_p + \left[ \left( \frac{\omega L_s}{R_s} \right)^2 + 1 \right] R_s} \quad (12.2)$$

where  $L_s$ ,  $R_s$ , and  $C_s$  represent the series inductance, the metal series resistance, and the capacitive coupling, respectively.  $C_p$  and  $R_p$  represent the overall parasitic effect of the oxide and the silicon substrate. Inductors implemented in a standard silicon (Si) technology such as CMOS have a low  $Q$  resulting from the relatively high-conductivity Si substrate. Planar spirals that are fabricated on GaAs substrates exhibit  $Q$  in the range of 20–40, while the  $Q$  of inductors implemented on a Si substrate is much lower. Discrete off-chip inductors provide a much higher quality factor, but it is desirable to reduce the board-level complexity and limit the cost by using on-chip inductors. Bond wires are frequently used as an alternative to some on-chip inductors due to their high  $Q$ . They provide a higher surface area per unit length when compared to planar spirals, thus having less resistive loss and as a consequence a higher quality factor. However, they also suffer from large variations in the inductance value. Additionally, wire bonding is a mechanical process that cannot be tightly controlled as in the case of a photolithographic process [9]. Remarkably, the inductance of on-chip inductors is solely defined by their physical geometry, since modern photolithographic processes have stringent geometric tolerances limiting any variations in the inductor performance [7].

## 12.4 Guidelines for On-Chip Inductor Design

The square spiral topology is the most commonly used in the implementation of on-chip inductors. Another frequently used topology is the octagonal spiral topology. As the number of geometry sides increases, both the resistance and the inductance of the structure increase since a larger length of metal track would be used. However, the inductance value increases at a faster rate than that of the resistance, thus resulting in an increase of the quality factor. In this regard, the circular spiral geometry provides the largest perimeter for the same radius, thus maximizing the inductance and quality factor. Although it is preferable to employ a circular configuration, it is often not permitted by standard integrated circuit technologies. Additionally, non-Manhattan geometries are not supported by many technologies [9].

Reducing the resistance per unit length of an inductor trace is imperative to increase the quality factor and this is usually done by making use of thick metal layers. Alternatively, in conventional CMOS processes, two or more metal layers are connected together to thicken the inductor trace to generate a so-called multi-layer spiral inductors. The resistance of the inductors becomes smaller as the number of layers shunted together is increased, thus leading to an increase in the quality factor. In practice, the number of metal layers in a CMOS process may vary and this increase in the quality factor is often limited because of the finite resistance of the interconnecting vias. It is not recommended to use the metal layers closer to the substrate, because this would increase the parasitic capacitance associated with the structure, thus reducing the self-resonant frequency of the inductor.

Due to the eddy current effect, the innermost turns of the coils suffer enormously from a high resistance which affects the overall quality factor. In addition, the innermost turns give minimal contribution to the inductance. Hence, it is recommended to design a hollow coil. The inductor opposite coupled lines must have a  $d_{in} \geq w$ , in order to enable magnetic flux to pass through the hollow part. In addition, the spacing between the outer spiral inductor turn and any other surrounding metal should be at least  $5w$ . The width of the inductor should be as wide as the limit where the skin effect starts to be dominant. The wider the metal track, the higher the exhibited quality factor, because the resistance of the inductor decreases, while the inductance value remains constant. However, when the width is significant to the skin effect, the inductor resistance starts to increase. It was observed for spiral inductors operating from 1 to 3 GHz that the  $Q$  is optimum for a track width between 10 and 15  $\mu\text{m}$  [10].

Due to mutual coupling between the spiral metal tracks, the spacing between the lines of the inductor should be as close as possible. Large spacing causes a reduction in the mutual coupling, thus lowering the inductance value [5]. Another design factor to take into account is the spiral radius. As the radius increases, the metal area overlapping the substrate increases accordingly and the parasitic capacitance between the spiral and the substrate increases. This results in a reduction of the self-resonant frequency. The substrate losses are also susceptible to the area occupied by the coil. Limiting the area, the magnetic field associated with the coil penetrates less deeply into the substrate, thus reducing the substrate losses.

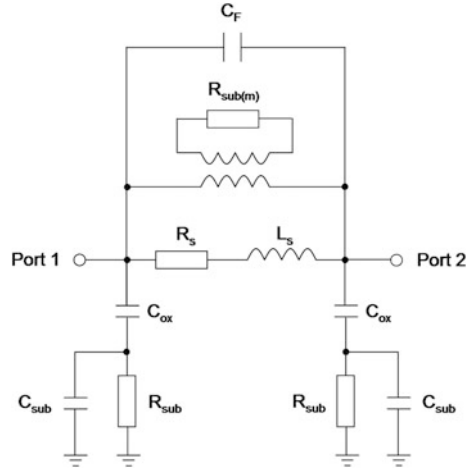
## 12.5 Modeling of Two-Port Inductors

A two-port lumped passive element  $\pi$ -type equivalent circuit, shown in Fig. 12.6, can be used to model a spiral inductor implemented on a silicon substrate. This equivalent circuit includes a number of components which altogether model the variation of the inductance with frequency and the loss mechanisms related to the structure of the spiral inductor. In particular,  $L_s$  represents the inductance,  $R_s$  models the resistance of the metal trace,  $C_F$  represents the capacitive coupling between the spiral trace and the underpass, and the magnetic eddy current effect is modeled as an ideal transformer coupled to a resistor  $R_{\text{sub}}(m)$ . In addition, the substrate is represented by three components  $C_{\text{sub}}$ ,  $R_{\text{sub}}$ , and  $C_{\text{ox}}$ , where  $C_{\text{ox}}$  is the oxide capacitance between the spiral and the substrate.

In order to estimate the value of these circuit elements, physically based equations related to the geometry of the spiral inductor and the parameters of the fabrication process can be used [11, 12]:

$$R_s = \frac{1}{\sigma \omega \delta (1 - e^{(-t/\delta)})} \quad (12.3)$$

**Fig. 12.6** General spiral inductor passive lumped element circuit model



$$C_F = n \omega^2 \frac{\epsilon_{ox}}{t_{ox} M1-M2} \tag{12.4}$$

$$C_{ox} = \frac{1}{2} l \omega \frac{\epsilon_{ox}}{t_{ox}} \tag{12.5}$$

$$C_{sub} = \frac{1}{2} l \omega C_{sub/A} \tag{12.6}$$

$$R_{sub} = \frac{2}{l \omega G_{sub/A}} \tag{12.7}$$

where  $\sigma$  is the conductivity of the metal layer,  $l$  is the total length of the metal trace,  $\delta$  is the metal skin depth,  $t$  is the metal thickness,  $t_{ox}$  is the thickness of the oxide situated between the spiral inductor and the substrate, and  $C_{sub/A}$  and  $G_{sub/A}$  are the substrate capacitance and conductance per unit area, respectively. The metal skin depth can be calculated using (12.8):

$$\delta = \sqrt{\frac{1}{\sigma \pi \mu f}} \tag{12.8}$$

where  $f$  is the frequency and  $\mu$  is the permeability of free space. The skin resistance  $R_s$  is given by Eq. (12.3), showing that as the frequency of operation increases, the resistance of a metal segment will increase due to the skin effect. The values of the quality factor  $Q$  and the inductance  $L_s$  can be calculated from the equivalent circuit by converting the measured or simulated two-port S-parameters into Y-parameters and using the equivalent  $\pi$ -network given in Fig. 12.7. For symmetrical inductors,  $Y_{12} = Y_{21}$  and  $Y_{11} = Y_{22}$ .

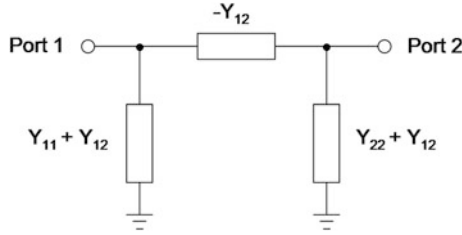


Fig. 12.7  $\pi$ -equivalent circuit for a two-port network

In order, to define  $L$  and  $Q$ , one needs to reduce the  $\pi$ -network to single element circuit consisting of an inductor in series with a resistor. For a simple series element  $R + jX$  (refer to Fig. 12.8), the inductance and the quality factor can be found using:

$$L = \frac{X}{2\pi f} \tag{12.9}$$

and

$$Q = \frac{X}{R} \tag{12.10}$$

Figure 12.8a shows the case in which one of the inductor terminals is grounded such that  $Y_{12} + Y_{22}$  is bypassed and the circuit looking into port 1 reduces to an admittance  $Y_{11}$  connected to ground.

In this case, the input impedance  $Z_{in}$  of the inductor can be calculated by:

$$R + jX = \frac{1}{Y_{11}} \tag{12.11}$$

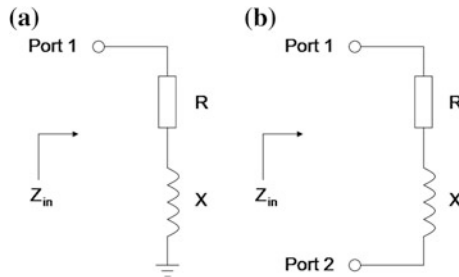


Fig. 12.8 Two methods of simplifying the two-port  $\pi$ -network. **a** Single-ended configuration and **b** differential configuration



Thus,  $L$  and  $Q$  is defined as follows:

$$L = \text{Im}\left(\frac{1/Y_{11}}{2\pi f}\right) = -\frac{1}{2\pi f \text{Im}(Y_{11})} \quad (12.12)$$

$$Q = \frac{\text{Im}\left(\frac{1}{Y_{11}}\right)}{\text{Re}\left(\frac{1}{Y_{11}}\right)} = -\frac{\text{Im}(Y_{11})}{\text{Re}(Y_{11})} \quad (12.13)$$

Equations (12.12) and (12.13) are valid for an inductor used in a circuit, in which one of its terminals is connected to ground. This is often the case in many RF circuits, such as in low-noise amplifiers and mixers where the inductors are used for degeneration or as a load.  $L$  and  $Q$  can be calculated by using measured or simulated one-port S-parameters with one terminal of the inductor grounded and converting the reflection coefficient into an input impedance. The series input impedance  $Z_{\text{in}}$  is given by:

$$R + jX = Z_{\text{in}} = Z_0 \frac{1 + \Gamma_1}{1 - \Gamma_1} \quad (12.14)$$

where  $\Gamma_1 = S_{11}$  and  $Z_0$  is the port characteristic impedance. In other applications, such as differential voltage-controlled oscillators, the on-chip inductors are used in a differential configuration, where both ports are not at a ground potential (refer to Fig. 12.8b). In this case, a different approach is required to determine  $Q$  and  $L$  and the input impedance is referred to as floating impedance seen between port 1 and port 2 of the  $\pi$ -network. Therefore, the differential input impedance is given by:

$$R + jX = \left(-\frac{1}{Y_{12}}\right) \parallel \left(\frac{1}{Y_{11} + Y_{12}} + \frac{1}{Y_{22} + Y_{12}}\right) = \frac{Y_{11} + Y_{22} + 2Y_{12}}{Y_{11}Y_{22} - Y_{12}^2} \quad (12.15)$$

In this case, where the shunt elements  $Y_{11} + Y_{12}$  and  $Y_{22} + Y_{12}$ , which are related to the substrate networks, can be neglected,  $L$  and  $Q$  can be calculated using:

$$L = \text{Im}\left(\frac{1/Y_{12}}{2\pi f}\right) = -\frac{1}{2\pi f \text{Im}(Y_{12})} \quad (12.16)$$

$$Q = \frac{\text{Im}\left(\frac{1}{Y_{12}}\right)}{\text{Re}\left(\frac{1}{Y_{12}}\right)} = -\frac{\text{Im}(Y_{12})}{\text{Re}(Y_{12})} \quad (12.17)$$

When the shunt elements  $Y_{11} + Y_{12}$  and  $Y_{22} + Y_{12}$  are not negligible, as in standard CMOS processes, the effective inductance  $L_{\text{diff}}$  and  $Q_{\text{diff}}$  are obtained using (12.19–12.19) [3].

$$R + jX = \frac{4}{Y_{11} + Y_{22} - Y_{12} - Y_{21}} \quad (12.18)$$

$$L_{\text{diff}} = \frac{\text{Im}\left(\frac{4}{Y_{11} + Y_{22} - Y_{12} - Y_{21}}\right)}{2\pi f} \quad (12.19)$$

$$Q_{\text{diff}} = -\frac{\text{Im}(Y_{11} + Y_{22} - Y_{12} - Y_{21})}{\text{Re}(Y_{11} + Y_{22} - Y_{12} - Y_{21})} \quad (12.20)$$

For symmetrical inductors,  $Y_{22}$  and  $Y_{21}$  are equal to  $Y_{11}$  and  $Y_{12}$ , respectively, such that Eqs. (12.21) and (12.22) are simplified as follows:

$$L_{\text{diff}} = \frac{\text{Im}\left(\frac{2}{Y_{11} - Y_{12}}\right)}{2\pi f} \quad (12.21)$$

$$Q_{\text{diff}} = -\frac{\text{Im}(Y_{11} - Y_{12})}{\text{Re}(Y_{11} - Y_{12})}. \quad (12.22)$$

## 12.6 Inductance Estimation

The inductance of a planar spiral inductor is a complex function which mainly depends on its geometry. An accurate estimation of the inductance can be made either by using expressions based on a numerical method or by using a field solver. There are two methods which may be used to calculate the inductance of a spiral using a closed-form equation. One of the basic methods is based on the self-inductance and the mutual coupling in single wires and is known as the greenhouse method. The other method relies on empirical equations applied for inductance calculations. A summary of comprehensive formulas is presented in [13], illustrating the tables for inductance estimation.

According to the greenhouse theory, the inductance of a square spiral inductor can be calculated by splitting up the different inductor sections into single wires. Then, the self-inductance of each wire is calculated and finally summed up. The self-inductance of a single wire with a rectangular cross section is given by the following equation [2]:

$$L_{\text{self}} = 2l \ln\left(\frac{2l}{w+t} + 0.5 + \frac{w+t}{3l}\right) \quad (12.23)$$

where  $L_{\text{self}}$  is the self-inductance in nH, the wire length  $l$  is in cm,  $w$  is the wire width in cm, and  $t$  is the wire thickness in cm. This equation is valid when the wire length is at least greater than twice the cross-sectional dimension. Additionally, to

calculate the overall inductance, the mutual inductance (positive or negative) between parallel lines is included. The mutual inductance between two parallel wires can be expressed as follows [14]:

$$M = 2lQ_m \quad (12.24)$$

where  $M$  is the mutual inductance in nH,  $l$  is the wire length in cm, and  $Q_m$  is the mutual inductance parameter which is calculated by (12.25):

$$Q_m = \ln \left( \frac{l}{\text{GMD}} + \sqrt{1 + \left( \frac{l}{\text{GMD}} \right)^2} \right) - \sqrt{1 + \left( \frac{\text{GMD}}{l} \right)^2} + \frac{\text{GMD}}{l} \quad (12.25)$$

where GMD is the geometric mean distance between the track center of the two wires and its exact value is given by:

$$\ln(\text{GMD}) = \ln(d) - \left[ \frac{1}{12 \left( \frac{d}{w} \right)^2} + \frac{1}{60 \left( \frac{d}{w} \right)^4} + \frac{1}{168 \left( \frac{d}{w} \right)^6} + \frac{1}{360 \left( \frac{d}{w} \right)^8} + \dots \right] \quad (12.26)$$

where  $d$  is the center to the center separation between the conductors and  $w$  is the width of the conductors. Thus, the inductance of a conductor is given by:

$$L_T = L_0 + M_+ - M_- \quad (12.27)$$

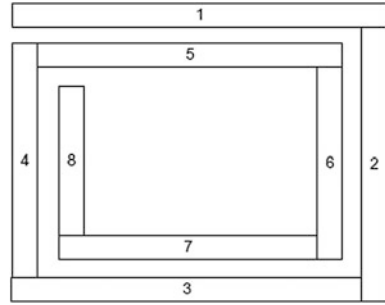
where  $L_T$  is the total inductance of the spiral inductor,  $L_0$  is the sum of self-inductances,  $M_+$  is the positive mutual inductance (where the current in two parallel segments is in the same direction), and  $M_-$  is the sum of the negative mutual inductance (where the current in two parallel wires is in the opposite direction) [2]. For instance, the inductance for a two-turn square spiral inductor shown in Fig. 12.9 can be calculated as follows:

$$\begin{aligned} L_T = & L_1 + L_2 + L_3 + L_4 + L_5 + L_6 + L_7 + L_8 \\ & + 2(M_{1,5} + M_{2,6} + M_{3,7} + M_{4,8}) \\ & - 2(M_{1,7} + M_{1,3} + M_{5,7} + M_{5,3} + M_{2,8} + M_{2,4} + M_{6,8} + M_{6,4}) \end{aligned} \quad (12.28)$$

where  $L_i$  is the self-inductance of wire  $i$  and  $M_{ij}$  is the mutual inductance between wires  $i$  and  $j$ .

The second method often used to estimate the inductance of a spiral coil is based on empirical equations. One such empirical equation is (12.29), which is based on the modified Wheeler formula [15] and is valid for planar spiral integrated inductors:

**Fig. 12.9** Application of the greenhouse method to a two-turn square spiral inductor



**Table 12.1** Coefficients for the modified Wheeler expression

Layout	$K_1$	$K_2$
Square	2.34	2.75
Hexagonal	2.33	3.82
Octagonal	2.25	3.55

$$L_{mw} = K_1 \mu_0 \frac{n^2 d_{avg}}{(1 + K_2 \rho)} \tag{12.29}$$

where  $L_{mw}$  is the inductance calculated by the modified Wheeler formula, the coefficients  $K_1$  and  $K_2$  are layout-dependent parameters presented in Table 12.1,  $n$  is the number of turns,  $d_{avg}$  is the average diameter defined as  $d_{avg} = 0.5(d_{in} + d_{out})$ , and  $\rho$  is the filling ratio defined as  $\rho = (d_{out} - d_{in}) / (d_{out} + d_{in})$ .

Another empirical expression is based on the current sheet approximation [15]. This method approximates the sides of the spirals by symmetrical current sheets of equivalent current densities. Since sheets with orthogonal current have zero mutual inductance, the inductance estimation is then reduced to just the evaluation of the self-inductance of a sheet and the mutual inductance between opposite current sheets. The self- and mutual inductances are established using the concepts of geometric mean distance (GMD), arithmetic mean distance (AMD), and arithmetic mean square distance (AMSD) [15]. The formula for this method is given by:

$$L_{gmd} = \frac{\mu n^2 d_{avg} c_1}{2} \ln((c_2/\rho) + c_3\rho + c_4\rho^2) \tag{12.30}$$

where  $c_i$  are layout-dependent coefficients provided in Table 12.2. As the ratio  $s/w$  increases, the accuracy of this equation degrades exhibiting a maximum error of 8 % for  $s \leq 3w$ . Practical integrated spirals are designed  $s \leq w$ .

**Table 12.2** Coefficients for the current sheet expression

Layout	$c_1$	$c_2$	$c_3$	$c_4$
Square	1.27	0.07	0.18	0.13
Hexagonal	1.09	2.23	0.00	0.17
Octagonal	1.07	2.29	0.00	0.19
Circle	1.00	2.46	0.00	0.20

**Table 12.3** Coefficients for the inductance monomial expression

Layout	$\beta$	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$
Square	$1.62 \times 10^{-3}$	-1.21	-0.147	2.40	1.78	-0.030
Hexagonal	$1.28 \times 10^{-3}$	-1.24	-0.174	2.47	1.77	-0.049
Octagonal	$1.33 \times 10^{-3}$	-1.21	-0.163	2.43	1.75	-0.049

The monomial expression is another empirical equation and it is based on a data-fitting technique which yields the following expression [15]:

$$L_{\text{mon}} = \beta d_{\text{out}}^{\alpha_1} w^{\alpha_2} d_{\text{avg}}^{\alpha_3} n^{\alpha_4} s^{\alpha_5} \quad (12.31)$$

where  $L_{\text{mon}}$  is the inductance in nH,  $d_{\text{out}}$  is the outer diameter in  $\mu\text{m}$ ,  $n$  is the number of turns, and  $s$  is the turn-to-turn spacing in  $\mu\text{m}$ . The coefficients  $\beta$  and  $\alpha_i$  are layout dependent and are given in Table 12.3. This expression can be solved using geometric programming which is an optimization method that applies monomial models.

Although the greenhouse method offers sufficient accuracy to estimate the inductance value [17], this method cannot provide a direct design for given specifications and it is a slow approach for a preliminary design. Additionally, simple inductor expressions may predict the correct order of magnitude of the inductance value, but they incur errors in the range of 20 % which is unacceptable for circuit design and optimization [17]. The three aforementioned empirical equations are accurate, with typical errors of 2–3 % [17]. Consequently, they present an excellent candidate for a design and synthesis tool. These equations can provide expressions for the inductance of square, hexagonal, octagonal and circular planar inductors.

Commercial 3D electromagnetic simulators can be used to estimate the inductance of planar spiral inductors, via the extracted  $Y$ -parameters of the two-port  $\pi$ -equivalent circuit model (refer to Sect. 12.5) using (12.32) [18]:

$$L_s = -\frac{1}{2\pi f} \text{Im}\left(\frac{1}{Y_{12}}\right) \quad (12.32)$$

where  $f$  is the frequency. The formulae used in the extraction of the inductor  $\pi$ -equivalent lumped circuit parameters are presented in [19]. The accuracy and limitations of such calculation are inherent to the inductor  $\pi$ -equivalent circuit model.

## 12.7 Boundary Conditions for the Spiral Inductor Optimization

The bounding of the layout parameters of the spiral inductor required for the optimization procedure can be expressed as follows [14]:

$$\begin{aligned}
& \text{maximize} && Q(d_{\text{out}}, w, s, n) \\
& \text{subject to} && L_{s,\min} \leq L_s(d_{\text{out}}, w, s, n) \leq L_{s,\max} \\
& && (2n + 1)(s + w) \leq d_{\text{out}} \\
& && d_{\text{out min}} \leq d_{\text{out}} \leq d_{\text{out max}} \\
& && w_{\min} \leq w \leq w_{\max} \\
& && s_{\min} \leq s \leq s_{\max} \\
& && n_{\min} \leq n \leq n_{\max}
\end{aligned} \tag{12.33}$$

where  $Q$ -factor is the objective function and  $d_{\text{out}}$ ,  $w$ ,  $s$ , and  $n$  are the optimization variables related to the spiral geometry, in which  $n$  is the number of turns,  $s$  is the track-to-track distance, and  $w$  is the track width. The domain of the design search space is determined by the lower and upper bounds of these variables. It is important to set these variables to restricted feasible values in order to reflect the limitations of the technology.

The geometry of the spiral inductor needs to be optimized in order to maximize its quality factor  $Q$  at a particular frequency. The inductance value is bounded by the first constraint. The boundary of the layout size is ensured by the second constraint. The other four constraints are the geometric constraints. Many optimization methods have been proposed to solve (12.33), such as the exhaustive enumeration, sequential quadratic programming (SQP), mesh adaptive direct search (MADS), genetic algorithm, and geometric programming (GP) [1]. Considering that the design parameters of the spiral inductor are independent from each other, it is important to constraint them together. The outer diameter can draw a correlation between  $n$ ,  $w$ , and  $s$  governed by (12.34):

$$d_{\text{out}} = d_{\text{in}} + 2nw + 2(n-1)s \tag{12.34}$$

where  $d_{\text{in}}$  is the inner diameter.

## 12.8 Optimization of Inductors via a Genetic Algorithm

A genetic algorithm (GA) optimization is a stochastic search method which replicates the natural biological evolution by applying the principle of survival of the fittest, in order to achieve the best possible solution to a given problem. In the context of integrated spiral inductor design, GA is being proposed as an adequate optimization tool since it does not rely on formal mathematical derivations or prior knowledge of the problem, is resistant to being trapped in local optima, and can handle noisy functions. In addition, GA has proven to be able to handle large variations within the boundary conditions and is able to search at specific point rather than at regions across the searched space [20].

The main principle behind the applied GA optimization is that it takes into consideration heuristic constraints regarding the inductor design. It offers a way to determine the various parameters of the inductor layout. Due to the technology and topology constraints, the layout parameters are inherently discrete and so discrete variable optimization techniques are used. In this chapter, two approaches are presented. In this section, the GA-based integrated inductor design is based on the lumped element two-port  $\pi$ -model (refer to Sect. 12.5) and the modified Wheeler formula given by (12.29) which is used to calculate the inductance value.

This approach can be implemented using the MATLAB GA toolbox in order to yield technology-feasible design parameters [21]. The design generated by this method can then be verified through an EM simulator. For the  $\pi$ -model inductor, the quality factor is defined as given by Eq. (12.22) and the evaluation of  $R_s$ ,  $R_{si} = R_{sub}$ ,  $C_s = C_F$ ,  $C_{ox}$  and  $C_{si} = C_{sub}$  can be obtained from Eqs. (12.3), (12.7), (12.4), (12.5), and (12.6), respectively. The shunt resistance  $R_p$  and capacitance  $C_p$  can be estimated by:

$$R_p = \frac{1}{\omega^2 C_{ox}^2 R_{si}} + \frac{R_{si}(C_{ox} + C_{si})^2}{C_{ox}^2} \quad (12.35)$$

$$C_p = C_{ox} \frac{1 + \omega^2(C_{ox} + C_{si}) + C_{si}R_{si}^2}{1 + \omega^2(C_{ox} + C_{si})^2 R_{si}^2} \quad (12.36)$$

The restricted technological constraints are defined as follows: minimum values for the track width  $w$ , track-to-track spacing  $s$ , and input diameter  $d_{in}$ . Moreover, the correlation between the layout parameters is considered as heuristic design rules for reducing the parasitic phenomena due to proximity effect [20] given by (12.37)

$$0.2 < d_{in}/d_{out} < 0.8, d_{in} > 5w \quad (12.37)$$

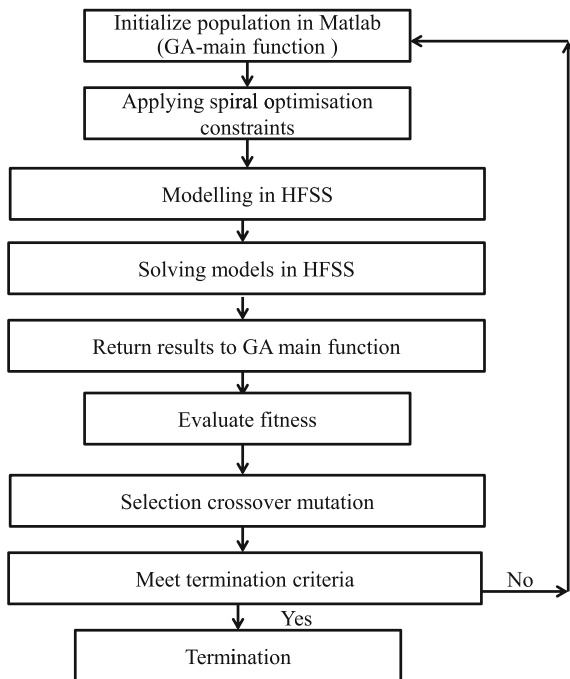
For a GA optimization procedure, a cost function is required in which it formulates the optimization problem as follows (12.38):

$$\begin{aligned} & \text{minimization of} && \text{Cost}(n, d_{in}, w) \\ & \text{subject to} && (1 - \delta)L_{exp} \leq L_s(d_{in}, w, n) \leq (1 + \delta)L_{exp} \\ & && w \in [w_{min} : step_w : w_{max}] \\ & && d_{in} \in [d_{min} : step_d : d_{max}] \\ & && n \in [n_{min} : step_n : n_{max}] \end{aligned} \quad (12.38)$$

where  $\text{Cost}(n, d_{in}, w)$  is the cost function,  $L_s(n, d_{in}, w)$  is the inductance of the spiral,  $L_{exp}$  is the targeted inductance value, and  $\delta$  is the tolerance limit for the inductance, which is the value by which it may deviate from the targeting value.

There are three different scenarios that can be applied to the cost function at a particular frequency of operation: either the minimization of the tolerance  $\delta$ , the minimization of the device area  $d_{out}$ , or else the maximization of the quality factor  $Q$ . In this work, the cost function is related to the maximization of the quality factor

**Fig. 12.10** Block diagram of the numerical GA used for the design optimization of the spiral inductor



[20]. Initially, the GA optimization algorithm randomly generates the initial population. Each individual constitutes three variables  $(w, d_{in}, n)$ , representing the layout geometry parameters. Each gene is formulated to real parameters, to abide to the objective boundaries' constraints. Following that, every quality factor and inductance of each particular gene (which refers to an inductor design) is evaluated. If these are not compliant, a fitness function is applied to pay a penalty so that it has a very low probability for being elected for the next population. If the termination condition is verified, the algorithm stops there, else the next steps create a new population, where selection and reproduction functions are used. For the selection, the roulette method is chosen, while afterward mutation is made. Figure 12.10 represents the flowchart of GA process to design the RFIC inductor.

To show the performance of the GA-based integrated inductor, an example of 1 nH square spiral inductor is shown. The technological parameters used to estimate  $R_{si}$ ,  $C_s$ ,  $C_{ox}$ , and  $C_{si}$  are shown in Table 12.4. The determination of the layout parameters is obtained through the constraints presented in Table 12.5. The GA optimization procedure was utilized to maximize the quality factor, given the tolerance for the required inductance. The result of GA optimization procedure is shown in Table 12.6.

The validity of the obtained design layout parameters was checked against a simulation performed using HFSS yielding the results shown in Table 12.7. The frequency response of the quality factor and inductance of the designed square



**Table 12.4** Technology parameters

Parameter	Value	Parameter	Value
$\epsilon_0$ (F/m)	8.85e-12	$t_{ox}$ ( $\mu\text{m}$ )	11.8
$\epsilon_r$	11.9	$C_{sub}$ (F/m <sup>2</sup> )	4.0e-6
$\sigma$ (S/m)	2.7e-7	$G_{sub}$ (S/m <sup>2</sup> )	2.43e-5

**Table 12.5** Design constraints

Parameter	Min	Max
$w_{in}$ ( $\mu\text{m}$ )	2	20
$d_{in}$ ( $\mu\text{m}$ )	70	90
$n$	2	7

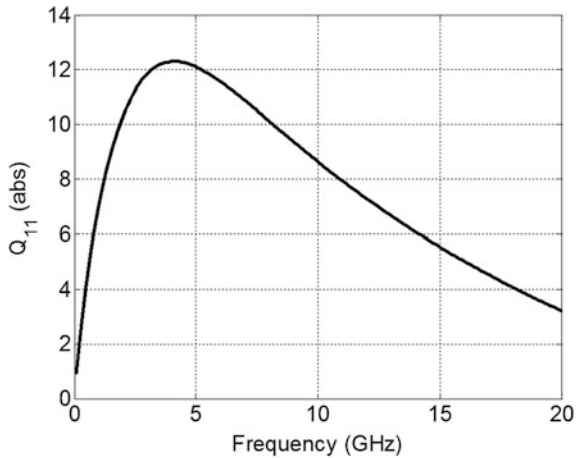
**Table 12.6** GA optimization results

$w_{in}$ ( $\mu\text{m}$ )	$d_{in}$ ( $\mu\text{m}$ )	$n$	$N$
15	48	2.5	4

**Table 12.7** Comparison of estimated and simulated results

$L_{GA}$ (nH)	$L_{HFSS}$ (nH)	Error (%)	$Q_{GA}$	$Q_{HFSS}$	Error (%)
1.2	1.15	4.2	6.8	7.2	5.8

**Fig. 12.11** Variation of the inductor’s quality factor with frequency obtained using HFSS



inductor are inductor is illustrated in Figs. 12.11 and 12.12, respectively. The inductor HFSS design model generated model is depicted in Fig. 12.13. The comparison between the HFSS simulation results and the GA estimations demonstrates a good agreement, where the GA inductance value is 1.2 nH and the simulations predict an inductance of 1.15 nH. The  $Q$  estimated via the GA is 6.8, while the simulations show that the inductor exhibits a  $Q$  of 7.2.

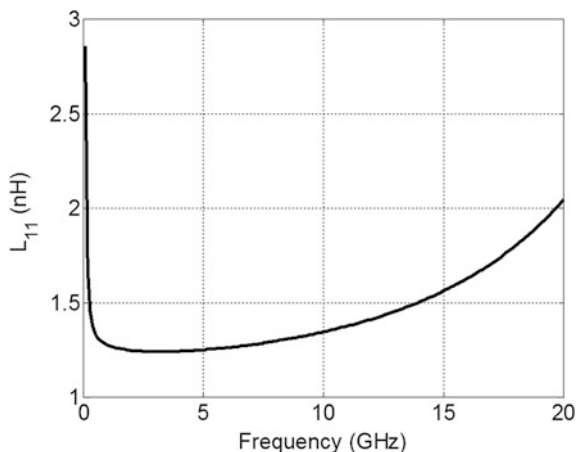


Fig. 12.12 Variation of the inductance with frequency obtained using HFSS

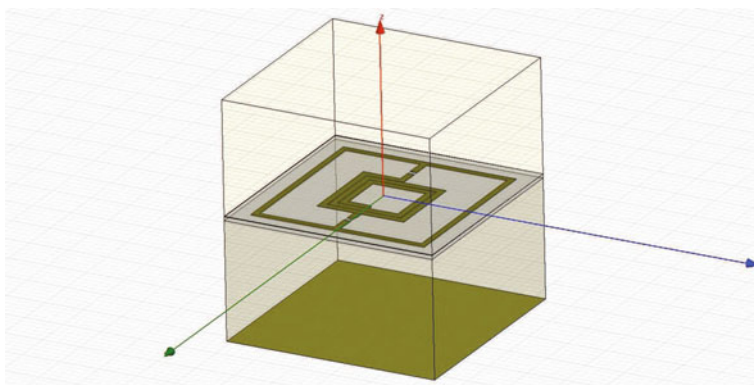


Fig. 12.13 HFSS square spiral inductor model

## 12.9 Optimization of Inductors via Geometric Programming

Geometric programming (GP) has a significant feature of determining if a design is feasible and if so finding the best possible inductor layout parameters [2]. Its main advantage is that it relates the sensitivity of the design objectives to its constraints, thus offering a rapid searching tool which enables the RFIC designer to spend more exploring and tuning the fundamental design trade-offs.

A GP problem has a form

$$\begin{aligned}
 & \text{minimize} && f_0(x) \\
 & \text{subject to} && f_i(x) \leq 1, i = 1, 2, \dots, m, \\
 & && g_i(x) = 1, i = 1, 2, \dots, p, \\
 & && x_i > 0, i = 1, 2, \dots, n,
 \end{aligned} \tag{12.39}$$

where  $f_i(x), i = 0, 1, \dots, m$ , are posynomial functions and  $g_i(x), i = 1, 2, \dots, P$ , are monomial functions. The posynomial function is defined as

$$f(x_1, \dots, x_n) = \sum_{k=1}^t c_k x_1^{\alpha_{1k}} x_2^{\alpha_{2k}} \dots x_n^{\alpha_{nk}} \tag{12.40}$$

where  $c_j \geq 0$  and  $\alpha_{ij} \in \mathbb{R}$ . When  $t = 1$ ,  $f$  is called a monomial function. Thus, for example,  $0.7 + 2x_1/x_3^2 + x_2^{0.3}$  is a posynomial and  $2.3(x_1/x_2)^{1.5}$  is a monomial. Posynomials are closed under sums, products, and nonnegative scaling.

Indeed, an initial point is unnecessary for it has no effect on the optimization algorithm procedure. The GP problem is solved globally and efficiently, converting it into a convex optimization problem. This is specifically done through the transformation of the objective and constraint functions using a set of new variables defined as  $y_i = \log x_i$ , such that  $x_i = e^{y_i}$  [22]. For a monomial function  $f$  given by (12.41)

$$f(x) = c_1 x_1^{\alpha_{1k}} x_2^{\alpha_{2k}} \dots x_n^{\alpha_{nk}} \tag{12.41}$$

Then,

$$\begin{aligned}
 f(x) &= f(e^{y_1}, \dots, e^{y_n}) \\
 &= c(e^{y_1})^{a_1} \dots (e^{y_n})^{a_n} \\
 &= e^{a^T(y+b)}
 \end{aligned} \tag{12.42}$$

where  $b = \log c$ .

Using the variable  $y_i = \log x_i$  transforms a monomial function to an exponential form of an affine function, as follows:

$$f(x) = \sum_{k=1}^K e^{a_k^T y + b_k} \tag{12.43}$$

where  $a_k = (a_{1k}, \dots, a_{nk})$  and  $b_k = \log c_k$ . Hence, a posynomial can be changed to a sum of exponentials of affine functions, and the GP problem is expressed in terms

of the new variable  $y$ . Then, the objective and constraint functions are transformed by taking the logarithm resulting in a convex optimization form

$$\begin{aligned} \text{minimize } & \bar{f}_0(y) = \log\left(\sum_{k=1}^{K_0} e^{a_{0k}^T y + b_{0k}}\right) \\ \text{subject to } & \bar{f}_i(y) = \log\left(\sum_{k=1}^{K_i} e^{a_{ik}^T y + b_{ik}}\right) \leq 0, \quad i = 1, \dots, m \\ & \bar{h}_i(y) = g_i^T y + h_i = 0, \quad i = 1, \dots, p. \end{aligned} \quad (12.44)$$

where the functions  $\bar{f}_0(y)$  are convex and  $\bar{h}_i$  are affine. Hence, this problem is referred to as geometric programming in convex form.

The formulation of spiral inductor optimization problem as a GP optimization problem was presented in [16], based on the monomial expression for inductance introduced in [15]. According to two-port lumped element circuit model, the monomial expression for the inductance is represented in terms of geometrical parameters ( $d_{\text{out}}$ ,  $w$ ,  $d_{\text{avg}}$ ,  $n$  and  $s$ ) [16], which has the form given by 31 [15].

Where the series resistance can be formulated as

$$R_s = \frac{l}{\sigma w \delta (1 - e^{-l/\delta})} = 4f(\omega)k_1 d_{\text{avg}} n/w \quad (12.45)$$

The spiral–substrate oxide capacitance  $C_{\text{ox}}$  that takes into consideration inductor's parasitic capacitance is given by the following monomial expression:

$$C_{\text{ox}} = \frac{\epsilon_{\text{ox}} l w}{2 t_{\text{ox}}} = 4k_2 d_{\text{avg}} n w \quad (12.46)$$

The series capacitance  $C_s$  that represents the capacitance between the spiral and the metal underpass required to connect the inner end of the spiral inductor to external circuitry. It is specified as a monomial expression

$$C_s = \frac{\epsilon_{\text{ox}} n w^2}{t_{\text{ox},M1-M2}} = k_3 n w^2 \quad (12.47)$$

where  $t_{\text{ox},M1-M2}$  is the oxide thickness between the spiral and the underpass.

The substrate capacitance  $C_{\text{si}}$  that refers to the substrate resistance can be modeled as a monomial equation

$$C_{\text{si}} = \frac{C_{\text{sub}/A} l w}{2} = 4k_4 d_{\text{avg}} n w \quad (12.48)$$

The monomial expression of the substrate resistance is  $R_{si}$

$$R_{si} = \frac{2}{G_{sub/A} l w} = k_5 / (4 d_{avg} n w) \quad (12.49)$$

where  $L_s$  is the inductance in nH,  $d_{out}$  is the outer diameter in  $\mu\text{m}$ ,  $n$  is the number of turns,  $s$  is the turn-to-turn spacing in  $\mu\text{m}$ ,  $k_1$  to  $k_5$  are coefficients dependent on technology, and  $f(\omega)$  is the coefficient dependent on frequency and technology

$$f(\omega) = \frac{1}{\sqrt{\frac{2}{\omega \mu_0 \sigma} (1 - e^{-t/\sqrt{2/(\omega \mu_0 \sigma)}})}} \quad (12.50)$$

The shunt resistance  $R_p$  and capacitance  $C_p$  are frequency dependent, expressed as monomials as follows:

$$R_p = \frac{1}{\omega^2 C_{ox}^2 R_{si}} + \frac{R_{si} (C_{ox} + C_{si})^2}{C_{ox}^2} = k_6 / (4 * d_{avg} n w) \quad (12.51)$$

$$C_p = C_{ox} \frac{1 + \omega^2 (C_{ox} + C_{si}) + C_{si} R_{si}^2}{1 + \omega^2 (C_{ox} + C_{si})^2 R_{si}^2} = 4k_7 d_{avg} n w \quad (12.52)$$

where  $k_6$  and  $k_7$  are coefficient dependent on technology and frequency. According to the  $\pi$ -model, the quality factor of a spiral inductor accounting for substrate loss factor and self-resonance factor is given by

$$Q_L = \frac{\omega L_s}{R_s} \cdot \frac{\overline{R_p} \left( 1 - \frac{R_s^2 \overline{C_{tot}}}{L_s} - \omega^2 L_s \overline{C_{tot}} \right)}{\overline{R_p} + \left[ \left( \frac{\omega L_s}{R_s} \right)^2 + 1 \right] R_s} \quad (12.53)$$

where  $\overline{R_p} = 2R_p$  and  $\overline{C_{tot}} = C_{tot}/2$  for two-port device, while for one-port device, it is  $\overline{R_p} = R_p$  and  $\overline{C_{tot}} = C_{tot}$ . When the inductor is used as one-port inductor, the total shunt capacitance is posynomial  $C_{tot} = C_s + C_p$  because  $C_s$  and  $C_p$  are monomial expressions. The quality factor represents the objective function in GP and can not be as a posynomial function of the design parameters. By introducing a new variable, the specification for minimum quality factor ( $Q_L \geq Q_{L,min}$ ) was written in [16] as a posynomial inequality in the design variables and  $Q_{L,min}$

$$\frac{Q_{L,min} R_s}{\omega L_s \overline{R_p}} \cdot \left[ \overline{R_p} + \frac{(\omega L_s)^2}{R_s} + R_s \right] + \frac{R_s^2 (C_s + C_p)}{L_s} + \omega^2 L_s (C_s + C_p) \leq 1 \quad (12.54)$$

This is because only inequality constraints in monomial form are allowed in GP. Accordingly, the GP design problem is formulated as

$$\begin{aligned}
 & \text{maximize} && Q_{\min} \\
 & \text{s.t.} && Q \geq Q_{\min} \\
 & && L = L_{\text{req}}, L_{s,\min} \leq L_s \leq L_{s,\max} \\
 & && (2n+1)(s+w) \leq d_{\text{out}} \\
 & && d_{\text{ang}} + n(s+w) \leq d_{\text{out}} \\
 & && d_{\text{out min}} \leq d_{\text{out}} \leq d_{\text{out max}} \\
 & && w_{\min} \leq w \leq w_{\max} \\
 & && s_{\min} \leq s \leq s_{\max} \\
 & && n_{\min} \leq n \leq n_{\max}
 \end{aligned} \tag{12.55}$$

Since the design parameters  $d_{\text{out}}$ ,  $w$  and  $s$  are independent, an inequality constraint to correlate them together  $d_{\text{ang}} + n(s+w) \leq d_{\text{out}}$  has been imposed. Also, the inductor area can be constrained by using the monomial inequality,  $d_{\text{out}}^2 \leq A_{\max}$ . The minimum self-resonant frequency can be handled by adding the following posynomial inequality:

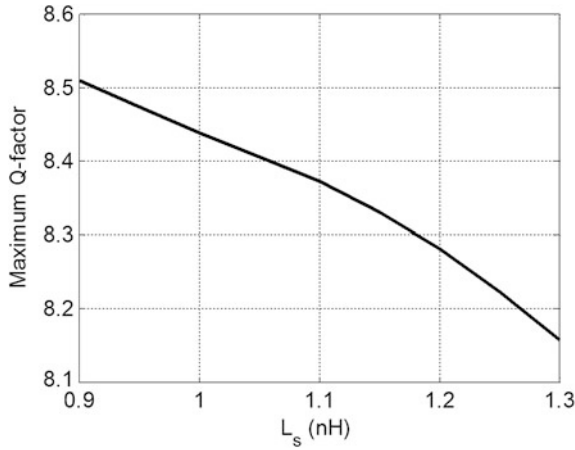
$$\omega_{\text{sr},\min}^2 L_s \overline{C_{\text{tot}}} + \frac{R_s^2 \overline{C_{\text{tot}}}}{L_s} \leq 1. \tag{12.56}$$

Yet, there are some cases that apply PGS beneath the inductor to eliminate the resistive and capacitive coupling to the substrate at the expense of the increased oxide capacitance. Hence, the inductor exhibits an improvement in its performance. In this case, the inductor lumped model parameters become  $R_p = \infty$ ,  $C_p = C_{\text{ox}} = (\epsilon_{\text{ox}} l w) / (2t_{\text{ox,po}})$ , where  $t_{\text{ox,po}}$  is the oxide thickness between the spiral and the polysilicon layer.

A simple MATLAB toolbox for solving geometric programming problems is proposed in [23]. This toolbox can be used to evaluate Eq. (12.55) and find feasible optimal parameters to model spiral inductors via geometric programming optimization method.

An optimal design of a 1-nH spiral inductor using the GP optimization is presented here, where the GP optimization tool maximizes the  $Q$ -factor for the inductor operating at 1 GHz. The GP tool was presented with the following constraints: Maximize  $\mu\text{m}$  subject to  $L_s = 1 \text{ nH}$ ,  $s \geq 2 \mu\text{m}$ ,  $\omega_{\text{sr}} \geq 10 \text{ GHz}$ .

Figure 12.14 illustrates the maximum  $Q$ -factor for 1-nH square inductor at 1 GHz without PSG, as a result of the GP optimization method. The corresponding geometrical dimensions are all in a feasible technological range, shown in Table 12.8. In order to verify GP results, commercial FEM simulation software of HFSS was used with the layout parameters depicted in Table 12.8. The results of HFSS verification are presented in Table 12.9, which show a very good agreement



**Fig. 12.14** Variation of the maximum quality factor with inductance

**Table 12.8** Maximum Q-factor and optimal value of geometry parameters for the 1-nH square inductor

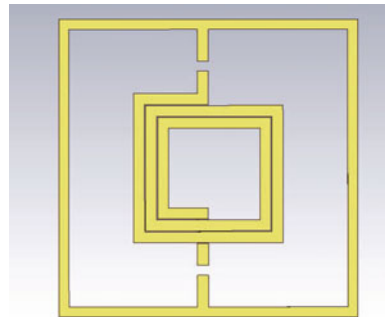
$L_s$ (nH)	$d_{out}$ ( $\mu\text{m}$ )	$w$ ( $\mu\text{m}$ )	$d_{avg}$ ( $\mu\text{m}$ )	$n$	$s$ ( $\mu\text{m}$ )
1	167.3	17	110	2.5	2

**Table 12.9** Comparison of the estimated and simulated results obtained using HFSS

$L_{GP}$ (nH)	$L_{HFSS}$ (nH)	$Q_{GP}$	$Q_{HFSS}$
1	1.1	6	7.3

with the GP estimated results. The GP algorithm gave an inductance of 1 nH with a  $Q$ -factor of 8.4, while HFSS reported that the designed inductor exhibits an inductance of 1.1 nH with a  $Q$ -factor of 7.3. The HFSS square spiral model is shown in Fig. 12.15.

**Fig. 12.15** Geometry of a square spiral inductor in HFSS



## 12.10 Genetic Algorithm Optimization Using EM Solvers

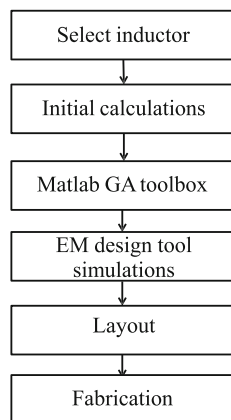
The implementation of integrated spiral elements relies on approximate quasi-static models that need to be verified by electromagnetic field solvers. The design of RF spiral inductors can be accomplished by integrating the use of a 3D electromagnetic (EM) solver together with an optimization method. A 3D EM solver is a CAD tool which can be used to compute multiport parameter data for a particular RF structure by using 3D electromagnetic field simulation. In this work, a new methodology of using the GA optimization MATLAB toolbox integrated with HFSS is presented, in order to demonstrate the implementation of an optimal RF CMOS inductor design. The proposed design procedure for the RFIC inductor is summarized in Fig. 12.16.

As discussed in Sect. 12.5,  $Q$  and  $L$  can be easily evaluated by simulating the inductor spiral and extracting the  $Y$ -parameters. However,  $Q$  is very sensitive to the simulation settings and environment. For an accurate determination of the  $Q$  value, the internal parts of the conductors should be finely meshed in order to account for the exponential decay of the current inside the conductors. The optimization boundary constraints employed in this approach are based on the set presented in (12.38), and a GA optimization is scripted so as to implement a spiral inductor. Using the extracted  $Y$ -parameter data,  $Q$  and  $L$  are estimated, and the results are automatically sent to the GA main function. A cost function is defined in order to eliminate genes with a low probability of achieving a maximum  $Q$  given by (12.57).

$$F(f) = \begin{cases} -Q, & \text{for } Q \geq 2 \\ 0, & \text{for } Q < 2 \end{cases} \quad (12.57)$$

To restrict the inductance value during the optimization procedure, a bounding condition is defined before calling the fitness function:

**Fig. 12.16** Design flow for an RFIC inductor





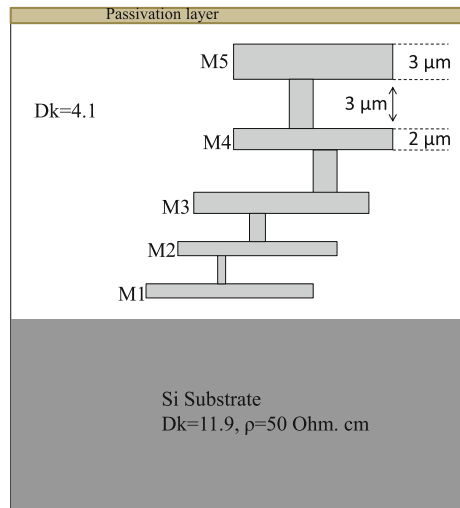
$$\begin{cases} \text{if } (1 - \delta) L_{\text{exp}} \leq L_s(d_{\text{in}}, w, N, n) \leq (1 + \delta) L_{\text{exp}} \\ Q_L = -\frac{\text{Im}(Y_{11}-Y_{12})}{\text{Re}(Y_{11}-Y_{12})} \\ \text{else} \\ Q = 0 \end{cases} \tag{12.58}$$

An optimum spiral inductor designed for a given inductance value at a particular operating frequency is targeted for a maximum  $Q$  and a minimum area consumption with an adequate self-resonant frequency. The physical characteristics of an inductor, such as the metal width  $w$ , outer diameter  $d_{\text{out}}$ , spacing  $s$ , and the number of turns  $n$ , are optimized in order to yield the required inductor. In addition, it was imperative to take into consideration the guidelines presented in Sect. 12.4. In practice, the values of on-chip inductors used in RF circuits fall in the range of 1–10 nH due to considerations in area utilization.

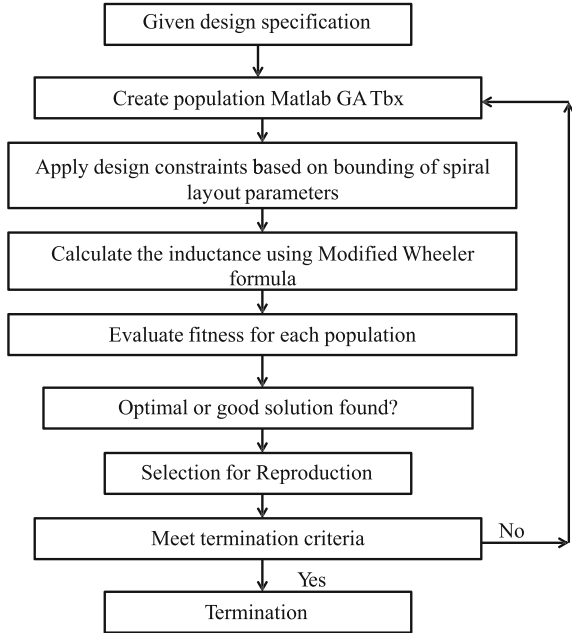
The CMOS process is modeled by drawing the substrate and the metal layers in a 3D-box-like fashion, where each layer is defined by its relative permittivity and bulk conductivity. The inductor layout is drawn by scripting HFSS commands through MATLAB using a library proposed in [24]. Figure 12.17 illustrates the main parameters of the generic CMOS process used in the simulations. The spiral is implemented using the top metal layer, and the underpass is made from the next metal layer level. A ground ring was added connecting each port of the inductor.

The block diagram of the genetic algorithm function used in this procedure is shown in Fig. 12.18. As a starting point of the optimization process, the initial population is created randomly, in which binary strings are generated from layout parameters. The GA is implemented in a way to code these layout parameters into genes via a binary-string coding. The four optimized parameters are  $s$ ,  $w$ ,  $n$ , and  $d_{\text{in}}$ , such that the chromosome structure is a four-part string, where each string

**Fig. 12.17** Layers in a generic CMOS process



**Fig. 12.18** Block diagram of the genetic algorithm used for the design optimization of the spiral inductor



corresponds to a parameter. The model is then created in HFSS according to the decoded parameters and is used to estimate the  $Y$ -parameters. The inductance is then evaluated using Eq. (12.21), while abiding to the condition given by (12.58). Following that, the algorithm automatically returns the  $Q$  value to the main function which applies the fitness function given by (12.57) to each individual in the GA population. Successive generations are produced by the application of selection, crossover, and mutation operators, until the optimal or a relatively optimal solution is found or termination criterion is met.

The 3D tool improves the design methodology of the on-chip inductors. Though it provides full freedom in implementation, it shows to be slower tool due to modeling through geometric construction and it uses the finite element method which requires many iterations in order to achieve convergence [18]. A relation was used to account for the accuracy in the quality factor estimated from the HFSS simulation results [25], where a cross-sectional solver was used to estimate the losses in coupled transmission lines, thus correcting the estimation.

The proposed optimization methodology is demonstrated through the design of a rectangular spiral inductor targeted for an operating frequency of 1 GHz. The design constraints and the technology parameters are given in Tables 12.10 and

**Table 12.10** Optimization constraints

Parameter	Values
Desired inductance	1 nH
Operating frequency	1 GHz
Outer diameter	$\leq 400 \mu\text{m}$

**Table 12.11** Technology parameters

Parameter	Values
Substrate resistivity	10 $\Omega$ cm
Silicon dielectric constant	11.9
Oxide thickness	4.5 $\mu$ m
Conductivity of the metal	$2.8 \times 10^5$ ( $\Omega$ cm) <sup>-1</sup>
Metal thickness	3 $\mu$ m

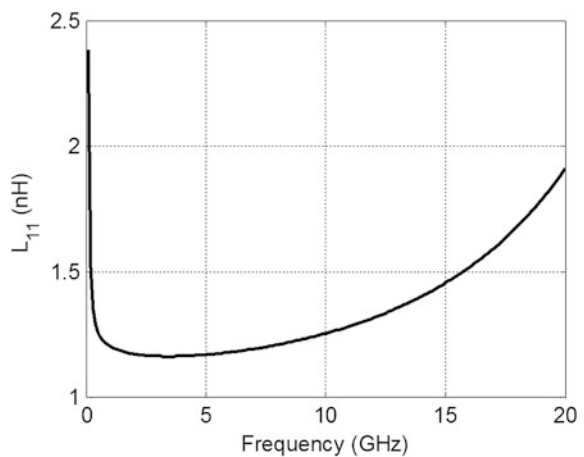
12.11. The determination of the upper and lower bounds of the width  $w$ , the number of turns  $n$ , and spacing  $s$  is based on an initial estimation of the inductance and layout parameters from the GP optimization; hence, a sweep is performed around these values. The GP optimization layout design parameters used in this example are those given in Table 12.8. The number of turns was varied from 2 to 4,  $w$  from 10 to 20  $\mu$ m, while  $d_{in}$  from 40 to 70  $\mu$ m.

The parameters of the square inductor design are given in Table 12.12, while the simulation results obtained from HFSS are illustrated in Figs. 12.19 and 12.20, where the variation of the inductance and quality factor with frequency is reported. The value of the quality factor and the inductance obtained from this procedure are compared with those obtained from the GP optimization procedure in Table 12.13.

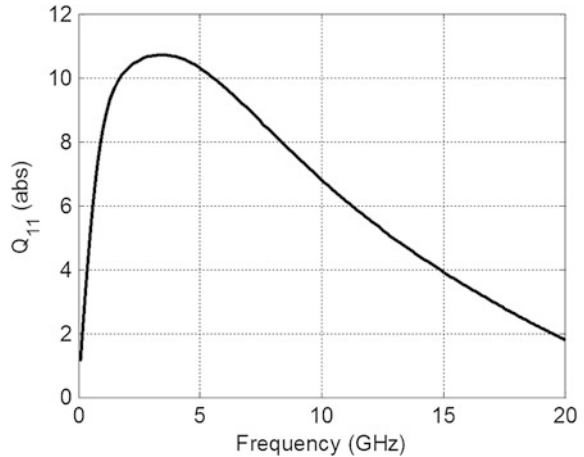
**Table 12.12** Optimization constraints

Parameter	Values
$w$	17 $\mu$ m
$s$	2 $\mu$ m
$n$	2.5
$d_{in}$	65 $\mu$ m

**Fig. 12.20** Variation of the inductance with frequency obtained using HFSS



**Fig. 12.19** Variation of the inductor's quality factor with frequency obtained using HFSS



**Table 12.13** GA optimization results

$L_{GA-HFSS}$ (nH)	$L_{GP}$ (nH)	$Q_{GA-HFSS}$	$Q_{GP}$
1.1	1	6.5	6

## 12.11 Conclusion

In this chapter, computational techniques employed to model and optimize radio frequency on-chip spiral inductors on a silicon substrate were presented and discussed. This work presents an efficient tool for analyzing, designing, and implementing any arbitrary inductor arrangement or topology. The optimization strategy is initialized by using a set of empirical formulae in order to estimate the physical parameters of the required structure as constrained by the technology, layout, and design specifications. Then, automated optimization using numerical techniques, such as genetic algorithms or geometric programming, is executed to further improve the performance of the inductor by means of dedicated software packages such as MATLAB. The optimization process takes into account substrate coupling, current constriction, and proximity effects. The results of such an optimization are then verified using a 3D EM simulator. This strategy was shown to be convenient in synthesizing optimal spiral inductors with adequate performance parameters such as the quality factor, area utilization, and self-resonant frequency, by combining lumped element model estimation with computational techniques within an EM simulation environment. This strategy provides a time-efficient and accurate design flow. A further improvement to this work would be to incorporate a method for correcting the inaccuracy of the EM simulators in calculating the quality factor of the spiral inductors.

## References

1. Yu, W., Bandler, J.W.: Modeling of spiral inductors. In: *Optimisation of Spiral Inductor on Silicon using Space Mapping*. Microwave Symposium Digest, 2006. IEEE MTT-S International, pp. 1085–1088 (2006)
2. Yu, W.: *Optimisation of spiral inductors and LC resonators exploiting space mapping technology*. Electrical and Computer Engineering. Hamilton, Ontario (2006)
3. Okada, K., Masu, K.: Modeling of spiral inductors. *Advanced microwave circuits and systems*. InTech. Institute of Technology, pp. 291–296 (2010)
4. Zhang, Y., Sapatnekar, S.S.: Optimization of integrated spiral inductors using sequential quadratic programming. In: *Proceeding of Design, Automation and Test in Europe Conference and Exhibition*, vol. 1. pp. 1–6 (2004)
5. Aguilera, J., Joaquim de No., Garcia-Alonso, A., Oehler, F., Hein, H., Sauerer, J.: A Guide for On-Chip Inductor Design in a Conventional CMOS Process for RF Applications. pp. 56–65 (2010)
6. Kaynak, M., Valenta, V., Schumacher, H., Tillack, B.: MEMS module integration into SiGe BiCMOS technology for embedded system applications. *Bipolar/BiCMOS Circuits and Technology Meeting (BCTM)*, 2012 IEEE. pp. 1–7 (2012)
7. Bunch, R.L., Sanderson, D.I., Raman, S.: Quality factor and inductance in differential IC implementations. *IEEE microwave magazine*. Appl. Notes. **3**(2), 82–91 (2002)
8. Yue, C.P., Wong, S.S.: On-chip spiral inductors with patterned ground shields for Si-based RF ICs. *Circ.* **33**(5), 743–752 (1998)
9. Lee, T.H.: *The Design of CMOS Radio-Frequency Integrated Circuits*. Cambridge University Press, Cambridge (1998)
10. Craninckx, J., et al.: A 1.8-GHz low-phase-noise CMOS VCO using optimized hollow spiral inductors. *IEEE J. Solid-State Circuits.* **32**, 736–744 (1997)
11. Yue, C.P., Ryu, C., Lau, J., Lee, T.H., Wong, S.S.: A physical model for planar spiral inductors on silicon. In: *Electron Devices Meeting, 1996. IEDM '96, International*, pp. 155–158 (1996)
12. Yue, C.P., Wong, S.S.: Physical modeling of spiral inductors on silicon. *IEEE Trans. Electron Dev.* **47**(3), 560–568 (2000)
13. Grover, F.W.: *Inductance calculations*. Van Nostrand, New York (1962)
14. Haobijam, G., Palathinkal, R.P.: Design and analysis of spiral inductors. In: Haobijam, G., Palathinkal, R.P. (eds.) *Optimisation of Spiral Inductor with Bounding of Layout Parameters*, pp. 21–51. Springer, New Delhi (2014)
15. Mohan, S.S., Hershenson, M., Boyd, S.P., Lee, T.H.: Simple accurate expression for planar spiral inductances. *IEEE J. Solid-State Circ.* **34**(10), 1419–1424 (1999)
16. Hershenson, M., Mohan, S.S., Boyd, S.P., Lee, T.H.: Optimisation of inductor circuit via geometric programming. In: *Proceedings of 36th Design Automation Conference*, pp. 994–998 (1999)
17. Crols, J., Kinget, P., Craninckx, J., Steyeart, M.: An analytical model of planar inductors on lowly doped silicon substrates for analog design up to 3 GHz. *VLSI Circuits, Dig Tech Papers*. pp. 28–29 (1996)
18. Paolo, G., Mayuga, T., Marc D. Rosales, M.D.: Inductor modeling using 3D EM design tool for RF CMOS process. *IEEE J. Solid-State Circuits.* **32**, 736–744 (1997)
19. Yoshitomi, S.: PAnalysis and simulation of spiral inductor fabricated on silicon substrate. *Electronics, Circuits and Systems, 2004. ICECS 2004. Proceedings of the 2004 11th IEEE International Conference on*. pp. 365–368 (2004)
20. Perdro, P., Helena Fino, M., Fernando, C., Mario Ventim-Neves: GADISI-genetic algorithms applied to the automatic design of integrated spiral inductors. In: *IFIP International Federation for Information Processing 2010*, vol. 314, pp. 515–522 (2010)
21. Haupt, R.L., Haupt, S.E.: *Practical Genetic Algorithms*. <http://onlinelibrary.wiley.com/book/10.1002/0471671746>. Cited May 2006

22. Boyd, S., Vandenberghe, L.: Convex Optimisation. Cambridge University Press, Cambridge (2004)
23. GGPLAB: A Simple Matlab Toolbox for Geometric Programming. <http://www.stanford.edu/boyd/ggplab/>. Cited May 2006
24. Vijay Ramasami: HFSS-MATLAB-SCRIPTING-API. <http://code.google.com/p/hfss-api/source/browse/trunk/?r=3>. Cited June 2009
25. Sani, A., Dunn, J., Veremey, V.: Using EM Planar simulator for estimating the Q of spiral inductors. AWR Corporation, a National Instruments Company
26. Greenhouse, H.M.: Design of planar rectangular microelectronic inductors. IEEE Trans. Parts Hybrids Package **10**(2), 101–109 (1974)

# Chapter 13

## Automated System-Level Design for Reliability: RF Front-End Application

Pietro Maris Ferreira, Jack Ou, Christophe Gaquière  
and Philippe Benabes

**Abstract** Reliability is an important issue for circuits in critical applications such as military, aerospace, energy, and biomedical engineering. With the rise in the failure rate in nanometer CMOS, reliability has become critical in recent years. Existing design methodologies consider classical criteria such as area, speed, and power consumption. They are often implemented using postsynthesis reliability analysis and simulation tools. This chapter proposes an automated system design for reliability methodology. While accounting for a circuit's reliability in the early design stages, the proposed methodology is capable of identifying an RF front-end optimal design considering reliability as a criterion.

### Acronyms

$G_{RF}$	Total gain
$F_{RF}$	Total noise
$IP3_{RF}$	Total linearity
$S_{11}$	Input matching
$V_{DD}$	Supply voltage
$f_{LO}$	Local oscillator frequency
$G_{LNA}$	LNA gain specification
$NF_{LNA}$	LNA noise specification

---

P.M. Ferreira (✉) · P. Benabes  
Department of Electronic Systems, GeePs, UMR CNRS 8507,  
CentraleSupélec - Campus Gif, Gif-sur-Yvette 91192, France  
e-mail: maris@ieee.org

P. Benabes  
e-mail: philippe.benabes@supelec.fr

J. Ou  
Department of Electrical and Computer Engineering, California State University Northridge,  
Northridge, CA 91330, USA  
e-mail: ou.jack@gmail.com

C. Gaquière  
IEMN, UMR CNRS 8520, Department of DHS, Lille-1 University, Lille, France  
e-mail: christophe.gaquiere@iemn.univ-lille1.fr

$IP3_{LNA}$	LNA linearity specification
$G_{PGA}$	PGA gain specification
$V_{nPGA}^2$	PGA noise specification
$IP3_{PGA}$	PGA linearity specification
$\Phi$	A general system-level specification where $\Phi \in [G_{RF}, F_{RF}, IP3_{RF}, S_{11}, V_{DD}, f_{LO}]$ at chapter's design example
$\psi$	A general building block characteristic where $\psi \in [G_{LNA}, NF_{LNA}, IP3_{LNA}, G_{PGA}, V_{nPGA}^2, IP3_{PGA}]$ at chapter's design example

### 13.1 Introduction

Reliability has become an important issue for critical applications such as military, aerospace, energy, and biomedical engineering since the nineties as introduced by Tu et al. [1] and Oshiro and Diego [2]. Facing a rise in failure rate in nanometer technologies, the ITRS report [3] identified reliability as a major challenge. Integrated circuits (ICs) are facing an increase in failure rate during product lifetime in nanometer CMOS technologies. In order to deal with reliability degradation, many analysis models and tools have been proposed by Maricau and Gielen [4]. Maricau et al. [5] defined reliability as the ability of a circuit to function in accordance with its specifications over its lifetime under stressful condition. Maricau and Gielen [4] propose the following guidelines for improving reliability:

1. Robust circuit using overdesign which attends specification even in worst case, or
2. Self-healing circuit able to reconfigure and compensate errors at run time by a digital control.

Both statements do not mitigate reliability, but consider its consequences after degradation event. Thus, designs are not optimized for reliability, and an overhead is required (e.g., redundancy). Moreover, the trade-off between high performance and reliability is not clear. Until now, the literature concentrates efforts in reliability estimation using physical equations and transistor-level models. This chapter presents why this reasoning cannot be applied to system-level synthesis.

Ferreira et al. [6, 7] have investigated new design considerations at transistor level. Ferreira et al. [8] have demonstrated that few changes on transistor sizing may increase the circuit lifetime. System-level analysis was highlighted at Ferreira et al. [9]. Cai et al. [10] have proposed a hierarchical reliability analysis associating transistor-level degradation with the system-level characteristic variation. Known design for reliability methodologies and implementations of reliability-aware AMS/RF performance optimization methods are detailed in a book chapter by Ferreira et al. [11].



This chapter proposes an innovative automated system-level design for reliability. Using reliability system-level modeling, five design experiments are presented to illustrate the proposition. An RF front-end architecture was chosen as an application example of the proposed automated system-level design.

The chapter is organized as follows. Section 13.2 introduces reliability degradation phenomena. Section 13.3 presents a brief description of the state of the art in circuit design methodologies undergoing in classical design and design for reliability. Section 13.4 proposes an automated method for system-level design for reliability describing in details the innovative design steps which allow a reliability control in early stages. Section 13.5 shows the implementation details of the proposed design automation on an RF front-end architecture example. Section 13.6 presents the experimental results of different runs comparing different design experiments. Each experiment follows a different optimization strategy reaching different component sizes and device characteristics. Finally, conclusions are drawn in Sect. 13.7.

## 13.2 Reliability Degradation Phenomena Background

### 13.2.1 Variability Phenomena

Undergoing variability, circuit performance shifts and may fail specifications. Variability phenomena are mainly random dopant fluctuations (RDF) and line edge roughness (LER). It can be divided into two categories:

- Systematic variations—repeatable electrical characteristic variations between two identical designed transistors and
- Random variations—statistical variations, classified as inter-die variations (also defined as global variation between lot-to-lot, wafer-to-wafer, and die-to-die) and intra-die (within-die or local) variations.

Variability phenomena are important during early circuit operation. Mutlu et al. [12] suggest inter-die variation as being much larger than intra-die (within-die) variations. Due to technology scaling, intra-die variations are growing in importance. Variations often modeled are as follows: transistor-level parameters, interconnections (width and length), and passive devices (resistivity and permittivity). To quantify variations at system level, building blocks are simulated separately using statistical Monte Carlo tools. Mean and variance are estimated from normal distribution of these devices. Details of physical sources of variations are lumped into a random statistical description.

### 13.2.2 *Environmental Phenomena*

Circuit performance and system performance are highly dependent on the working environment. Performance is subjected to variations in power supply voltage ( $V_{DD}$ ), temperature, workload dependency, and noise coupling from electromagnetic compatibility (EMC). Environmental phenomena depend on time, architecture topology, and external agents. Environmental effects are often called dynamic variations or temporal variations, because they impact performance during usage.

Typical  $V_{DD}$  and temperature fluctuations lead to transistor-level characteristic variation. Induced charges result in drain current and transconductance shifts. Commonly, process–voltage–temperature (PVT) variations are recognized by designers. Combining process variations and environmental variations, robust systems should be reconfigurable to dynamically adapt system’s behavior to its environment and workload. Furthermore, EMC has become a major cause of failure due to inadequate design methods in parasitic noise reduction and topology EMC immunity, presented in Ramdani et al. [13]. In mixed signal designs, digital switching noise degrades the ground reference voltage, causing a systematic  $V_{DD}$  variation. RF front end may also suffer from instability due to positive feedback or nonlinearity issues due to self-mixing as presented by Rosa et al. [14].

### 13.2.3 *Aging Phenomena*

During early circuit operation, variability phenomena are the most important reliability degradation phenomena. During lifetime usage, systems are susceptible to environmental effects, until wear-out and aging takes place. To reduce infant mortality due to degradation phenomena, ICs are intensively tested and subjected to degradation stress (burn-in).

Although nanometer technologies offer faster and smaller transistors, White and Chen [15] have commented that technology scaling results in devices more vulnerable to aging. According to this report, technology nodes as 45 and 28 nm are expected to have a lifetime lower than to 10 years. Consequently, system-level design has been including overdesigned margins to extend circuit lifetime. This choice is in contrast to a classical design optimization.

Much of reliability degradation is due to circuit aging under stressful environment. The main aging phenomena are as follows:

- Bias temperature instability (BTI)—refers to the generation of oxide charge and interface traps. BTI is affected by increased gate bias stress and mostly at elevated temperature. These traps can be partially recovered when bias and temperature are reduced, see Maricau and Gielen [4] for details.
- Hot carrier injection (HCI)—refers to high energy charges migration and accumulation of a high-density area of interface traps near the transistor drain, as defined by Maricau and Gielen [4]. HCI can be mitigated by biasing transistors

in moderate or weak inversion. However, this solution is not always feasible for gate lengths lower than 50 nm as circuit is designed for high-power or high-frequency application.

- Time-dependent dielectric breakdown (TDDB)—refers to a temporal stochastic oxide damage caused by defects generation inside gate dielectric. A cumulative failure probability of oxide breakdown can be predicted using the Weibull distribution, presented by Stathis [16]. Oxide breakdown may lead to a catastrophic failure (hard breakdown) or an ohmic path generation with an increased gate leakage current (soft breakdown).
- Electromigration (EM)—refers to metal erosion caused by excessive current density. EM phenomenon estimation is not available in early design stages. Design for manufacturing recommendations for EM mitigation are sizing widened wires in layout to do not fail current density rule check.

### 13.3 Circuit Design Methodologies

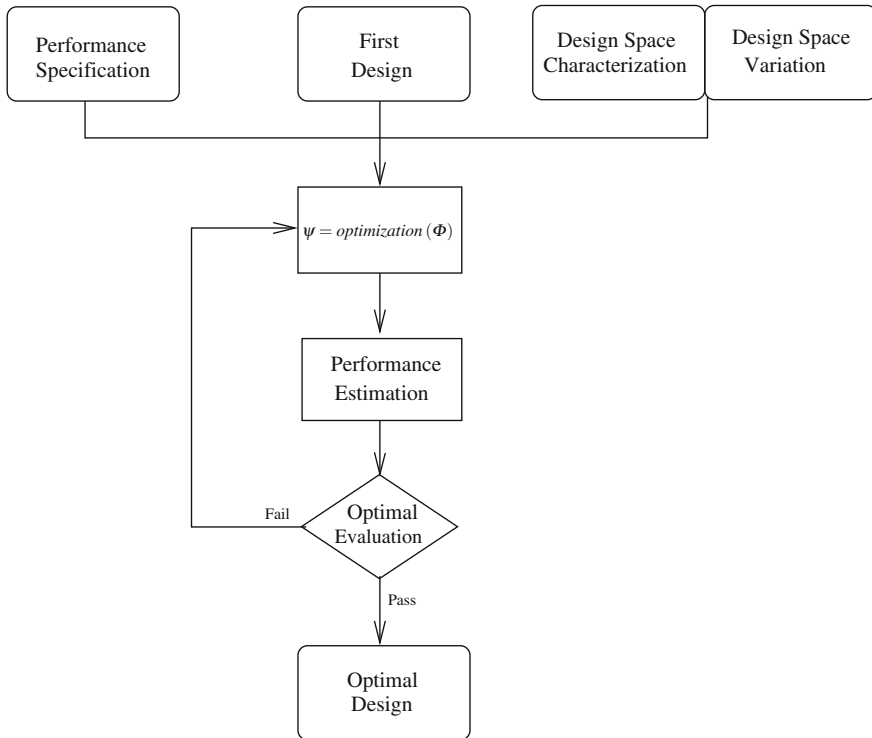
Classical methodology considers basic design criteria as die area, power consumption and speed, and sometimes noise and linearity (AMS/RF applications). Objectives of research in new design methodologies aim to:

- increase estimation accuracy;
- reduce convergence time;
- control the computational cost as the design complexity increases;
- propose new criteria trade-offs; and
- reuse the design experience to improve itself.

In spite of the importance of conventional design criteria, not considering reliability degradation is being negligent to performance variation. Cai et al. [10], and Maricau and Gielen [4] have demonstrated that variations might fail circuit specification in a shrinking lifetime. In the following sections, a brief description of design methodologies state of the art is presented undergoing in classical design and design for reliability.

#### 13.3.1 Classical Design Methodology

Classical design methodology relies on powerful numerical optimization coupled with accurate performance estimation tools. Design methodologies can be divided into three steps: architecture validation, schematic implementation, and layout synthesis. An automatic design methodology (see Fig. 13.1) starts by a *First Design*, which can be obtained using hand analysis, or a random draw based on the optimization method. Liu et al. [17] have presented some of these optimization



**Fig. 13.1** Classical design methodology: design steps illustration

methods. Design is conformed to a *Performance Specification* and placed in a *Design Space*. A *Design Space* is a set of device characterization previous obtained. In general, *Design Space* can be built using previous experience in designing similar blocs, or using the state of the art. In transistor level, a *Design Space* is a database establishing an association of different transistor's sizing and bias to transistor's transconductance and drain current. Under unreliability phenomena (described in Sect. 13.2), such device characteristics may suffer from degradation resulting in a *Design Space Variation*. Variations are not taken into account in a classical design methodology.

To determine whether a design is optimal, *Optimization* tool (see Fig. 13.1) searches the best set of characteristics according to a cost function and runs a *Performance Estimation*. If the design is not declared optimal by an *Optimal Evaluation*, the *Optimization* will be iterated using the previous solution to improve convergence. *Optimization* iteration depends on the cost function and the performance estimation model adopted. The methodology will converge if the found design conforms to *Performance Specification* and is in *Design Space* of characterized devices, becoming so an *Optimal Design*. Circuit design flow is finished by applying classical design methodology for system and circuit level.

The classical design methodology can be illustrated in transistor-level design. *Optimal Design* consists of sizing and biasing of all transistors and values of required passive components. Design parameters have to be determined in order to meet the circuit specifications at some trade-off cost. Using a *First Design* as start point, *Optimization* tool determines these design parameters belonging to *Design Space*. Then, a *Performance Estimation* tool, often a simulator, characterizes the optimal design candidate. Finally, *Optimal Evaluation* judges if the achieved performance is correct according to the *Performance Specification*. After some iterations and computational cost, *Optimal Design* may be found.

Tugui et al. [18] have presented a classical design methodology for system-level design. The problem considered by the auteurs is a 6th-order CT Sigma-Delta modulator design. Using Bayesian approach based on a Kriging probabilistic meta-model, a top-down design is carried out. System performance is optimized using Simulink/MATLAB. A simple transimpedance amplifier was considered as building block at transistor level. Transistor sizing is demonstrated. Coherent results at the system and transistor level were presented.

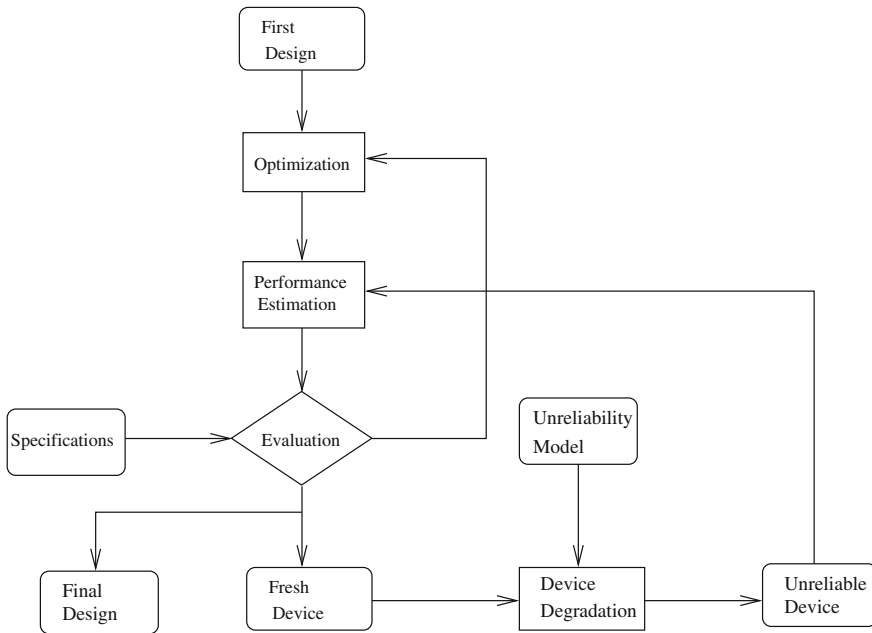
### 13.3.2 Design for Reliability Methodology

Design for reliability methodologies emulate the degradation process of reliability loss based on physical failure mechanisms. Design for reliability methodologies are inspired by Berkeley Reliability Tool (BERT) concept presented by Tu et al. [1]. Recent studies present reliability modeling as Huard et al. [19], or Parthasarathy et al. [20], and simulation tools as Ruberto et al. [21], or Quemerais et al. [22]. Others are interested in circuit lifetime estimation as Wunderle and Michel [23].

According to Li et al. [24], design for reliability methodology is based on reliability estimation using a simulation flow illustrated in Fig. 13.2. *Optimization* tool is required to obtain a non-stressed device (named here as fresh device). The device is stressed by degradation phenomena (described in Sect. 13.2) according to *Unreliability Model* to obtain a *Degraded Device*. Then, a *Performance Estimation* of the *Degraded Device* is required to verify if *Specifications* are met. If the *Evaluation* result is false, a new *Optimization* loop is required (see Fig. 13.2) until a *Final Design* is found.

In this method, a complex loop is used to simulate characteristics for both fresh and degraded devices. This step is computationally intensive and does not guarantee the choice of optimal and reliable design. Since optimization tool has no reliability insight, trade-off between optimal and reliable is not clear. Thus, the designer has no information to aim performance betterments during design for reliability iterations.

To address optimal and reliable device trade-off, recent publications study reliability issues using new statistical analyses. The main contribution in the literature is mapping device characteristics and characteristic variations. Both form a set of design variables for a feasible device, which is named in this work as *Design*



**Fig. 13.2** The state of the art of design for reliability methodology: unreliability phenomena simulation flow presented by Li et al. [24]

*Space.* Although advances made in design for reliability methodologies, reliability improvements remain in designing robust or self-healing circuits as Maricau and Gielen [4] have presented. These techniques have the drawback of circuitry redundancy and complexity. To the best of our knowledge, there is no automatic system-level reliability control implemented in a design methodology to reduce area and power consumption overhead due to reliability enhancements.

### 13.3.2.1 Reliability Analyses

Wang et al. [25] have introduced a simplified model for process variability, bridging the gap between Monte Carlo simulation and circuit design. Using this model, variability-aware design became possible since process variability could be estimated with a reduced computational cost.

Integrated-circuit reliability simulation is not a new concept, and a number of reliability models and simulation tools have been developed in Bernstein et al. [26]. Most of the studies are interested in modeling and simulating aging stress and circuit lifetime as Huard et al. [19] and Yuan and Tang [27]. They have highlighted stress environment and stress time as agents of variation in device characteristics. However, the aging degradation is not totally independent of the IC process variability. Before aging, the circuit suffers from the process variability which changes

the influence of the agents of stress (environment and time). Although both are combined in this physical event happening order, nominal reliability analysis tools published by Wang et al. [25], and Wunderle and Michel [23] do not take such order into account.

To solve this problem, Maricau and Gielen [28] have demonstrated a variability-aware reliability modeling and simulation tool. Moreover, Pan and Graeb [29] have proposed an efficient method to predict analog circuit reliability considering the joint effects of manufacturing process variations and parameter lifetime degradations. Therefore, the published works highlight tools capable of estimating the variation in circuit characteristics according to the variability and aging degradations.

### 13.3.2.2 Statistical Analyses

According to reliability degradation phenomena described in Sect. 13.2, the nominal characteristics cannot represent the IC performance. To ensure robust designs, statistical analyses have been widely used to describe the variation in circuit characteristics. Commonly used statistics methods are correlation analysis, regression analysis, design of experiments (DoEs), and response surface modeling (RSM).

Cai et al. [10] have highlighted correlation analysis as an interesting tool to filter correlated parameters reducing the number of design variables. Using semiconductor physics, the correlation analysis is able to identify uncorrelated design variables. Considering only these variables during design optimization, the complexity of reliability analysis is extremely reduced. Regression analysis aims to identify less significant parameters. Thus, some design variables are neglected in the estimation modeling because such variables are less significant in the performance estimation.

DoEs is an information-gathering process. During controlled experiments, a statistical analysis of circuit performance is carried out. Applying DoEs to IC design, designers are able to characterize the impact of the input design variables on the output circuit characteristics. RSM is another important statistical method that builds relationship between input design variables on the output circuit characteristics. Aside from DoEs, RSM directly maps design variables to circuit characteristics generating accurate functions to estimate the performance. Details about statistical analyses methods here introduced are available in Cai et al. [10], and Maricau and Gielen [4] works.

## 13.4 Automatic System-Level Design for Reliability

In this section, automatic system-level design for reliability methodology is discussed. The challenge is to consider system-level reliability without conceiving transistor-level reliability. In this way, changes in the optimization have to be made

to guide a classical design methodology (see Fig. 13.1) to an optimal and a reliable solution. Moreover, such a tool should avoid intensive reliability simulations with a high computational cost (see Fig. 13.2).

Cai et al. [10], and Maricau and Gielen [4] have presented analysis tools with reduced simulation effort using DoEs and RSM methods. Using these tools, transistor characteristics and their variations (by reliability degradation) have been identified and mapped in a database named design space. The design space correlates design variables (e.g., transistor sizing and bias) to circuit characteristics (e.g., gain and noise). These tools are important to build a design space of the system-level blocs as it is presented in this work. Also, failure conditions are obtained using the previous characterization which came from transistor-level knowledge presented in [7, 10, 11, 22, 29].

System-level design for reliability treats reliability issues in a top-down approach, leading to a feasible and reliable system in early stages. Previous investigations by Ferreira et al. [7, 9] have identified some common steps in the design flow aiming to control the circuit reliability and proposing it as a design criterion. To consider circuit reliability, the design flow should estimate circuit characteristic variations and compare with reliability requirements. Controlling the reliability, an automatic design for reliability flow has to guide an optimizer tool to a reliable solution.

Figure 13.3 proposes an automatic design for reliability diagram, which can be implemented to design reliable circuits and systems in bottom-up or top-down approach. The proposed method starts from the *Optimal Design* issued from classical design methodology (see Fig. 13.1). In the proposed method, *Optimal Design* is only a reliable solution candidate. If the design is reliable, the proposed methodology ends as a result of a pass in the *Failure Evaluation*. However, an *Optimal Design* hardly passes at first iteration since a classical design methodology does not use *Design Space Variation* information during *Optimization*. This means that circuit characteristic variation inside *Design Space Variation* is not considered in classical design methodology.

Using *Performance Estimation*, system characteristics  $\Phi_j$  are

$$\Phi_j = f(\psi_1, \dots, \psi_n), \quad \forall j \in [1, m], \quad (13.1)$$

where  $\psi_i$  are building block design variables ( $\psi_i \forall i \in [1, n]$ ). However, the  $\psi_i$  may change by  $\Delta\psi_i$  according to reliability degradation phenomena.  $\Delta\psi_i$  is a positive or a negative variation of  $\psi_i$  due to reliability degradation. Thus, the *Variation Estimation* has to evaluate

$$\Phi_{j_{\text{degraded}}} = f(\psi_1 + \Delta\psi_1, \dots, \psi_n + \Delta\psi_n). \quad (13.2)$$

After that, *Failure Evaluation* can compare the identified  $\Phi_{j_{\text{degraded}}}$  with the  $\Phi_{j_{\text{spec}}}$  and determine if the specification is still met after degradation. The *Variation Estimation* operates using a previous characterized set of design variables and circuit characteristics database named *Design Space* and *Design Space Variation*.



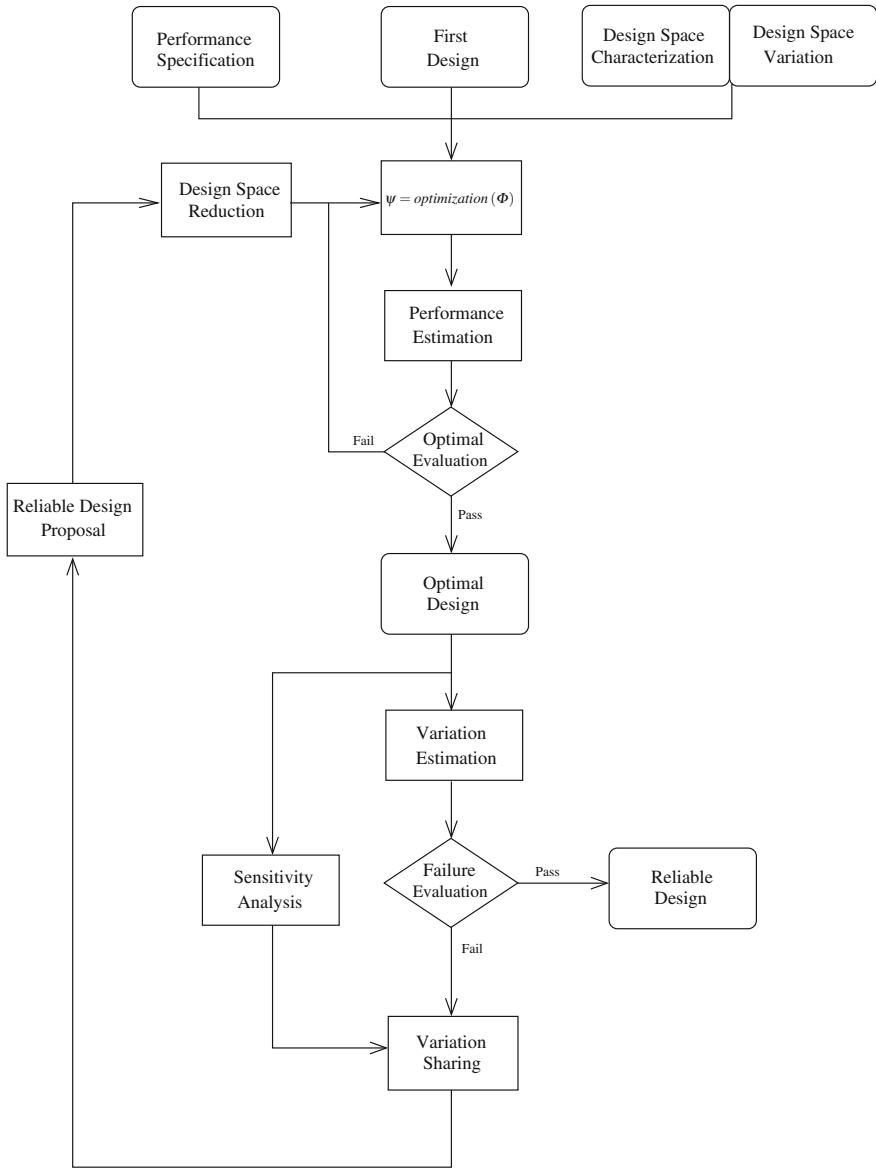


Fig. 13.3 The proposed automatic design for reliability: design steps illustration

Both information assembles mapped design variables, circuit characteristics, and failure conditions.

For transistor-level design, *Variation Estimation* is often done using Monte Carlo, Corner-based, and aging simulators. To speed up *Variation Estimation*, modeling is necessary and accuracy is often reduced. Reliability and statistical

analysis tools proposed in Cai et al. [10], Maricau and Gielen [4], and Pan and Graeb [29] works have presented a good trade-off in speedup and accuracy. However, these tools are hardly suitable for automatic design because they are simulation intensive. Moreover, these tools present serious limitations in order to guide the optimizer to a reliable solution in the following iteration (see Sect. 13.3.2).

In order to implement *Variation Estimation* suitable for automatic system design, this chapter proposes to reduce estimation accuracy using system-level design equations. It is assumed that circuit characteristics suffer from large variation due to unreliability phenomena. Thus, a failure condition is established by this hypothesis to meet early design stages and automatic analysis requirements. The variation of circuit characteristics is defined by the designer to state the circuit failure or lifetime. The amount of variation for failure condition is studied in publications of Cai et al. [10], Ferreira et al. [7], Pan and Graeb [29], and Quemerais et al. [22]. So that, Eq. 13.2 can be simplified to

$$\Phi_{j_{\text{degraded}}} = \Phi_{j_{\text{typ}}} + \Delta\Phi_j, \quad (13.3)$$

where  $\Delta\Phi_j$  (system characteristic variation) can be estimated using:

- Nominal reliability analysis

$$\Delta\Phi_j \approx \sum_{i=1}^n \Delta\Phi_{ij}, \text{ or} \quad (13.4)$$

- Variability-aware reliability analysis

$$\sigma_{\Phi_j}^2 = (\Delta\Phi_j)^2 \approx \sum_{i=1}^n (\Delta\Phi_{ij})^2. \quad (13.5)$$

In both analysis,  $\Delta\Phi_{ij}$  is the part of  $\Delta\Phi_j$  due to  $\Delta\psi_i$ , defined by

$$\Delta\Phi_{ij} = \left. \frac{\partial\Phi_j}{\partial\psi_i} \right|_{\psi} \Delta\psi_i. \quad (13.6)$$

Automatic methodology requires building block design variables ( $\psi_i \forall i \in [1, n]$ ) and system characteristics ( $\Phi_j \forall j \in [1, m]$ ) described in the database named *Design Space*. Moreover, methodology also requires that reliability degradation was described by  $\Delta\psi_i$  in *Design Space Variation*.

It is important to highlight that Eq. (13.6) assumes small variations of  $\Delta\psi_i$ . Despite isolated transistor-level characteristic varies much more than expected small variations, building block architecture can be applied in masking and reducing variations. For example, full differential balanced amplifiers are able to reduce transconductance variability, usually more than 30 % due to global

variability, in lesser than 1 %, due to differential pair mismatch only. Another example is temperature variation which has a minimal effect in building block characteristic if band-gap circuits supply a temperature-stable reference voltage. Aging variation is also masked and reduced in building block architecture; for example, Ferreira et al. [8] has presented in a DCO design having  $f_{osc} \leq 2$  %. If  $\Delta\Phi_{ij}$  cannot be modeled by a linear estimator (Eq. 13.6), a more complex estimator should be employed. However, it is hardly the case in system-level design.

According to the system specification  $[\Phi_j, \Delta\Phi_j] \forall j \in [1, m]$ , a reliable design will prefer building blocks having optimal characteristics ( $\psi_i \forall i \in [1, n]$ ) and a controlled reliability ( $\Delta\psi_i \forall i \in [1, n]$ ). Evaluating the proposed algorithm (see Fig. 13.3), the convergence will be found if guidelines of design for reliability can be provided to the optimization tool. Next, following sections describe how *Variation Sharing* and *Sensitivity Analysis* provide a *Reliable Design Proposal*. They also explain how *Design Space Reduction* can provide guidelines to *Optimization* in order to find a *Reliable Design*.

### 13.4.1 Variation Sharing

The *Variation Sharing* is an interactive decision process, which can be modeled and analyzed using a set of mathematical tools called Game Theory introduced by Osborne [30]. Game Theory proposes that decisions are made based on favoring some constraints in expense of others. Mostly applied to finance and economy research field, they intrinsically treat what is known as design trade-offs in circuit and systems. Thus, different set of decisions identifies a design strategy, which gives priority to some constraint. According to Osborne [30], the *Variation Sharing* decision process can be defined by three primary components, identified as:

1. a set of  $\psi_i \forall i \in [1, n]$ , varying due to reliability degradation;
2. a sharing strategy space composed by the  $\psi_i$  share priority defined by the set of positive sharing weights  $W_{ij}$ ;
3. a set of utility functions deciding how much is  $\Delta\psi_i \forall i \in [1, n]$  and defined by

$$\Delta\psi_i = \min_j |\Delta\psi_{ij}|, \quad (13.7)$$

where the variation allowed to  $\psi_i$  under the  $\Phi_j$  criterion is

$$\Delta\psi_{ij} = \frac{\Delta\Phi_{ij}}{\frac{\partial\Phi_j}{\partial\psi_i}}, \quad (13.8)$$

and  $\Delta\Phi_{ij}$  defined in Eq. (13.6) depends on an adopted design strategy.

In a general *Variation Sharing*, the decision strategy often chooses a characteristic  $i = k$  to give it the highest priority. Then, this strategy shares a bigger

variation to this characteristic  $\psi_k$  in detriment of a smaller variation to  $\psi_i \forall i \in [1, n - 1]$  for each  $\Phi_{ij}$ . Thus, the  $W_{ij}$  shall be defined in a way that the highest priority characteristic ( $\psi_k$ ) has

$$W_{kj} = \max_i (W_{ij}). \quad (13.9)$$

The influence of  $\Delta\psi_i$  on  $\Delta\Phi_j$  is calculated by the equation

$$\Delta\Phi_{ij} = \Delta\Phi_{kj} \frac{W_{ij}}{W_{kj}}. \quad (13.10)$$

At this point, shared variation is defined according to a  $W_{ij}$  priority. For each  $\psi_k$  favored by  $W_{ij}$ , a new design strategy is established. The set of design strategies is decisions that can be adopted in *Variation Sharing*. As a consequence, each favored block has relaxed constraints and variations are acceptable in a certain margin. Actually, a design margin is proposed to allow circuit characteristic variation if it is smaller than the specified margin. Thus, the design margin optimum can be found using Games Theory tool.

Still there are unanswered questions in *Variation Sharing* since reliability estimation may influence the decision process. In fact, the  $\Delta\Phi_{kj}$  is the  $\max_i |\Delta\Phi_{ij}|$  estimated from reliability degradation. If the design criterion is a **nominal reliability analysis**, then a *Linear* estimator can be employed by solving Eq. (13.4) as

$$\Delta\Phi_{kj} = \frac{\Delta\Phi_j}{\sum_{i=1}^n \frac{W_{ij}}{W_{kj}}}. \quad (13.11)$$

If the design criterion is a **variability-aware reliability analysis**, then a *Quadratic* estimator can be employed by solving Eq. (13.5) as

$$\Delta\Phi_{kj} = \sqrt{\frac{(\Delta\Phi_j)^2}{\sum_{i=1}^n \left| \frac{W_{ij}}{W_{kj}} \right|^2}}. \quad (13.12)$$

The best available  $W_{ij}$  strategy for any characteristic  $k \in [1, n]$  is the strategy that maximizes  $\Delta\psi_i$  under the belief that all  $n$  characteristics do the same as well. This set of best  $W_{ij}$  strategy forms an equilibrium as described in Games Theory introduced by Osborne [30]. Searching an equilibrium is the process proposed in design margin optimization. Hence, *Variation Sharing* decision at this equilibrium will always impose a smaller variation in  $\Phi_j$  than the first estimation of  $\Delta\Phi_j$  during failure evaluation. Moreover, this variation sharing decision leads to an optimal sharing by reducing the required design margin. The *Reliable Design Proposal* (see Fig. 13.3) will be found by specifying such a margin in the condition that the performance after variation is always better than the specification.

Nevertheless, defining the  $W_{ij}$  is not a simple task and may involve a lot of interaction among the lower level designers and the higher level designers, including powerful simulations, reliability, and statistical analyses. Actually,  $\Phi \in \Phi_j \forall j \in [1, m]$  are system-level characteristics such as die area, power consumption, gain, speed, bandwidth, noise, and linearity. Hence, this task will need a team effort finding all the  $W_{ij}$  which may be an over time-to-market solution. It suggests that the best response cannot be applied in automated design methodologies and early design stages.

In order to propose a better solution than an equal *Variation Sharing*, and without a strong design experience defining  $W_{ij}$ , a first-order sensitivity analysis is used in defining the variation-sharing strategy (see below). Actually, a sensitivity analysis gives us an accurate sharing weights if the variations are smaller than previously defined circuit failure. Few works describing circuit failure conditions are as follows: Cai et al. [10], Ferreira et al. [7], Pan and Graeb [29], and Quemerais et al. [22]. Also,  $\Phi_j$  modeling accuracy, defined in Eq. (13.1), is an important factor to guarantee an accurate sharing weights. It is expected that a sensitivity analysis will lead the design methodology to a good strategy without a time costly analysis.

### 13.4.2 Sensitivity Analysis

A higher level characteristic ( $\Phi_j$ ) sensitivity to a lower level characteristic ( $\psi_i$ ) is defined in first-order approximation as

$$S_{\Phi_j}^{\psi_i} = \frac{\psi_i}{\Phi_j} \frac{\partial \Phi_j}{\partial \psi_i}. \quad (13.13)$$

The  $S_{\Phi_j}^{\psi_i}$  is a measure of how much a  $\psi_i$  variation is able to change  $\Phi_j$ . For instance, a big  $S_{\Phi_j}^{\psi_i}$  implies in a significant  $\Delta \Phi_{ij}$  for a given  $\Delta \psi_i$ . Thus, the magnitude of the sensitive can be used in a *Variation Sharing* as a design strategy, defining the sharing weights ( $W_{ij}$ ).

From this point, this work proposes a  $W_{ij}$  definition to solve *Variation Sharing* by two different design strategies:

1. Giving priority to  $\Delta \Phi_{ij}$  with a *Lower*  $\left| S_{\Phi_j}^{\psi_i} \right|$ , so that

$$W_{ij} = \frac{\sum_{i=1}^n \left| S_{\Phi_j}^{\psi_i} \right|}{\left| S_{\Phi_j}^{\psi_i} \right|}, \quad \forall i \in [1, n] \text{ and } \forall j \in [1, m]; \text{ or} \quad (13.14)$$

2. Giving priority to  $\Delta\Phi_{ij}$  with a *Higher*  $|S_{\Phi_j}^{\psi_i}|$ , so that

$$W_{ij} = \frac{|S_{\Phi_j}^{\psi_i}|}{\sum_{i=1}^n |S_{\Phi_j}^{\psi_i}|}, \quad \forall i \in [1, n] \text{ and } \forall j \in [1, m]. \quad (13.15)$$

It is remarkable that using strategy *Lower* for  $W_{ij}$  defined by Eq. (13.14), trade-offs are assumed unlike using strategy *Higher* for  $W_{ij}$  defined by Eq. (13.15). Actually, each strategy will grant different margins to  $\psi_i$  variation privileging a few in disadvantages for others characteristics.

### 13.4.3 Reliable Design Proposal

A reliable design shall meet its specifications during its whole lifetime. Thus,  $\Phi_{J_{\text{degraded}}}$  should be better than  $\Phi_{J_{\text{spec}}}$  and from Eq. (13.3) concluding that

$$\Phi_{J_{\text{typ}}} = \Phi_{J_{\text{spec}}} \pm \Delta\Phi_j. \quad (13.16)$$

*Reliable Design Proposal* is changing  $\psi_i$  feedback, including a design margin estimated during previous iteration. Therefore, the proposed design methodology will optimize a circuit from such characteristics which will be reliable if the  $\Delta\Phi_j$  is not overcome in *Design Space*. In order to guarantee this assumption, the set of  $\psi_i \forall i \in [1, n]$  of optimal design has to be changed. *Design Space* have to include shared variations  $\Delta\psi_i \forall i \in [1, n]$ . This mean reduces the *Design Space* to fulfill the *Reliable Design Proposal* improving an *Optimization* convergence to a *Reliable Design*.

### 13.4.4 Design Space Reduction

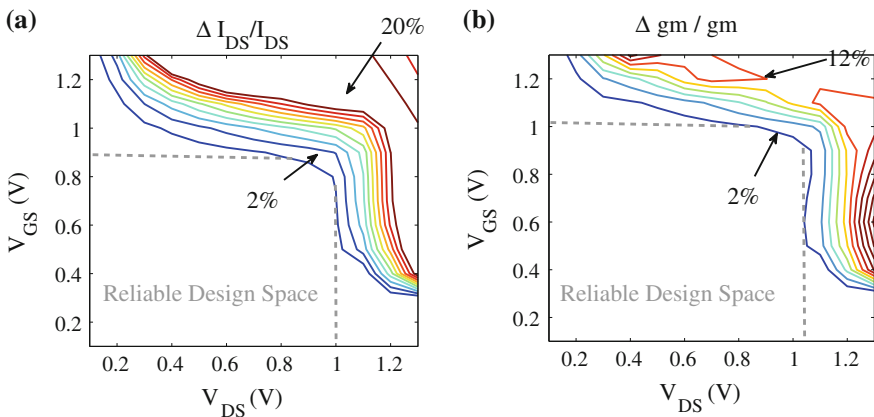
Including only the characteristic variations as design margins is not sufficient to guarantee the proposed design methodology convergence. Due to the complexity of *Design Space*, such space might present every *Optimization* iteration a new *Optimal Design* which is not reliable if the characteristic variation is bigger than the previous margin.

That is why, this chapter proposes a *Design Space Reduction* at each iteration for which a *Reliable Design* is not found. Then, a *Design Space* containing only a *Reliable Design Space* is given to the *Optimization* tool for following iteration. Knowing the relationship between  $\psi_i$  and  $\Delta\psi_i \forall i \in [1, n]$  in the *Design Space*, such space is reduced by cropping the regions where it is found a  $\Delta\psi_i$  bigger than the design margins obtained during the variation sharing.

Using a transistor-level design example presented in Ferreira et al. [7], the design variables  $\psi_i$  are often transistor bias, represented by  $V_{GS}$  and  $V_{DS}$ . An *Optimization* tool searches for an *Optimal Design* which meets  $\Phi_j$  transistor characteristics, represented by  $I_{DS}$  and gm in this example. Using any  $V_{GS}, V_{DS} \in [0, V_{DD}]$  as a *Design Space*, such an *Optimal Design* might result in a high  $\Delta\Phi_j$  due to a *Design Space Variation*, and thus the found solution is not a *Reliable Design*. However, strong inversion bias leads to high reliability degradation [7]. Figure 13.4 shows a high  $\Delta\Phi_j$  value for a bias close to  $V_{DD}$ , where transistors are in strong inversion. By a proper design strategy, the proposed methodology will find a new *Reliable Design Space* guiding an *Optimization* tool by *Design Space Reduction*. *Reliable Design Space* is represented by dashed lines in Fig. 13.4a, b, where a *Reliable Design Proposal* is met by a failure condition of 2% variation for  $\Delta I_{DS}/I_{DS}$  and  $\Delta gm/gm$ .

If it is not possible to have high performance and controlled reliability at the same time, *Design Space Reduction* may lead *Optimization* tool to an unfeasible design. By using a proper model (RF system-level equations in this work) and optimization tool (constrained optimization by linear approximation in this work) to detect the design feasibility, *Design Space Reduction* will produce the stop condition when *Optimization* is not able to converge. Thus, designers will know that imposed reliability and characteristics have a trade-off that cannot be feasible. In this way, they should relax reliability or characteristics specification. If designers are not able to do it, so they have to change the circuit topology or the IC technology.

The relationship between circuit topologies and reliability, or IC technologies and reliability is a prominent subject to be researched in order to improve our methodology convergence diagnosing unfeasibility. A conventional wisdom is to use an older IC technology which is naturally more reliable. Schematics with full differential and balanced structure are more reliable against variability degradation,



**Fig. 13.4** Normalized NMOS ( $W = 1 \mu\text{m}$  and  $L = 60 \text{ nm}$ ) simulation results for: **a**  $I_{DS}$  unreliability degradation and **b** gm unreliability degradation [7]

remaining only mismatch. Also, few changes on the schematics may be able to increase the circuit lifetime as introduced by Ferreira et al. [8]. Although many advances are made in reliability control, reliability enhancement remains intrinsically in designing robust or self-healing circuits with a payoff of circuitry redundancy and complexity according to Maricau and Gielen [4].

## 13.5 Automatic System-Level Design for Reliability Implementation

In order to demonstrate the proposed design for reliability methodology, a system-level RF front-end design was chosen. A design example is introduced in Sect. 13.5.1, and system-level model details are presented. Section 13.5.2 uses a system-level model to implement the algorithm illustrated in Fig. 13.3 using Python 2.7 programming language. Constrained optimization by linear approximation (COBYLA) from Python library is used.

### 13.5.1 System-Level Design Case

In order to demonstrate the proposed methodology, an automatic design of an RF front-end reliability control is implemented. The RF front-end architecture is presented in Sect. 13.5.1.1. The cost function used in optimization tool is presented in Sect. 13.5.1.2, and the performance estimation is presented in Sect. 13.5.1.3.

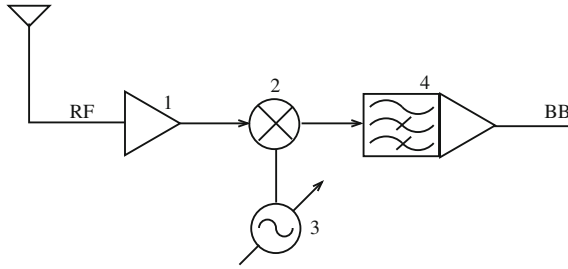
#### 13.5.1.1 RF Front-End Architecture

One of the most popular architectures for multi-standard wireless applications is the direct conversion RF front end, see Rosa et al. [14] for details. The architecture, illustrated in Fig. 13.5, has a low-noise amplifier (LNA—1), a passive mixer (MIXER—2), a voltage-controlled oscillator (VCO—3), and a programmable gain amplifier (PGA—4) with its baseband (BB) filter. In this design case, this chapter focuses on LNA and PGA design optimization, because together they are able to represent most of the design challenges in controlling the architecture reliability without complex models and optimization tool. Architecture specification is presented in Table 13.1 for WLAN/WiMAX direct conversion RF front end.

The design optimization is stated as follows:

- Given specification  $\Phi \in [G_{\text{RF}}, F_{\text{RF}}, \text{IP3}_{\text{RF}}, S_{11}]$ , as illustrated in Table 13.1.
- Find the building block characteristics





**Fig. 13.5** RF front-end architecture for multi-standard wireless applications: illustration

**Table 13.1** RF front-end architecture specifications for WLAN/WiMAX applications presented by Ferreira et al. [9]

$G_{RF}$	>30 dB
$20\log F_{RF}$	3.5 dB @ 1 GHz—6 dB @ 6 GHz
$IP3_{RF}$	>0 dBm
$S_{11}$	<-10 dB

$$\psi \in [G_{LNA}, NF_{LNA}, IP3_{LNA}, G_{PGA}, V_{nPGA}^2, IP3_{PGA}]$$

- Subjected to the constrained minimization of multivariate scalar functions introduced by Gu [31]

$$P_{\min} = \min \left( \sum_{i=1}^4 P_i \right), \text{ cost function being the power consumption} \quad (13.17)$$

$$G_{RF} = \prod_{i=1}^4 G_i, \text{ gain constraint} \quad (13.18)$$

$$F_{RF} = 1 + \sum_{i=1}^4 N_i, \text{ noise constraint} \quad (13.19)$$

$$P_{IP3_{RF}} = \frac{1}{\sum_{i=1}^4 \frac{1}{L_i}}, \text{ linearity constraint} \quad (13.20)$$

### 13.5.1.2 Cost Function

The architecture presented in Fig. 13.5 is modeled using RF system-level functions from Friis formulas detailed by Gu [31]. The optimization cost function chosen is the power consumption of the LNA added to the PGA. The cost function is

calculated by minimizing the power consumption in Eq. (13.17). The LNA and the PGA power consumption were modeled using a polynomial function to represent the trade-off among gain, linearity, and noise described by Xu et al. [32], Li et al. [33]. The power consumption equations are:

$$P_1 = P_{\text{LNA}} = k_1 \times 10^{G_{\text{LNA}}/20.0} \times 10^{\text{IP}_{3\text{LNA}}/10.0-3.0} \times \left(10^{\text{NF}_{\text{LNA}}/10.0} - 1\right), \text{ and} \quad (13.21)$$

$$P_4 = P_{\text{PGA}} = k_4 \times 10^{G_{\text{PGA}}/20.0} \times 10^{\text{IP}_{3\text{PGA}}/20.0} \times \frac{V_{N_{\text{PGA}}}^2}{\left(10^{G_{\text{LNA}}/20.0}\right)^2 \left(\frac{\pi}{4}\right)^2 V_{N_s}^2}. \quad (13.22)$$

The power consumption of the mixer (index 2 in Fig. 13.5) and of the local oscillator (index 3 in Fig. 13.5) was not included in the cost function because they are not included in the set  $\psi$  previously defined (see Sect. 13.5.1.1 for details).

### 13.5.1.3 Estimation Model

The architecture performance estimation is obtained by a model often employed in RF front-end architectures (also known as Friis formulas, see Gu [31]). For the architecture performance estimation, gain, noise, linearity, and input matching characteristics are evaluated. The model equations of the RF front-end architecture are as follows.

Gain: constraint defined in Eq. (13.18) is modeled with

$$G_1 = 10^{G_{\text{LNA}}/20.0}, \quad (13.23)$$

$$G_2 = G_{\text{MIXER}} = \frac{\pi}{4}, \quad (13.24)$$

$$G_3 = G_{\text{LO}} = \frac{\hat{V}_{\text{LO}}}{V_{\text{LO}}}, \quad (13.25)$$

where  $\hat{V}_{\text{LO}}$  is the estimated LO amplitude and  $V_{\text{LO}}$  is expected LO amplitude, and

$$G_4 = 10^{G_{\text{PGA}}/20.0}. \quad (13.26)$$

Noise: constraint defined in Eq. (13.19) is modeled with

$$N_1 = 10^{\text{NF}_{\text{LNA}}/10.0} - 1, \quad (13.27)$$

$$N_2 = F_{\text{MIXER}} - 1 = \frac{\pi}{4}, \text{ and} \quad (13.28)$$

$$N_4 = \frac{V_{n\text{PGA}}^2}{(10^{G_{\text{LNA}}/20.0})^2 \left(\frac{\pi}{4}\right)^2 V_{N_s}^2}, \text{ being} \quad (13.29)$$

$$V_{N_s}^2 = 4 kTR_s. \quad (13.30)$$

Linearity: constraint defined in Eq. (13.20) is modeled with

$$L_1 = 10^{\text{IP3}_{\text{LNA}}/10.0-3}, \text{ and} \quad (13.31)$$

$$L_4 = \frac{(10^{\text{IP3}_{\text{PGA}}/20.0})^2}{R_{\text{RF}}(10^{G_{\text{LNA}}/20.0})^2 \left(\frac{\pi}{4}\right)^2}. \quad (13.32)$$

The linearity from the mixer (index 2 in Fig. 13.5) and the local oscillator (index 3 in Fig. 13.5) were not included in the constraint because such blocks are not included in the set  $\psi$  previously defined (see Sect. 13.5.1.1 for details).

### 13.5.2 Automatic System-Level Design for Reliability Algorithm

The system-level design case presented in Sect. 13.5.1 is optimized to reliability control. For comparison purposes, the automatic design methodology is run for nominal reliability using *Linear* estimator in Eq. (13.4), variability-aware reliability (with *Quadratic* estimator presented in Eq. 13.5). According to sensitivity analyses, *Lower* and *Higher* priorities are applied for both estimators. Also, a classical design methodology (presented in Sect. 13.3.1) is implemented to demonstrate the trade-offs in controlling reliability for such an RF front-end architecture. This chapter presents four different experiments implementing reliability as a criterion and one extra experiment, named *Classical*, where reliability is neglected.

The automatic design for reliability methodology is implemented in Python 2.7 programming language using Algorithm 13.1. Optimization tool employed in this methodology is COBYLA from Python library. The implementation of the general methodology shown in Fig. 13.3 aims to control the reliability of the RF front-end architecture presented in Fig. 13.5 by using the model presented in Sect. 13.5.1.3. This tool should improve system-level design and optimization, including reliability as a design criterion.

---

**Algorithm 13.1** System-Level Design for Reliability Algorithm: RF front-end application.
 

---

```

 $\psi_0 = [G_{LNA}, N_{FLNA}, IP3_{LNA}, G_{PGA}, V_{N_{PGA}}^2, IP3_{PGA}]$ 
designSpace = loadDesignSpace()
specification = loadSpecification()
optimal = 0
while (not optimal) do
   $\psi_{opt}$  = optimization(evaluation.costFunction,
   $\psi_0$ , designSpace)
  optimal = 1
   $\Phi_{opt}$  = performanceEstimation( $\psi_{opt}$ )
  if (not evaluation.optimalDesign(specification,  $\Phi_{opt}$ ,  $\psi_{opt}$ )) then
    {Fail: Design is not optimal!}
    optimal = 0
     $\psi_0 = \psi_{opt}$ 
  else
    {
    emphPass: Design is optimal!}
     $\Delta\Phi$  = variationEstimation( $\psi_{opt}$ )
    if (not failureEvaluation.reliableDesign(specification,  $\Phi_{opt}$ ,  $\Delta\Phi$ )) then
      {Fail: Design is not reliable!}
      optimal = 0
       $\nabla\Phi$  = sensitivityAnalysis.gradientCalculation( $\psi_{opt}$ )
       $W$  = sensitivityAnalysis.strategyWeights( $\Phi_{opt}$ ,  $\psi_{opt}$ ,  $\nabla\Phi$ )
       $\Delta\psi$  = variationSharing( $\Delta\Phi$ ,  $W$ ,  $\nabla\Phi$ )
       $\psi_{reliable}$  = reliableDesign.solution( $\psi_{opt}$ ,  $\Delta\psi$ )
      designSpace = reliableDesign.spaceReduction( $\Delta\psi$ )
       $\psi_0 = \psi_{reliable}$ 
    else
      {Pass: Design is reliable!}
    end if
  end if
end while

```

---

## 13.6 Automatic Design for Reliability Results

The following experimental results are presented for the proposed methodology in this chapter:

- *Classic*, neglecting characteristic variation in a design optimization without reliability control (presented in Sect. 13.3.1);
- *Lower Quadratic*, combining variability-aware reliability analysis of Eq. (13.5) and  $W_{ij}$  priority of Eq. (13.14);

- *Lower Linear*, combining nominal reliability analysis of Eq. (13.4) and  $W_{ij}$  priority of Eq. (13.14);
- *Higher Quadratic*, combining variability-aware reliability analysis of Eq. (13.5) and  $W_{ij}$  priority of Eq. (13.15);
- *Higher Linear*, combining nominal reliability analysis of Eq. (13.4) and  $W_{ij}$  priority of Eq. (13.15).

The experiments have been executed on an AMD Turion X2 Dual-Core Mobile RM-74 2.2 GHz with 3 GB of RAM, and Windows 7 32-bits operating system. Table 13.2 summarizes the optimized characteristics for each building block. The execution time ( $t_{\text{exec}}$ ) of the algorithm was obtained using a mean time of 1k experiment runs avoiding CPU time measurement artifacts due to other tasks. The total power consumption ( $P_{\text{tot}}$ ) presented is the result of the cost function (see Eq. (13.17)) when the design is optimal. The RF front-end characteristics ( $G_{\text{LNA}}, \text{NF}_{\text{LNA}}, \text{IP3}_{\text{LNA}}, G_{\text{PGA}}, V_{\text{mpga}}^2, \text{IP3}_{\text{PGA}}$ ) are described in Sect. 13.5.1.

Table 13.3 shows the design margins in percentages. If no margin is imposed, Table 13.3 has identified such characteristic with *none*. In this case, an optimization of the margin was not done, and thus, there is no reliability control in such a case. In order to control reliability, *none* value means an unknown information about the variation of the related characteristics. Obviously, the classical methodology does not optimize design margins, leaving to the designer a lack of information. In *none* cases, circuit redundancy and overdesigned margins should be considered as a solution to guarantee reliability enhancement.

The results of Tables 13.2 and 13.3 highlight different  $t_{\text{exec}}$  among the experiments. There is an important increase of  $t_{\text{exec}}$ , while reliability control is considered. However, the  $t_{\text{exec}}$  variation among design for reliability experiments represents the difficulty in finding an optimal solution inside the specification and reliability trade-off.

Nominal reliability analysis assumes that circuit variability is negligible due to architecture characteristics. However, this is not often the case. *Quadratic* estimator should be used instead of the *Linear* estimator in order to take variability into account. In Table 13.3, *Lower* strategy imposes reliability constraints to the LNA, relaxing the PGA variation. Nevertheless, *Lower* strategy will lead to higher LNA's characteristic variation and a total lack of information for PGA's linearity variation. Thus, *Lower* strategy is not suitable, because such strategy cannot control the system reliability as required.

The *Higher Linear* strategy (using higher sensitivity priority and linear estimator) identified the lowest power consumption. The experiment has shown an optimal margin for  $\text{NF}_{\text{LNA}}$ , and  $\text{IP3}_{\text{LNA}}$  by the reduction of the overdesigned margins:  $V_{\text{mpga}}^2$  and  $\text{IP3}_{\text{PGA}}$ . However, such strategy does not take variability into account. Thus, the *Higher Quadratic* strategy is the most suitable for this design example. If the variability of the building blocks is negligible, then the *Higher Linear* strategy can be chosen.

**Table 13.2** RF front-end design for reliability experiments

Design strategy	$t_{\text{exec}}$ (s)	$P_{\text{tot}}$ (mW)	$G_{\text{LNA}}$ (dB)	$\text{NF}_{\text{LNA}}$ (dB)	$\text{IP}_{3\text{LNA}}$ (dBm)	$G_{\text{PGA}}$ (dB)	$V_{\text{mpGA}}^{2-}$ ( $V^2/\text{Hz}$ )	$\text{IP}_{3\text{PGA}}$ (dBm)
Classical	0.44	4.90	10.00	1.50	0.01	25.42	8.33e-19	22.77
Lower quadratic	2.62	4.10	18.03	1.50	1.00	27.38	8.83e-19	29.16
Lower linear	1.55	4.99	14.77	1.50	0.99	24.40	1.01e-18	29.88
Higher quadratic	3.83	4.42	22.40	1.50	0.42	33.24	8.37e-19	28.32
Higher Linear	1.07	3.06	15.74	1.50	0.48	24.32	8.90e-19	25.87

**Table 13.3** RF front-end design margins for reliability control experiments

Design strategy	$\frac{\Delta G_{\text{LNA}}}{G_{\text{LNA}}} (\%)$	$\frac{\Delta \text{NF}_{\text{LNA}}}{\text{NF}_{\text{LNA}}} (\%)$	$\frac{\Delta \text{IP}_{3\text{LNA}}}{\text{IP}_{3\text{LNA}}} (\%)$	$\frac{\Delta G_{\text{PGA}}}{G_{\text{PGA}}} (\%)$	$\frac{\Delta V_{\text{mpGA}}^{2-}}{V_{\text{mpGA}}^{2-}} (\%)$	$\frac{\Delta \text{IP}_{3\text{PGA}}}{\text{IP}_{3\text{PGA}}} (\%)$
Classical	None	None	None	None	None	None
Lower quadratic	3.24	0.55	0.03	9.98	12.98	None
Lower linear	2.16	0.37	0.02	7.06	8.65	None
Higher quadratic	1.73	1.73	3.56	14.13	1.73	3.56
Higher linear	1.88	1.88	3.45	7.06	1.88	3.45

The proposed design methodology convergence depends on the existence of an optimal and a reliable solution. The convergence cannot be achieved if no solution is found for such specifications and reliability requirements. In this work, an RF system-level model has been chosen to search the optimal solution. If the methodology does not achieve convergence, then some characteristic represents a hard constraint in design for reliability. This characteristic that represents a constraint can be identified by the system-level modeling presented in this chapter.

Using the failure evaluator to log the characteristic failure, methodology can diagnose the divergence. If a large number of iterations fail for the same characteristic, then a maximum iteration counter would be achieved, stopping the automatic design. Hence, the experiment log would reveal the divergence and can diagnose a characteristic representing a hard constraint.

Once a hard constraint is identified, the methodology convergence can be achieved if it is possible to relax such constraint or the reliability requirement. If designers are not able to relax one of them, the optimal circuit with a controlled reliability cannot be found for such architecture and technology. To identify reliable IC technologies or architectures, the propositions in this work will be able to improve such a featured subject by IC technologies' and architectures' reliability research.

## 13.7 Conclusion

This work proposed an innovative automated system-level design for reliability methodology. The proposed methodology is able to find an optimal solution to a RF front-end design while considering circuit reliability. Experimental results are presented for five different implementations of the automated design for reliability method. An RF front-end architecture illustrates the methodology in system level. The presented implementations are as follows: *Classic*, *Lower Quadratic*, *Lower Linear*, *Higher Quadratic*, and *Higher Linear*.

*Classic* implementation does not include unreliability phenomena in a design optimization. *Lower* implementations have reliability constraints for LNA, but do not include it for PGA. Nevertheless, these strategies led to a lower LNA's characteristic variation and a total lack of information about PGA's linearity variation. Without a proper reliability estimation, *Classic* and *Lower* implementations are not suitable in RF front-end architecture design presented in this chapter. *Higher Quadratic* experiment with an execution time of 3.83 s is the optimum solution since it accounts for variability and aging. *Higher Linear* experiment can also be chosen if the variability of the building blocks is negligible.

Finally, methodology convergence is studied. Having an optimal and reliable solution in the *Design Space* is a condition to achieve convergence. In order to guarantee this condition, the methodology uses a *Design Space* previous published by Ferreira et al. [9]. Moreover, proposed methodology has applied specific system-level modeling for RF front-end architecture, which is used to identify unfeasible designs. If there is no solution for specifications and reliability requirements, hard constraints are identified using a sensitivity analysis. This chapter has also highlighted the increasing importance of IC technologies' and architectures' reliability research as a featured subject favored by this work.

## References

1. Tu, R.H., Rosenbaum, E., Chan, W.Y., Li, C.C., Minami, E., Quader, K., Ko, P.K., Hu, C.: Berkeley reliability tools-BERT. Technical Report. Electronics Research Lab, Dec 1991 (1992)
2. Oshiro L, Diego S (1995) A Design Reliability Methodology for CMOS. In: Proc. of Int. Integrated Reliability Workshop, 619, pp. 34–39
3. ITRS (2011) Modeling and simulation. Technical Report, International Roadmap for Semiconductors
4. Maricau, E., Gielen, G.: Computer-aided analog circuit design for reliability in nanometer CMOS. *IEEE Trans. Emerg. Select Top. Circ. Syst.* **1**(1), 50–58 (2011)
5. Maricau, E., De Wit, P., Gielen, G.: An analytical model for hot carrier degradation in nanoscale CMOS suitable for the simulation of degradation in analog IC applications. *Microelectron. Reliab.* **48**(8–9), 1576–1580 (2008). doi:[10.1016/j.microrel.2008.06.016](https://doi.org/10.1016/j.microrel.2008.06.016)
6. Ferreira, P.M, Petit, H., Naviner, J.F.: AMS and RF design for reliability methodology. In: Proceedings of IEEE ISCAS, IEEE, pp. 3657–3660 (2010)

7. Ferreira, P.M., Petit, H., Naviner, J.F.: A new synthesis methodology for reliable RF front-end design. In: Proceedings of IEEE ISCAS, pp. 1–4 (2011)
8. Ferreira, P.M., Petit, H., Naviner, J.F.: A synthesis methodology for AMS/RF circuit reliability: application to a DCO design. *Microelectron. Reliab.* **51**(4), 765–772 (2010). doi:[10.1016/j.microrel.2010.11.002](https://doi.org/10.1016/j.microrel.2010.11.002)
9. Ferreira, P.M., Petit, H., Naviner, J.F.: WLAN/WiMAX RF front-end reliability analysis. In: IEEE Proceedings of EAMTA 2010, IEEE, pp. 46–49 (2010)
10. Cai, H., Petit, H., Naviner, J.F.: A hierarchical reliability simulation methodology for AMS integrated circuits and systems. *J. Low Power Electron.* **8**(5), 697–705 (2012). doi:[10.1166/jolpe.2012.1228](https://doi.org/10.1166/jolpe.2012.1228)
11. Ferreira, P.M., Cai, H., Naviner, L.: Reliability aware AMS/RF performance optimization. In: Fakhfakh, M., Tlelo-Cuautle, E., Fino, M.H.S. (eds.) *Performance Optimization Techniques in Analog, Mixed-Signal, and Radio-Frequency Circuit Design*. IGI-Global, Hershey (2014)
12. Mutlu, A.A., Rahman, M., Member, S.: Statistical methods for the estimation of process variation effects on circuit operation. *IEEE Trans. Electron. Packag. Manuf.* **28**(4), 364–375 (2005)
13. Ramdani, M., Sicard, E., Boyer, A., Dhia, S.B., Whalen, J.J., Hubing, T.H., Coenen, M., Wada, O.: The electromagnetic compatibility of integrated circuits—past, present, and future. *IEEE Trans. Electron. Packag. Manuf.* **51**(1), 78–100 (2009)
14. Rosa, J.M.D.L., Castro-López, R., Morgado, A., Becerra-Alvarez, E.C., Ro, R.D., Fernández, F.V., Pérez-Verdú, B.: Adaptive CMOS analog circuits for 4G mobile terminals—review and state-of-the-art survey. *Microelectron. J.* **40**(1), 156–176 (2009). doi:[10.1016/j.mejo.2008.07.001](https://doi.org/10.1016/j.mejo.2008.07.001)
15. White, M., Chen, Y.: Scaled CMOS technology reliability users guide. JPL Publication, 08–14 Mar 2008
16. Stathis, J.H.: Percolation models for gate oxide breakdown. *J. Appl. Phys.* **86**(10), 5757 (1999). doi:[10.1063/1.371590](https://doi.org/10.1063/1.371590)
17. Liu, B., Wang, Y., Yu, Z., Liu, L., Li, M., Wang, Z., Lu, J., Fernandez, F.: Analog circuit optimization system based on hybrid evolutionary algorithms. *Integr. VLSI J.* **42**(2), 137–148 (2009). doi:[10.1016/j.vlsi.2008.04.003](https://doi.org/10.1016/j.vlsi.2008.04.003)
18. Tugui, C., Benassi, R., Apostol, S., Benabes, P.: Efficient optimization methodology for CT functions based on a modified Bayesian Kriging Approach. In: IEEE Proceedings of International Conference on Electronics Circuits and Systems, pp. 456–459 (2012)
19. Huard, V., Parthasarathy, C.R., Bravaix, A., Hugel, T., Guérin, C., Vincent, E.: Design-in-reliability approach for NBTI and hot-carrier degradations in advanced nodes. *IEEE Trans. Dev. Mater. Reliab.* **7**(4), 558–570 (2007)
20. Parthasarathy, C.R., Bravaix, A., Guérin, C., Monnet, J.: Design-in reliability for 90–65 nm CMOS nodes submitted to hot-carriers and NBTI degradation. In: *Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation*, pp. 191–200. Springer, Berlin. doi:[10.1007/978-3-540-74442-9\\_19](https://doi.org/10.1007/978-3-540-74442-9_19) (2007)
21. Ruberto, M., Degani, O., Wail, S., Tendler, A., Fridman, A., Goltman, G.: A reliability-aware RF power amplifier design for CMOS radio chip integration. In: Proceedings of IEEE International Reliability Physics Symposium, Phoenix, pp. 536–540 (2008)
22. Quemerais, T., Moquillon, L., Huard, V., Fournier, J.M., Benech, P., Corrao, N.: DC hot carrier stress effect on CMOS 65 nm 60 GHz power amplifiers. In: Proceedings of IEEE Radio Frequency Integrated Circuits Symposium, vol. 31(9), pp. 351–354. doi:[10.1109/RFIC.2010.5477310](https://doi.org/10.1109/RFIC.2010.5477310) (2010)
23. Wunderle, B., Michel, B.: Lifetime modelling for microsystems integration: from nano to systems. *Microsyst. Technol.* **15**(6), 799–812 (2009). doi:[10.1007/s00542-009-0860-z](https://doi.org/10.1007/s00542-009-0860-z)
24. Li, X., Qin, J., Huang, B., Zhang, X., Bernstein, J.B., Member, S.: A new SPICE reliability simulation method for deep submicrometer CMOS VLSI circuits. *IEEE Trans. Dev. Mater. Reliab.* **6**(2), 247–257 (2006)



25. Wang, V., Agarwal, K., Nassif, S.R., Nowka, K.J., Markovic, D.: A simplified design model for random process variability. *IEEE Trans. Semicond. Manuf.* **22**(1), 12–21 (2008). doi:[10.1109/TSM.2008.2011630](https://doi.org/10.1109/TSM.2008.2011630)
26. Bernstein, J.B., Gurfinke, M., Li, X., Walters, J., Shapira, Y., Talmor, M.: Electronic circuit reliability modeling. *Microelectron Reliab* **46**(12), 1957–1979 (2006). doi:[10.1016/j.microrel.2005.12.004](https://doi.org/10.1016/j.microrel.2005.12.004)
27. Yuan, J., Tang, H.: CMOS RF design for reliability using adaptive gate-source biasing. *IEEE Trans. Electron. Dev.* **55**(9), 2348–2353 (2008). doi:[10.1109/TED.2008.928024](https://doi.org/10.1109/TED.2008.928024)
28. Maricau, E., Gielen, G.: Efficient variability-aware NBTI and hot carrier circuit reliability analysis. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **29**(12), 1884–1893 (2010)
29. Pan, X., Graeb, H.: Reliability analysis of analog circuits using quadratic lifetime worst-case distance prediction. In: *IEEE Proceedings of Custom Integrated Circuits*, pp. 1–4 (2010)
30. Osborne, M.J.: *An Introduction to Game Theory*. Oxford University Press, Oxford (2003). doi:[10.1360/99ws0111](https://doi.org/10.1360/99ws0111)
31. Gu, Q.: *RF System Design of Transceivers for Wireless Communications*. Springer, New York (2006)
32. Xu, Y., Hsiung, K.L., Li, X., Pileggi, L.T., Boyd, S.P.: Regular analog/RF integrated circuits design using optimization with recourse including ellipsoidal uncertainty. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **28**(5), 623–637 (2009). doi:[10.1109/TCAD.2009.2013996](https://doi.org/10.1109/TCAD.2009.2013996)
33. Li, X., Gopalakrishnan, P., Xu, Y., Pileggi, L.T.: Robust analog/RF circuit design with projection-based performance modeling. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **26**(1), 2–15 (2007). doi:[10.1109/TCAD.2006.882513](https://doi.org/10.1109/TCAD.2006.882513)

# Chapter 14

## The Backtracking Search for the Optimal Design of Low-Noise Amplifiers

Amel Garbaya, Mouna Kotti, Mourad Fakhfakh and Patrick Siarry

**Abstract** The backtracking search algorithm (BSA) was recently developed. It is an evolutionary algorithm for real-valued optimization problems. The main feature of BSA vis-à-vis other known evolutionary algorithms is that it has a single control parameter. It has also been shown that it has a better convergence behavior. In this chapter, the authors deal with the application of BSA to the optimal design of RF circuits, namely low-noise amplifiers. BSA performance, viz. robustness and speed, are checked against the widely used particle swarm optimization technique, and other published approaches. ADS simulation results are given to show the viability of the obtained results.

### 14.1 Introduction

Radio-frequency circuit (RF) design is a laborious strained and iterative cumbersome task that mainly relies on the experience of the skilled designers. The literature offers a plethora of papers dealing with techniques, approaches, and

---

A. Garbaya (✉) · M. Fakhfakh  
National School of Electronics and Telecommunications of Sfax, University of Sfax,  
Sfax, Tunisia  
e-mail: amelgarbaya11@gmail.com

M. Fakhfakh  
e-mail: fakhfakhmourad@gmail.com

M. Kotti  
High School of Sciences and Technologies of Hammam Sousse, University of Sousse,  
Sousse, Tunisia  
e-mail: kot.mouna@gmail.com

P. Siarry  
Laboratoire LiSSi (EA 3956), Université Paris-Est Créteil, Vitry-sur-Seine, Créteil, France  
e-mail: siarry@u-pec.fr

algorithms aimed at assisting the designer in such a cumbersome task, see for instance [22, 47].

Mathematical approaches have been used for alleviating the sizing task of such circuits, and it has already been proven that classical approaches are powerless vis-a-vis these NP-hard optimization problems [23].

Metaheuristics bid interesting and arguably efficient tools for overcoming impotence of the classical techniques. This can be briefly explained by the fact that due to the stochastic aspect of metaheuristics, ‘efficient’ sweeping of large dimension search spaces can be insured. Furthermore, metaheuristics allow dealing with many objective problems as well as constrained ones [15, 51, 52, 57].

Evolutionary metaheuristics have been used to deal with the optimal design of RF circuits, as well as analog circuits, and a large number of algorithms have been tested [2, 19, 24, 25, 29, 32, 40–44, 47, 50, 53, 54].

Swarm intelligence techniques (SI) have also been used, such as particle swarm optimization techniques (PSO) [20, 21, 38, 55, 56], ant colony optimization techniques (ACO) [3, 5], and bacterial foraging techniques (BFO) [10, 31]. SI metaheuristics are nowadays largely adopted for the resolution of similar optimization problems. Actually, it has been shown that when compared to notorious optimization algorithms, mainly genetic algorithms (GA) [26, 33] and simulated annealing (SA) [35], SI techniques can be much interesting to be used because they can be more robust, faster, and require much less tuning of control parameters, see for instance [48].

Very recently, an evolutionary algorithm’s enhanced version has been proposed, and it is called the backtracking search optimization technique (BSA or BSOA), and it has been shown via mathematical test functions and few engineering problems that BSA offers superior qualities [11].

Thus, in this work, we have put BSA to the test. It was used for the optimal sizing of low-noise amplifiers (LNAs), namely an UMTS LNA and a multistandard LNA.

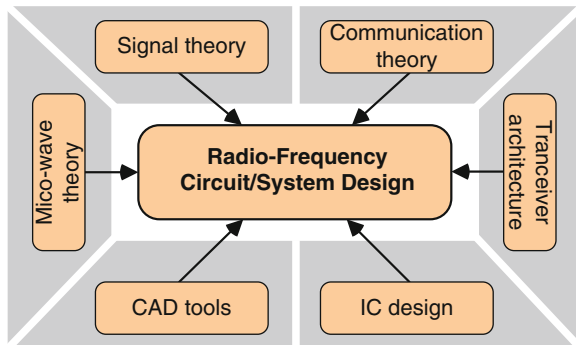
BSA performances were checked with those obtained using conventional PSO algorithm and also with published results (for the same circuits) using ACO and BA-ACO techniques [42] as it is highlighted in the following sections.

The rest of this chapter is structured as follows. In Sect. 14.2, we offer a brief introduction to the considered RF circuits. In Sect. 14.3, the BSA technique is detailed, and a concise overview of the PSO technique is recalled. Section 14.4 presents the BSA obtained results, which provides a comparison with performances from the other techniques. ADS simulation techniques are also given in this section. Finally, Sect. 14.5 concludes this chapter and discusses the reached results.

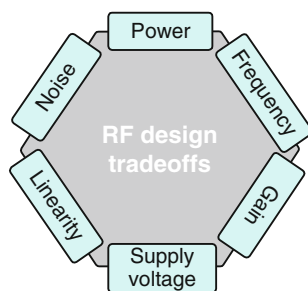
## 14.2 Low-Noise Amplifiers

Despite the tremendous efforts on RF circuit design automation, this realm remains very challenging. This is due to the complexity of the domain and its high interaction and dependency on other disciplines, as depicted in Fig. 14.1 [45].

**Fig. 14.1** Pictorial illustration of some disciplines involved in the RF design process



**Fig. 14.2** RF design tradeoffs



It is to be stressed that one among the most different tasks in this design is the handling of various tradeoffs, known by the famous hexagon introduced in [45], see Fig. 14.2.

The most important block of a front-end receiver is arguably the low-noise amplifier, which principal role consists in amplifying the weak RF input signal fed from the external antenna with a sufficient gain, while adding as less noise as possible, hence its name [1].

Advances in CMOS technology have resulted in deep submicron transistors with high transit frequencies. Such advances have already been investigated for the design of CMOS RF circuits, particularly LNAs [39].

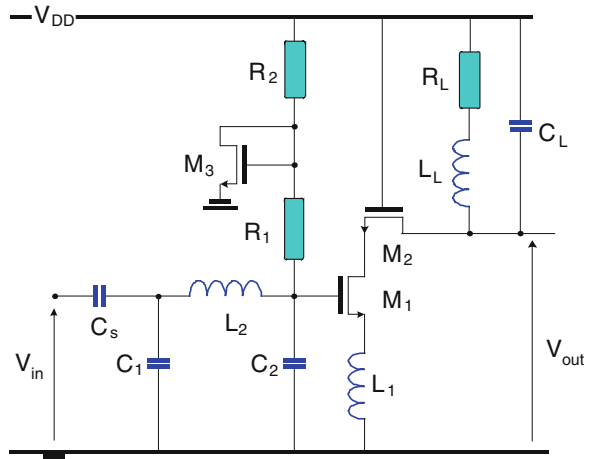
In this work, we deal with two CMOS LNAs, namely a wideband LNA and a multistandard LNA. Both architectures are chosen for comparison reasons with an already published paper [4] regarding performance optimization, as it is detailed in Sect. 14.4.

- A multistandard LNA

The CMOS transistor level schematic of the LNA is shown in Fig. 14.3. It is intended for multistandard applications in the frequency range 1.5–2.5 GHz [8].

In short, this LNA encompasses a cascade architecture for reducing the Miller effect and uses the reverse isolation.  $M_3$ ,  $R_2$ , and  $R_1$  for the biasing circuitry of the input transistor;  $L_2$ ,  $C_1$ , and  $C_2$  allow the input matching.

**Fig. 14.3** A multistandard CMOS LNA

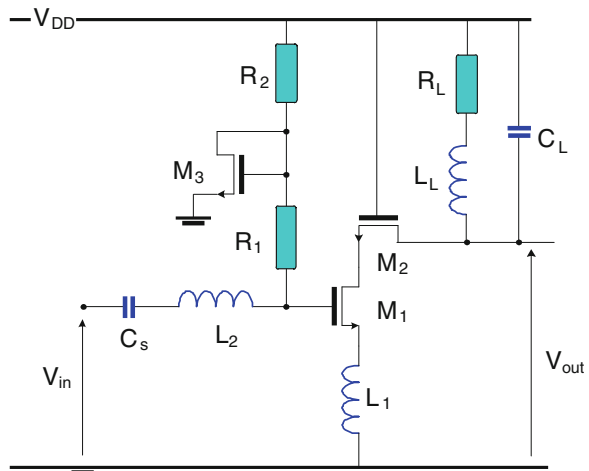


- An UMTS dedicated LNA

Figure 14.4 presents a CMOS LNA, in which topology was optimized in order to be dedicated for UMTS applications.  $R_1$ ,  $R_2$ , and  $M_3$  form the bias circuitry.  $M_2$  forms the isolation stage between the input and the output of the circuit.  $L_L$ ,  $R_L$ , and  $C_L$  form the circuit's output impedance.

In Sect. 14.4, we will deal with the optimal sizing of these circuits. Most important performances of such LNAs are considered, i.e., the voltage gain and the noise figure. It is to be noted that the voltage gain is handled via the scattering parameter 'S21' [8]. Corresponding expression (generated using a symbolic analyzer [18]), as well as expressions of the noise figure and the input/output matching conditions, is not provided. We refer the reader to [8] for details regarding these issues.

**Fig. 14.4** An UMTS CMOS LNA



### 14.3 PSO and BSA Metaheuristics

As introduced in Sect. 14.1, metaheuristics exhibit a wide spectrum of advantages when compared to the conventional mathematical optimization techniques. Metaheuristics are intrinsically stochastic techniques. They ensure random exploration of the parameter search space, allowing converging to the neighborhood of the global optimum within a reasonable computing time. According to [49], the name ‘metaheuristics’ was attributed to nature-inspired algorithms by Fred Glover [28].

Genetic algorithms [26, 33], which are parts of the evolutionary algorithms, are the oldest most known metaheuristics. A large number of variants of GA were proposed since the introduction of the basic GA (see for instance [15, 51]).

More recently, a new discipline was proposed, so-called swarm intelligence (SI). SI is an artificial reproduction of the collective behavior of individuals that is based on a decentralized control and self-organization [6].

A large number of such systems were studied by swarm intelligence, such as schools of fishes, flocks of birds, colonies of ants, and groups of bacteria, to name few processes [7, 9, 27, 34, 46]. Nowadays, particle swarm optimization may be the most known and the most used technique, particularly in the analog and RF circuits and systems designs, see for instance [13, 20, 21, 37, 48, 55].

More recently, a new improved variant of GA was proposed, and it is called backtracking search optimization technique (BSA) [11]. It offers some interesting features, mainly its robustness (vis-a-vis GA), its rapidity, and the low number of control parameters.

BSA is being used in the fields of analog and RF designs, see for instance [14, 16, 17, 30, 36]. BSA will be used for optimizing performances of both LNAs given in Sect. 14.2.

Presently, PSO is, as introduced above, largely used in design fields; it will also be considered for comparison reasons with BSA.

Furthermore, obtained results are also compared to the ones published in [3], using ant colony optimization (ACO) and backtrack ACO (BA-ACO) techniques.

- PSO technique is inspired from the observation of social behavior of animals, particularly birds and fish. It is a population-based approach that has the particularity that the decision within the group is not centralized [12, 34]. In short, PSO algorithm can be presented as follows.

The group, which is formed of potential solutions called particles, moves (flies) within the hyper search space seeking for the best location of food (the fitness value).

Movements of the group are guided by two factors: the particle velocity and the particle position, with respect to Eqs. (14.1) and (14.2).

$$\vec{v}_i(t + 1) = \begin{cases} \omega \vec{v}_i(t) \\ + C_1 \text{rand}(0, 1)(x_{p\text{best}i}(t) - \vec{x}_i(t)) \\ + C_2 \text{rand}(0, 1)(x_{G\text{best}i}(t) - \vec{x}_i(t)) \end{cases} \quad (14.1)$$

$$\vec{x}_i(t + 1) = \vec{x}_i(t) + \vec{v}_i(t) \quad (14.2)$$

$\vec{x}_{p\text{best}i}$  is the best position of particle  $i$  reached so far;  $\vec{x}_{G\text{best}i}$  is the best position reached by particle  $i$ 's neighborhood.

$\omega$ ,  $C_1 \text{rand}(0, 1)$ ,  $C_2 \text{rand}(0, 1)$  are weighting coefficients.  $\omega$  controls the diversification feature of the algorithm, and it is known as the inertia weight. It is a critical parameter that acts on the balance between diversification and intensification. Thus, a large value of  $\omega$  makes the algorithm unnecessarily slow. On the other hand, small values of  $\omega$  promote the local search ability.  $C_1$  and  $C_2$  control the intensification feature of the algorithm. They are known as the cognitive parameter and the social parameter, respectively.

PSO algorithm is given in Fig. 14.5.

As shown above, PSO algorithm is simple to be implemented and is computationally inexpensive. Thus, it does not require large memory space and is rapid, as well. These facts are on the basis of its popularity.

- BSA (or BSOA) is a new population-based global minimizer evolutionary algorithm for real-valued numerical optimization problems [11]. BSA offers some enhancements over the evolutionary algorithms, mainly the reduction in sensitivity to control parameters and improvement in convergence performances.

Classic genetic operators, namely selection, crossover, and mutation, are used in BSA, but in a novel way.

BSA encompasses five main processes: (i) initialization, (ii) selection ①, (iii) mutation, (iv) crossover, and (v) selection ②. BSA structure is simple, which confers low computational cost, rapidity, and necessitates low memory space. Moreover, the power of BSA can be summarized through its control process of the search directions within the parameters' hyperspace.

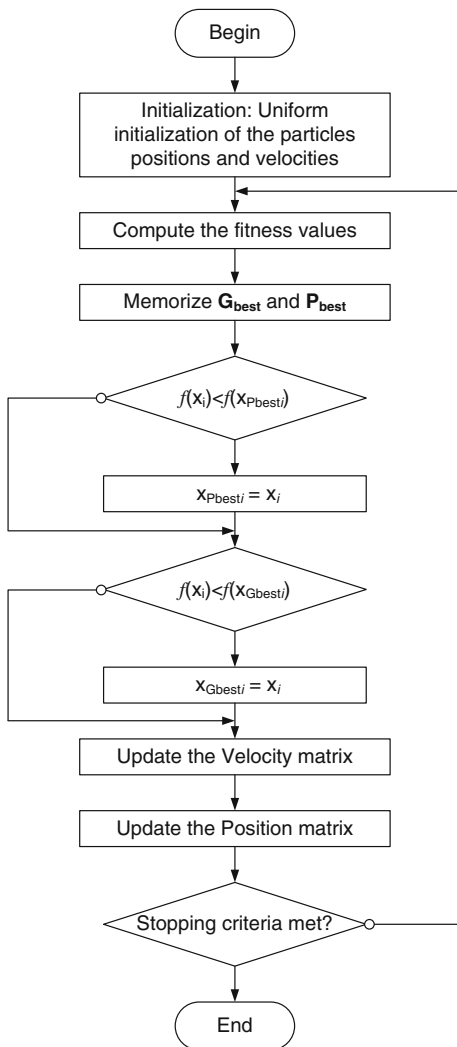
BSA algorithm is given in Fig. 14.6.

- *Initialization*: The population  $\mathbf{P} = (p_{ij})_{(N,M)}$  is initialized via a uniform stochastic selection of particles values within the hypervolume search space, as shown by expression (14.3):

$$p_{ij} = p_{j\text{min}} + \text{rand}(0, 1)(p_{j\text{Max}} - p_{j\text{min}}) \quad (14.3)$$

$$(i, j) \in \{1, \dots, N\} \times \{1, \dots, M\}$$

**Fig. 14.5** Basic flowchart of PSO

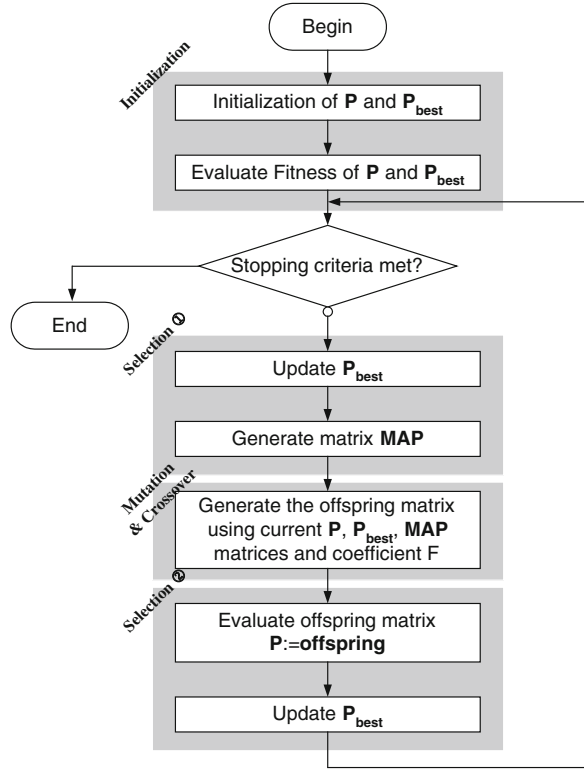


BSA takes benefits from previous experiences of the particles; thus, it uses a memory where the best position of each particle visited so far is memorized. The corresponding matrix noted  $\mathbf{P}_{best} = (p_{bestij})_{(N,M)}$  is initialized in the same way as matrix  $\mathbf{P}$ :

$$\begin{aligned}
 p_{bestij} &= p_{jmin} + \text{rand}(0, 1)(p_{jMax} - p_{jmin}) \\
 (i, j) &\in \{1, \dots, N\} \times \{1, \dots, M\}
 \end{aligned}
 \tag{14.4}$$



Fig. 14.6 BSA algorithm



- Selection ① consists of the update of the  $\mathbf{P}_{best}$  matrix,
- The Mutation process operates as follows. A mutant  $\mathbf{MUTANT} = (\text{mutant}_{ij})_{(N,M)}$  is generated as shown in Eq. (14.5).

$$\text{mutant}_{ij} = p_{ij} + \mathcal{F}(p_{best_{ij}} - p_{ij}) \quad (14.5)$$

$$(i,j) \in \{1, \dots, N\} \times \{1, \dots, M\}$$

$\mathcal{F}$  is a normally distributed factor that is used to control the search path, i.e., the direction.

- Crossover: It consists in generating a uniformly distributed integer valued matrix  $\mathbf{MAP} = (\text{map}_{ij})_{(N,M)}$ .  $\mathbf{MAP}$  elements values are controlled via a strategy that defines the number of particle components that mutate. This is performed via the ‘dimension-rate’ coefficient. Matrix  $\mathbf{MAP}$  is used for determining the matrix  $\mathbf{P}$  components to be handled: the offspring matrix.
- Selection ② consists of the update of the trial population via  $\mathbf{P}_{best}$  matrix.

### 14.4 Experimental Results and Comparisons

In this section, we will first deal with the application of the BSA technique to some mathematic test functions and give comparison results with the ones obtained by means of PSO regarding the robustness and the algorithm execution time. Then, we will consider the case of both LNAs introduced in Sect. 14.2. It is to be noted that a Core™2 Duo Processor T5800 (2 M Cache, 2.00 GHz, 4.00 Go) PC was used for that purpose.

- **Test functions**

Five test functions were considered: *DeJong’s*, *Eason 2D*, *Griewank*, *Parabola*, and *Rosenbrock*.

The corresponding expressions are given by (14.6)–(14.10), respectively. Figure 14.7 shows a plot of these functions.

Both algorithms, i.e., PSO and BSA, were run 100 times. The algorithms’ parameters are given in Table 14.1.

$$\begin{aligned}
 f(x) &= \sum_{i=1}^n ix_i^4 \\
 &- 5.12 \leq x_i \leq 5.12
 \end{aligned}
 \tag{14.6}$$

$$\begin{aligned}
 f(x) &= -\cos(x_1) \cos(x_2) e^{-(x_2-\pi)^2 - (x_2-\pi)^2} \\
 &- 5 \leq x_1 \leq 5, -5 \leq x_2 \leq 5
 \end{aligned}
 \tag{14.7}$$

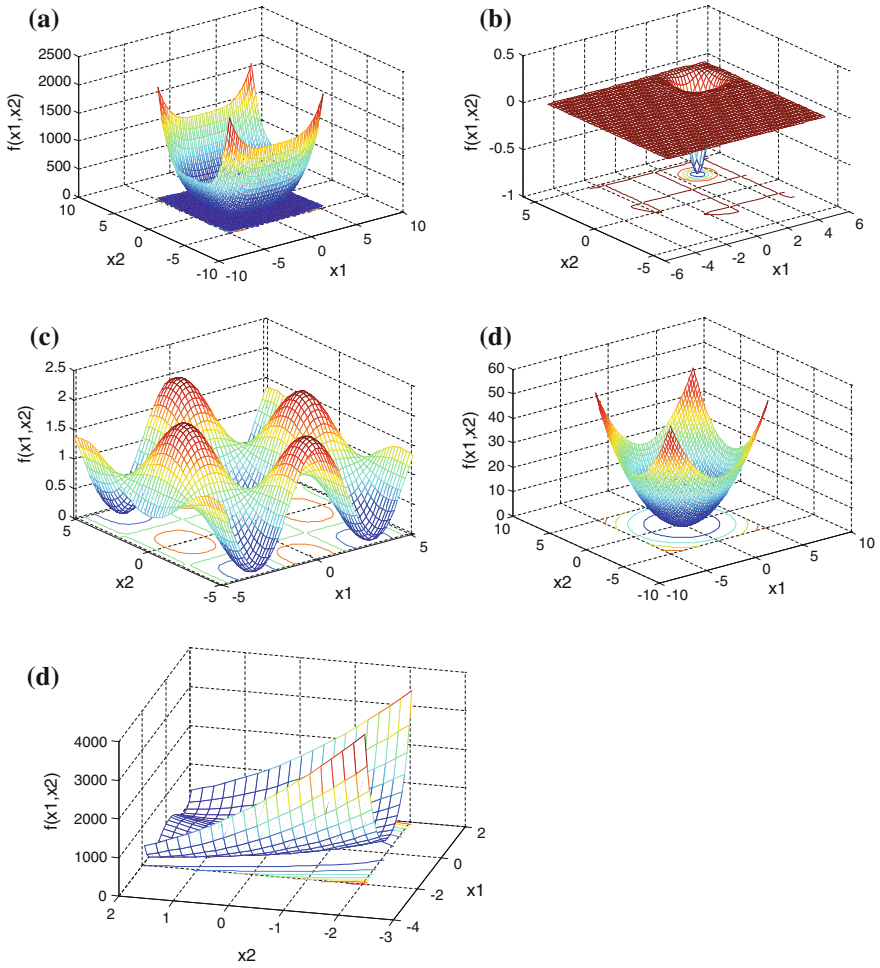
$$\begin{aligned}
 f(x) &= \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \\
 &5.12 \leq x_i \leq 5.12
 \end{aligned}
 \tag{14.8}$$

$$\begin{aligned}
 f(x) &= \sum_{i=1}^n x_i^2 \\
 &- 5.12 \leq x_i \leq 5.12
 \end{aligned}
 \tag{14.9}$$

$$\begin{aligned}
 f(x) &= \sum_{i=1}^{n-1} \left[ 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right] \\
 &- 2.048 \leq x_i \leq 2.048
 \end{aligned}
 \tag{14.10}$$

Figure 14.8 gives a whisker boxplot relative to the 100 executions of both algorithms.

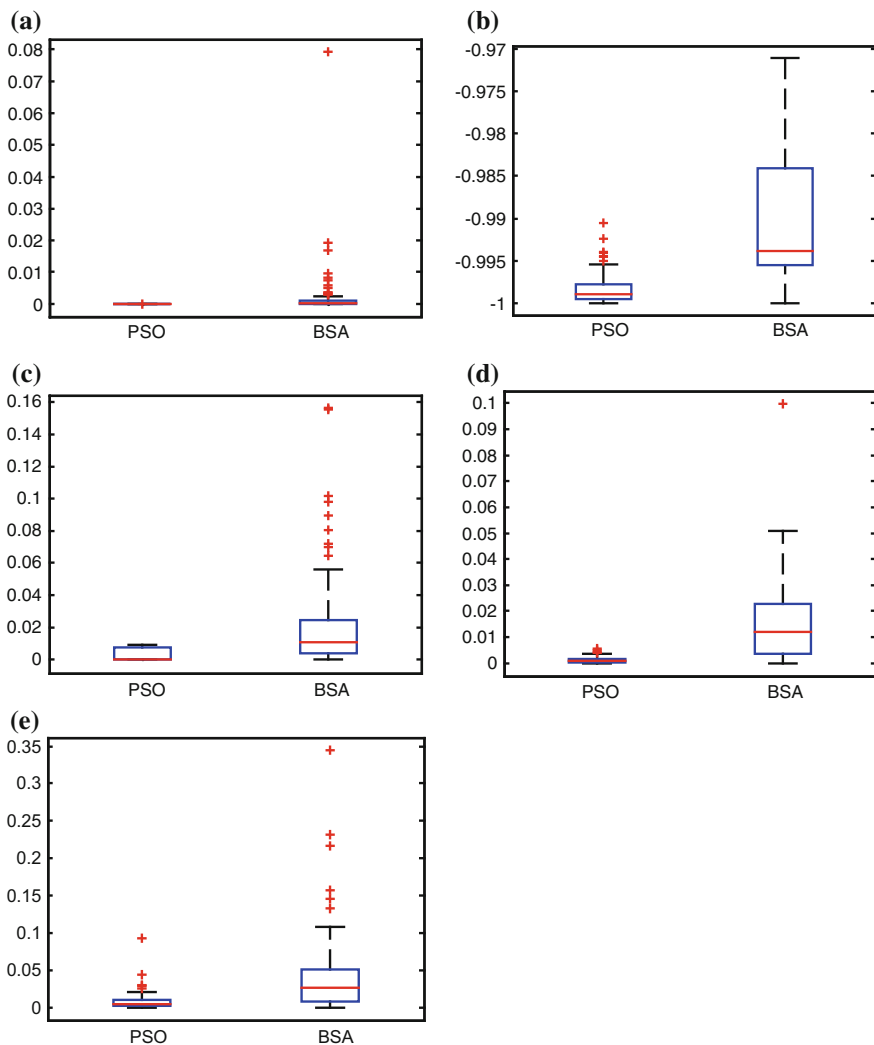
Table 14.2 gives the mean execution time of both algorithms with respect to the five considered functions.



**Fig. 14.7** Plots of the five considered functions ( $n = 2$ ). ( $x^*$  is the minimum of  $f$ ). **a** DeJong's  $x^* = (0, 0)$ ,  $f(x^*) = 0$ . **b** Eason 2D  $x^* = (\pi, \pi)$ ,  $f(x^*) = -1$ . **c** Griewank  $x^* = (0, 0)$ ,  $f(x^*) = 0$ . **d** Parabola  $x^* = (0, 0)$ ,  $f(x^*) = 0$ . **e** Rosenbrock  $x^* = (1)$ ,  $f(x^*) = 0$

**Table 14.1** PSO and BSA algorithms' parameters

PSO	$\omega$	$C_1$	$C_2$
	0.9	2	2
BSA	$F$	Dimension-rate	
	3	1	



**Fig. 14.8** Boxplot of the 100 executions results for both PSO and BSA applied to the five test functions. **a** DeJong's. **b** Eason 2D. **c** Griewank. **d** Parabola. **e** Rosenbrock

**Table 14.2** Mean execution time for PSO and BSA (sec.)

Test function	DeJong's	Eason 2D	Griewank	Parabola	Rosenbrock
PSO	24.133	26.848	24.313	25.044	25.219
BSA	0.163	0.292	0.195	0.163	0.170

• LNAs

PSO and BSA algorithms were used for optimally sizing both LNAs presented in Sect. 14.2. The same conditions observed above were considered. Tables 14.3 and 14.4 give the circuits’ optimized parameters. Moreover, simulations were performed using ADS software to check the viability of these results. Obtained performances are given in Tables 14.5 and 14.6. In addition, these tables present the results published in [4] obtained by application of ACO and BA-ACO techniques.

**Table 14.3** Multistandard LNA’s optimal parameters’ values

	$W_{1,2}$ ( $\mu\text{m}$ )/ $L_{1,2}$ ( $\mu\text{m}$ )	Id (mA)	Cch (pF)	Rch ( $\Omega$ )	Lch (nH)
PSO	429.98/0.35	96.96	10.00	0.39	0.62
BSA	441.28/0.35	100.00	6.06	0.49	0.90
	Lg (nH)	Ls (nH)	$C_1$ (pF)	$C_2$ (pF)	$W_3$ ( $\mu\text{m}$ )/ $L_3$ ( $\mu\text{m}$ )
PSO	8.14	0.52	916.90	405.30	40.00/0.35
BSA	10.00	0.35	1000.00	214.00	40.00/0.35

**Table 14.4** UMTS LNA’s optimal parameters’ values

	$W_1$ ( $\mu\text{m}$ )/ $L_1$ ( $\mu\text{m}$ )	$W_2$ ( $\mu\text{m}$ )/ $L_2$ ( $\mu\text{m}$ )	$W_3/L_3$ ( $\mu\text{m}$ )	Id (mA)	Cch (pF)
PSO	727.41/0.35	995.92/0.35	40.00/0.35	32.33	8.35
BSA	553.19/0.35	513.20/0.35	40.00/0.35	17.7	9.50
	Lch (nH)	Rch ( $\Omega$ )	Lg (nH)	Ls (nH)	
PSO	0.70	1.24	6.39	0.40	
BSA	0.55	1.20	10	0.27	

**Table 14.5** UMTS LNA’s optimal performances

$S_{21}$ (dB)	C++ results	ADS simulation results
PSO	16.33	16.46
BSA	16.11	15.64
ACO [4]	16.49	16.47
BA-ACO [4]	16.49	16.36

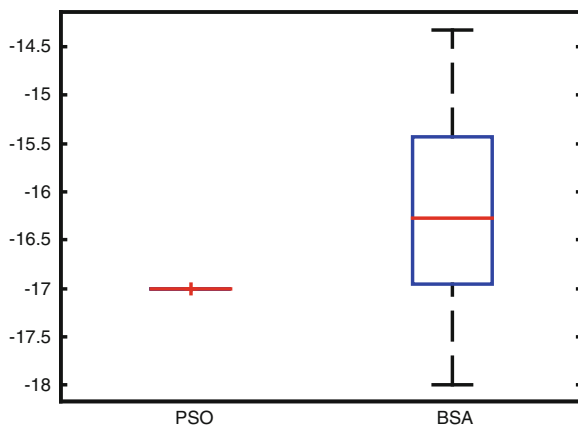
**Table 14.6** Multistandard LNA’s optimal performances

$S_{21}$ (dB)	C++ results		ADS simulation results	
	@1.5 GHz	@2.5 GHz	@1.5 GHz	@2.5 GHz
PSO	8.27	11.44	8.78	11.10
BSA	11.56	22.55	8.85	11.02
ACO [4]	10.65	15.48	9.00	11.46
BA-ACO [4]	11.32	18.65	10.68	11.62

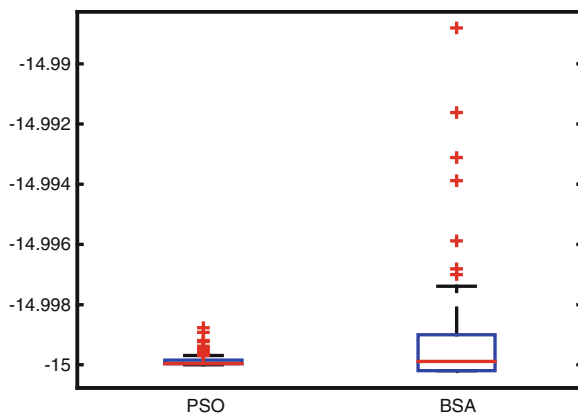
**Table 14.7** Mean execution time per run

	UMTS LNA (s)	Multistandard LNA (s)
PSO	3.56	2.96
BSA	1.12	0.60
ACO [4]	27.56	38.73
BA-ACO [4]	19.00	31.22

**Fig. 14.9** Boxplot for the 100 executions runs for the UMTS LNA using PSO and BSA



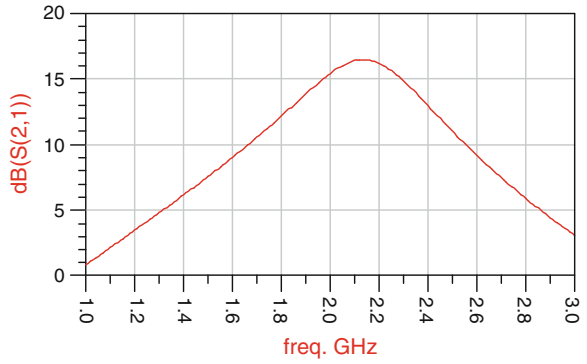
**Fig. 14.10** Boxplot for the 100 executions runs for the multistandard LNA using PSO and BSA



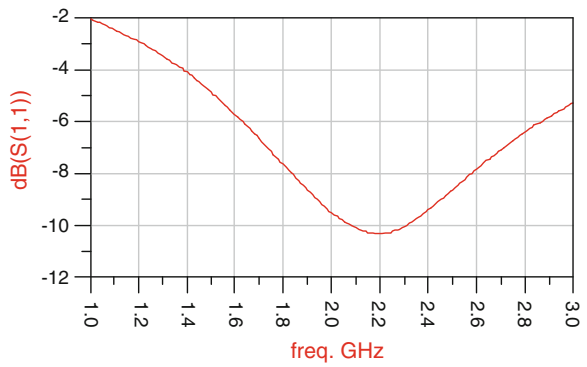
The mean execution times for both problems are given in Table 14.7. Robustness results are shown in Figs. 14.9 and 14.10.

Simulation results obtained using the ‘a priori’ optimal parameters for both circuits are depicted in Figs. 14.11, 14.12, 14.13, 14.14, 14.15, 14.16, 14.17, 14.18, 14.19, 14.20, 14.21, 14.22, 14.23, 14.24, 14.25, and 14.26.

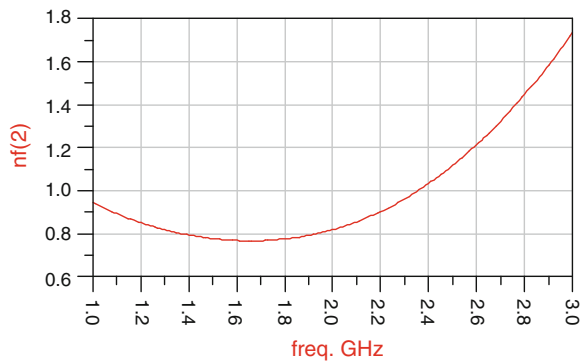
**Fig. 14.11** ADS simulation results of S21 of the UMTS LNA, using PSO results



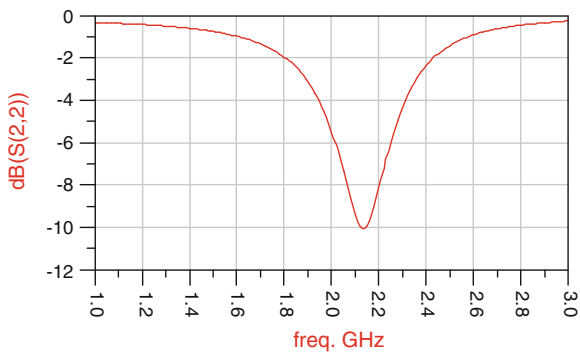
**Fig. 14.12** ADS simulation results of S11 of the UMTS LNA, using PSO results



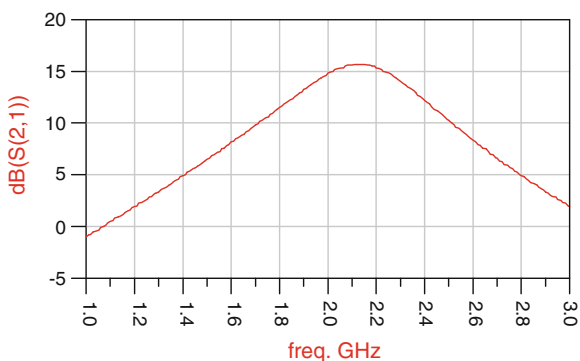
**Fig. 14.13** ADS simulation results of the noise figure of the UMTS LNA, using PSO results



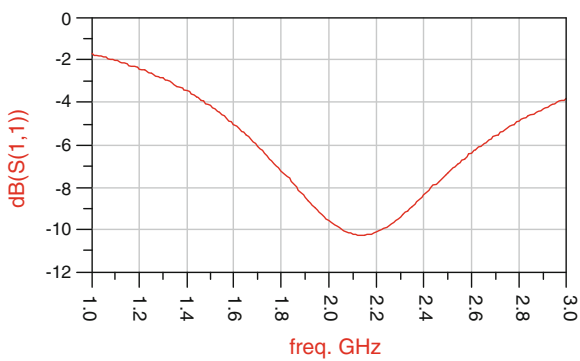
**Fig. 14.14** ADS simulation results of S22 of the UMTS LNA, using PSO results



**Fig. 14.15** ADS simulation results of S21 of the UMTS LNA, using BSA results

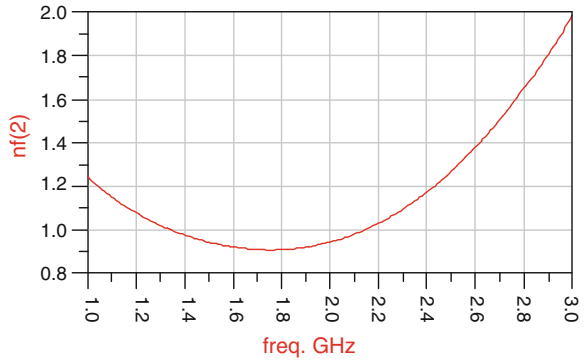


**Fig. 14.16** ADS simulation results of S11 of the UMTS LNA, using BSA results

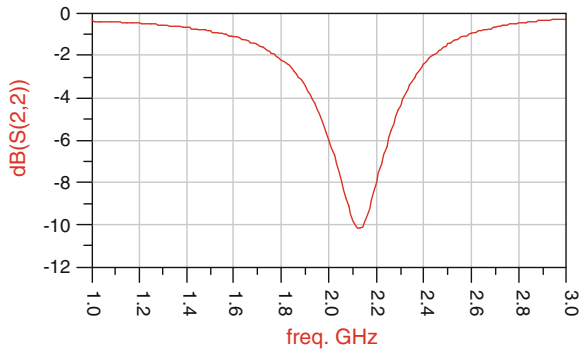




**Fig. 14.17** ADS simulation results of the noise figure of the UMTS LNA, using BSA results



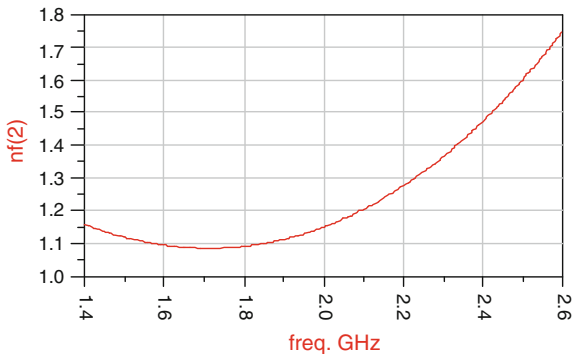
**Fig. 14.18** ADS simulation results of S22 of the UMTS LNA, using BSA results



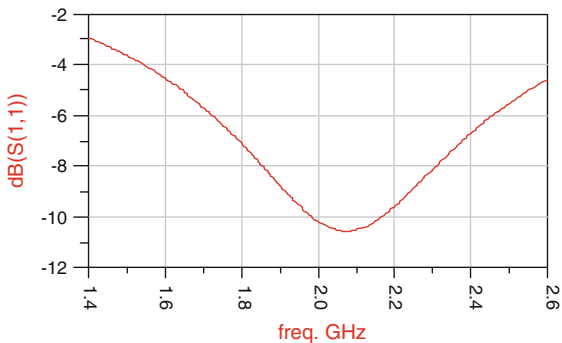
**Fig. 14.19** ADS simulation results of S21 of the multistandard LNA, using PSO results



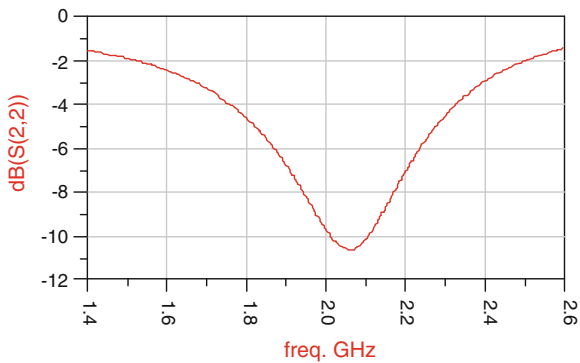
**Fig. 14.20** ADS simulation results of the noise figure of the multistandard LNA, using PSO results



**Fig. 14.21** ADS simulation results of S11 of the multistandard LNA, using PSO results



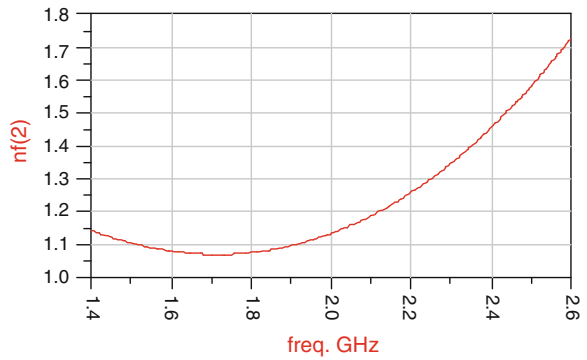
**Fig. 14.22** ADS simulation results of S22 of the multistandard LNA, using PSO results



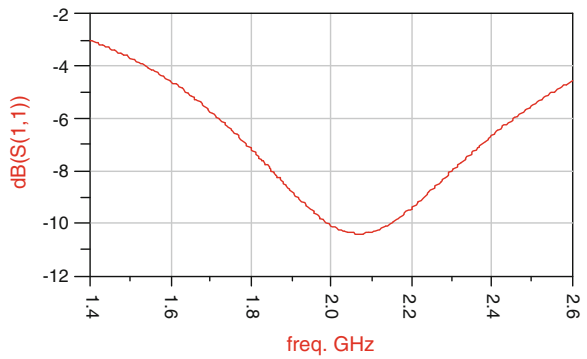
**Fig. 14.23** ADS simulation results of S21 of the multistandard LNA, using BSA results



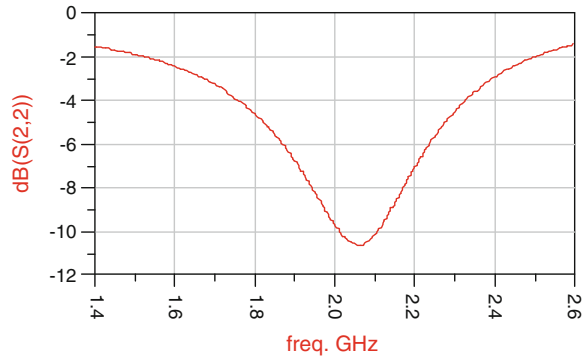
**Fig. 14.24** ADS simulation results of the noise figure of the multistandard LNA, using BSA results



**Fig. 14.25** ADS simulation results of S11 of the multistandard LNA, using BSA results



**Fig. 14.26** ADS simulation results of S22 of the multistandard LNA, using BSA results



## 14.5 Discussion and Conclusion

This chapter investigated the application of BSA, the very recently proposed EA technique, on the resolution of RF sizing problems. For comparison reasons, PSO technique was also applied for optimizing these circuits (namely two LNAs). Furthermore, obtained performances were also compared with the already published results dealing with the same circuits but using ACO and BA-ACO, and also the application to the resolution of some test functions.

The obtained results show that BSA outperforms the other optimization techniques in terms of computing time. However, it has been noted that PSO is relatively more robust. Nonetheless, the rapidity of BSA and its good performances make this algorithm a good and interesting technique to be considered within a computer-aided design approach/tool.

## References

1. Allstot, D.J., Choi, K., Park, J. (eds.): Parasitic-Aware Optimization of CMOS RF Circuits. Kluwer academic publishers, New York (2003)
2. Barros, M., Guilherme, J., Horta, N.: Analog Circuits and Systems Optimization Based on Evolutionary Computation Techniques. Springer, New York (2010)
3. Benhala, B., Ahaitouf, A., Kotti, M., Fakhfakh, M., Benlahbib, B., Mecheqrane, A., Loulou, M., Abdi, F., Abarkane, E.: Application of the ACO technique to the optimization of analog circuit performances. In: Tlelo-Cuautle, E. (ed.) Analog Circuits: Applications, Design and Performances, pp. 235–255. Nova Science Publishers Inc, New York (2011)
4. Benhala, B., Kotti, M., Ahaitouf, A., Fakhfakh, M.: Backtracking ACO for RF-circuit design optimization. In: Fakhfakh, M., et al. (eds.) Performance Optimization Techniques in Analog, Mixed-Signal, and Radio-Frequency Circuit Design, pp. 1–22. IGI-Global, USA (2014)
5. Benhala, B., Bouattane, O.: GA and ACO techniques for the analog circuits design optimization. *J. Theor. Appl. Inf. Technol.* **64**(2), 413–419 (2014)
6. Blum, C., Merkle, D. (eds.): Swarm intelligence: introduction and applications. Springer, New York (2008)

7. Bonabeau, E., Theraulaz, G., Dorigo, M.: *Swarm intelligence: from natural to artificial systems*. Oxford University Press, New York (1999)
8. Boughariou, M., Fakhfakh, M., Loulou, M.: Design and optimization of LNAs through the scattering parameters. In: *The IEEE Mediterranean Electrotechnical Conference, Malta, 26–28 Apr 2010*
9. Chan, F.T.S., Tiwari, M.K.: *Swarm Intelligence: Focus on Ant and Particle Swarm Optimization*. I-Tech education and publishing, Croatia (2007)
10. Chatterjee, A., Fakhfakh, M., Siarry, P.: Design of second-generation current conveyors employing bacterial foraging optimization. *Microelectron. J.* **41**(10), 616–626 (2010)
11. Civicioglu, P.: Backtracking search optimization algorithm for numerical optimization problems. *Appl. Math. Comput.* **219**(15), 8121–8144 (2013)
12. Clerc, M.: *Particle Swarm Optimization*. ISTE Ltd., London (2006)
13. Datta, R., Dutta, A., Bhattacharyya, T.K.: PSO-based output matching network for concurrent dual-band LNA. In: *International Conference on Microwave and Millimeter Wave Technology, Chengdu, China, 8–11 May 2010*
14. de Sá, A.O., Nedjah, N., Mourelle, L.-M.: Genetic and backtracking search optimization algorithms applied to localization problems. In: Murgante, B. et al. (eds.) *Computational Science and Its Applications. Lecture notes in computer science (8583)*, pp. 738–746. Springer, Heidelberg (2014)
15. Dréo, J., Pétrowski, A., Siarry, P., Taillard, E.: *Metaheuristics for Hard Optimization: Methods and Case Studies*. Springer, New York (2006)
16. Duan, H., Luo, Q.: Adaptive backtracking search algorithm for induction magnetometer optimization. *IEEE Trans. Magn.* **50**(12), 1–6 (2014)
17. El-Fergany, A.: Optimal allocation of multi-type distributed generators using backtracking search optimization algorithm. *Electr. Power Energy Syst.* **64**, 1197–1205 (2015)
18. Fakhfakh, M., Loulou, M.: Live demonstration: CASCADES. 1: a flow-graph-based symbolic analyzer. In: *The IEEE International Symposium on Circuits and Systems, Paris, France, 30 May 2010–02 June 2010*
19. Fakhfakh, M., Sallem, A., Boughariou, M., Bennour, S., Bradai, E., Gaddour, E., Loulou, M.: Analogue circuit optimization through a hybrid approach. In: Koeppen, M., Schaefer, G., Abraham, A. (eds.) *Intelligent Computational Optimization in Engineering: Techniques and Applications*, pp. 297–327. Springer, New York (2010)
20. Fakhfakh, M., Cooren, Y., Sallem, A., Loulou, M., Siarry, P.: Analog circuit design optimization through the particle swarm optimization technique. *Analog Integr. Circ. Sig. Process* **63**(1), 71–82 (2010)
21. Fakhfakh, M., Siarry, P.: MO-TRIBES for the optimal design of analog filters. In: Fornarelli, G., Mescia, L. (eds.) *Swarm Intelligence for Electric and Electronic Engineering*, pp. 57–70. IGI-Global, USA (2013)
22. Fakhfakh, M., Tlelo-Cuautle, E., Castro-Lopez, R. (eds.): *Analog/RF and Mixed-Signal Circuit Systematic Design*. Springer, New York (2013)
23. Fakhfakh, M., Tlelo-Cuautle, E., Fino, M.H. (eds.): *Performance Optimization Techniques in Analog, Mixed-Signal, and Radio-Frequency Circuit Design*. IGI-Global, Hershey, Pennsylvania (2014)
24. Fallahpour, M.B., Hemmati, K.D., Pourmohammad, A.: Optimization of a LNA using genetic algorithm. *Electr. Electron. Eng.* **2**(2), 38–42 (2012)
25. Fathianpour, A., Seyedtabaii, S.: Evolutionary search for optimized LNA components geometry. *J. Circ. Syst. Comput.* **23**(1), 1–16 (2014)
26. Fisher, R.A.: *The Genetical Theory of Natural Selection*. Oxford University Press, New York (1958)
27. Fornarelli, G., Mescia, L.: *Swarm Intelligence for Electric and Electronic Engineering*. IGI-Global, USA (2013)
28. Glover, F.: <http://leeds-faculty.colorado.edu/glover/>. Accessed Nov 2014
29. Grimbleby, J. B.: Automatic analogue circuit synthesis using genetic algorithms. In: *IEEE Proceedings-Circuits, Devices and Systems*, vol. 147, no. 6, pp. 319–323 (2000)

30. Guney, K., Durmus, A., Basbug, S.: Backtracking search optimization algorithm for synthesis of concentric circular antenna arrays. *Int. J. Antennas Propag.* **2014**, 1–11 (2014)
31. Gupta, N., Saini, G.: Performance analysis of BFO for PAPR reduction in OFDM. *Int. J. Soft Comput. Eng.* **2**(5), 127–133 (2012)
32. Hemmati, K.D., Fallahpour, M.B., Golmakani, A.: Design and optimization of a new LNA. *Majlesi J. Telecommun. Devices* **1**(3), 91–96 (2012)
33. Holland, J.H.: *Hidden Order: How Adaptation Builds Complexity*. Addison-Wesley, Redwood City (1995)
34. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *IEEE International Conference on Neural Networks*, Perth, 27 Nov 1995–01 Dec 1995
35. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *J. Sci.* **220**, 671–680 (1983)
36. Kolawole, S.O., Duan, H.: Backtracking search algorithm for non-aligned thrust optimization for satellite formation. In: *The IEEE International Conference on Control and Automation*, Taichung, Taiwan, 18–20 June 2014
37. Kotti, M., Sallem, A., Bougharriou, M., Fakhfakh, M., Loulou M.: Optimizing CMOS LNA circuits through multi-objective meta-heuristics. *The international workshop on symbolic and numerical methods, modeling and applications to circuit design*, Gammarth, Tunisia, 4–6 Oct 2010
38. Kumar, P., Duraiswamy, K.: An optimized device sizing of analog circuits using particle swarm optimization. *J. Comput. Sci.* **8**(6), 930–935 (2012)
39. Li, Y., Yu, S.-M., Li, Y.-L.: A simulation-based hybrid optimization technique for low noise amplifier design automation. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) *Proceedings of the 7th International Conference on Computational Science, Part IV: ICCS 2007*, May 2007. *Lecture Notes in Computer Science*, vol. 4490, pp. 259–266. Springer, Heidelberg (2007)
40. Li, Y.: A simulation-based evolutionary approach to LNA circuit design optimization. *Appl. Math. Comput.* **209**(1), 57–67 (2009)
41. Li, Y.: Simulation-based evolutionary method in antenna design optimization. *Math. Comput. Model.* **51**(7–8), 944–955 (2010)
42. Liu, Z., Liu, T., Gao, X.: An improved ant colony optimization algorithm based on pheromone backtracking. In: *The IEEE International Conference on Computational Science and Engineering*, Dalian, China, 24–26 Aug 2011
43. Lourenço, N., Martins, R., Barros, M., Horta, N.: Analog circuit design based on robust POFs using an enhanced MOEA with SVM models. In: Fakhfakh, M., Tlelo-Cuautle, E., Castro-Lopez, R. (eds.) *Analog/RF and Mixed-Signal Circuit Systematic Design*. Springer, New Jersey (2013)
44. Neoh, S.C., Marzuki, A., Morad, N., Lim, C.P., Abdul Aziz, Z.: An interactive evolutionary algorithm for MMIC low noise amplifier design. *ICIC Express Lett.* **3**(1), 15–19 (2009)
45. Razavi, B.: *RF Microelectronics*. Prentice Hall, Canada (2012)
46. Reyes-Sierra, M., Coello-Coello, C.A.: Multi-objective particle swarm optimizers: a survey of the state-of-the-art. *Int. J. Comput. Intell. Res.* **2**(3), 287–308 (2006)
47. Roca, E., Fakhfakh, M., Castro-Lopez, R., Fernandez, F.V.: Applications of evolutionary computation techniques to analog, mixed-signal and RF circuit design—an overview. In: *The IEEE International Conference on Electronics, Circuits, and Systems*, Yasmine Hammamet, Tunisia, 13–16 Dec 2009
48. Sallem, A., Benhala, B., Kotti, M., Fakhfakh, M., Ahaitouf, A., Loulou, M.: Application of swarm intelligence techniques to the design of analog circuits: evaluation and comparison. *Analog Integr. Circ. Sig. Process* **75**(3), 499–516 (2013)
49. Scholarpedia.org [http://www.scholarpedia.org/article/Metaheuristic\\_Optimization](http://www.scholarpedia.org/article/Metaheuristic_Optimization). Accessed Nov 2014
50. Shin, L.W., Chin, N.S., Marzuki, A.: 5 GHz MMIC LNA design using particle swarm optimization. *Inf. Manag. Bus. Rev.* **5**(6), 257–262 (2013)

51. Siarry, P., Michalewicz, Z.: *Advances in Metaheuristics for Hard Optimization*. Springer, New York (2007)
52. Siarry, P. (ed.): *Heuristics, Theory and Applications*. Nova publisher, USA (2013)
53. Silva, L.G., Jr A.C.S., da Silva S.C.H.: Development of tri-band RF filters using evolutionary strategy. *AEU Int. J. Electron. Commun.* **68**(12), 1156–1164 (2014)
54. Sönmez, Ö.S., Dündar, G.: Simulation-based analog and RF circuit synthesis using a modified evolutionary strategies algorithm. *Integr. VLSI J.* **44**(2), 144–154 (2011)
55. Tripathi, J.N., Mukherjee, J., Apte, P.R.: Design automation, modeling, optimization, and testing of analog/RF circuits and systems by particle swarm optimization. In: Fornarelli, G., Mescia, L. (eds.) *Swarm Intelligence for Electric and Electronic Engineering*, pp. 57–70. IGI-Global, USA (2013)
56. Ushie, O.J., Abbod, M.: Intelligent optimization methods for analogue electronic circuits: GA and PSO case study. In: *The International Conference on Machine Learning, Electrical and Mechanical Engineering*, Dubai 8–9 Jan 2014
57. Valadi, J., Siarry, P. (eds.): *Applications of Metaheuristics in Process Engineering*. Springer, New York (2014)

# Chapter 15

## Design of Telecommunication Receivers Using Computational Intelligence Techniques

Laura-Nicoleta Ivanciu and Gabriel Oltean

**Abstract** This chapter proposes system-, block-, and circuit-level design procedures that use computational intelligence techniques, taking into consideration the specifications for telecommunication receivers. The design process starts with selecting the proper architecture (topology) of the system, using a fuzzy expert solution. Next, at the block level, the issue of distributing the parameters across the blocks is solved using a hybrid fuzzy-genetic algorithms approach. Finally, multiobjective optimization using genetic algorithms is employed in the circuit-level design. The proposed methods were tested under specific conditions and have proved to be robust and trustworthy.

### 15.1 Introduction

Current trends in circuit and system design converge toward automating the design process and obtaining a final version of the circuit or system, by evolving a previous one. Computational intelligence techniques, such as genetic algorithms (GA), are widely employed in modern circuit and system design [6].

Computational intelligence techniques include practical concepts of adaptation and self-organization, and algorithms and implementations that facilitate the intelligent behavior in complex and variable environments. Computational intelligence techniques are successfully applied to solve problems that cannot be fully described by formal models or those for which formal models involve the use of expensive algorithms [4, 15].

Compared to artificial intelligence and, thus, hard computing, computational intelligence benefits from the ability to generalize, work with imprecise data, fault tolerance and noise, and the ability to work in various environments. Computational

---

L.-N. Ivanciu (✉) · G. Oltean  
Technical University of Cluj-Napoca, Cluj-Napoca, Romania  
e-mail: laura.ivanciu@bel.utcluj.ro



intelligence techniques include evolutionary computation, fuzzy logic, neural networks, and self-organizing maps.

On the other hand, telecommunication systems are ever present in day-to-day life, in various forms, from the most basic mobile terminal, to the most sophisticated 4G smartphone, Internet, and so on.

The theme of this chapter was chosen with respect to the importance of the domain—well represented in industrial applications, and also highly focused on in research activities. The important number of scientific journals and conferences that approach the use of computational intelligence techniques in various fields was also taken into account.

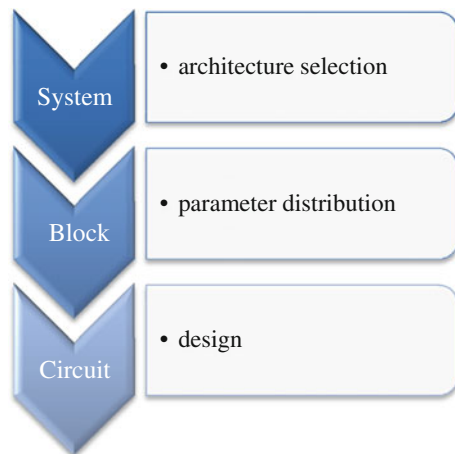
Moreover, these techniques are still considered new research fields, as their consistent use only emerged in the 1990s. They have proved to deliver very good results in electronic circuits and systems design and optimization, as well as in many other areas, and so the extension of their use is fully justified.

This chapter aims to investigate, develop, and implement innovative methods that use computational intelligence techniques, such as fuzzy logic, GA, or multi-objective optimization, in the design of telecommunication receivers. Based on a top-down approach, contributions are made starting from system level, continuing with block level, and finally reaching the circuit level, as suggested in Fig. 15.1.

## 15.2 Fuzzy Expert System for Receiver Architecture Selection

The section starts with fundamentals regarding radiofrequency (RF) receivers. Four of the most commonly used structures are then briefly presented, highlighting their advantages and drawbacks. The proposed solution, a fuzzy expert system that

**Fig. 15.1** Structure of this chapter—a top-down approach



selects the most proper architecture, given certain constraints, is then described. The entire design and implementation procedure is illustrated; the system is tested and evaluated and the results are listed and discussed.

### 15.2.1 Basic Receiver Architectures

A complete telecommunication system is made of two complementary signal paths, the transmission path and the reception one, respectively. The complementarity of these paths derives from the functionality of the consisting blocks: for instance, if a transmitter contains a modulator and a high-frequency converter, the reception part must include a low-frequency converter and a demodulator.

Both analog and digital signal processing operations are involved for transmission and reception. Digital-to-analog or analog-to-digital converters (DAC/ADC) make the transition between the analog and digital sections of the system.

The part of the system which only works with analog signals is called “front-end” (FE) or “analog front-end,” and the one dedicated to digital signals is known as “back-end” or “digital back-end” [13]. The discussion will be focused on the receiver FE, from now on.

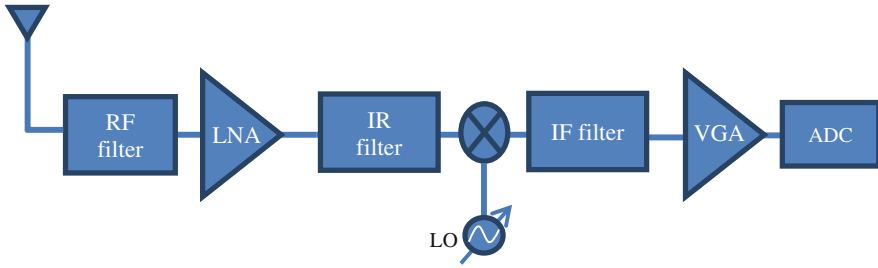
Receiver architectures for which the ADC works with low-frequency signal can be classified as follows:

- super-heterodyne,
- zero-IF or homodyne or direct conversion,
- low-IF, and
- double-conversion.

The super-heterodyne receiver is most commonly met in RF applications, due to its ability to precisely select high-frequency, narrowband signals, from an interference-affected environment [1]. The super-heterodyne architecture is based on multiplying the RF carrier with a locally generated signal, in order to obtain a super-sonic signal, which will be amplified and demodulated [3, 11].

The super-heterodyne architecture schematic is depicted in Fig. 15.2. The RF signal from the antenna is fed to an RF band-pass filter, which selects the band that contains the useful signal and rejects, as much as possible, the out of band interferers. The low-noise amplifier (LNA) amplifies the filtered signal, which is then translated toward lower frequencies; this translation is made possible by the mixer, which multiplies the signal with a generated sine wave, produced by the local oscillator (LO). Another filtering operation is applied, to select the channel. The last step involves amplification via the variable gain amplifier (VGA), and then, the signal reaches the ADC.

In most cases, two parallel paths are necessary for signal processing at the reception. They are called the I-path (in phase) and Q-path (in quadrature), the only difference between them being the phase of the signal generated by the LO.



**Fig. 15.2** Schematic representation of the super-heterodyne receiver architecture

The mixer translates to an intermediate frequency the bands located to the left and to the right from the LO frequency. Thus, an image rejection (IR) filter needs to be placed before the mixer, to reject the image frequency. This filter is usually located off-chip, and made of surface acoustic wave (SAW) components, which is the main draw-back of this architecture.

The other three basic RF architectures (zero-IF, low-IF, and double conversion) consist of similar processing blocks and thus will not be presented in detail. Their main advantages and drawbacks are outlined in the following section.

### 15.2.2 Comparison Between Basic Receiver Architectures

A synthesis of the advantages and drawbacks of each of the previously discussed architectures is presented in Table 15.1, based on [2, 12]. The zero-IF architecture seems to be the most attractive, from the point of view of integration and reconfiguration. The DC offset and flicker noise, common issues for this architecture, can be dealt with at block level (LNA, mixer).

**Table 15.1** Advantages and drawbacks of basic receiver architectures

Receiver architecture	Advantages	Drawbacks	Notes
Super-heterodyne	High selectivity	Image frequency off-chip components	Difficult to integrate difficult to reconfigure
Zero-IF	No image frequency no off-chip components	DC offset flicker noise	Easy to integrate easy to reconfigure
Low-IF	Low DC offset low flicker noise	Image frequency	Easy to integrate easy to reconfigure
Double-conversion	Low DC offset low flicker noise	Many components	Easy to integrate easy to reconfigure strict ADC design constraints

### 15.2.3 Design and Implementation of the Fuzzy Expert System for Receiver Architecture Selection

The solution pool from which the system chooses consists of the four previously mentioned receiver structures, namely super-heterodyne, zero-IF, Low-IF, and double conversion. The system is implemented using the fuzzy logic toolbox provided by the MATLAB/Simulink environment.

The system can be seen as an expert rule-based system, as it provides at the output the degrees of activation for each rule, rather than the defuzzified value of the output surface. This fuzzy expert system is dedicated to users who only have basic knowledge in the field and need a starting point in designing a telecommunication receiver.

Five characteristics are taken into account, in order to build the inputs of the system [8]: selectivity, analog requirements, flexibility, noise, and integrability. The call of the fuzzy system in MATLAB, listed below, gives us information about the type—Sugeno [also known as Takagi-Sugeno (TS) or Takagi-Sugeno-Kang (TSK)], AND method—min, OR method—max, defuzzification method—centroid, implication method—min, aggregation method—max, three inputs, four outputs, and four rules. Figure 15.3 [8] illustrates the fuzzy system.

```

fis =      name: 'fuzzyArchSelTSK'
          type: 'sugeno'
          andMethod: 'prod'
          orMethod: 'probor'
          defuzzMethod: 'wtaver'
          impMethod: 'prod'
          aggMethod: 'sum'
          input: [1x5 struct]
          output: [1x1 struct]
          rule: [1x4 struct]

```

The fuzzy sets for the input variables are depicted in Fig. 15.4 [8]. Z-type, Gaussian, and S-type fuzzy sets were chosen. The labels or linguistic values associated to the three membership functions are “low” for Z-type, “moderate” for Gaussian, and “high” for S-type.

It is worth mentioning that as the system is dedicated to inexperienced users, the universe of discourse for each input variable was chosen between 0 and 100, assuming a percentage-type representation. Also, this choice of interval description is favorable when the crisp values of the inputs can substantially change, due to implementation issues: take noise for instance—depending on the technology, the noise value for each structure can cover different ranges.

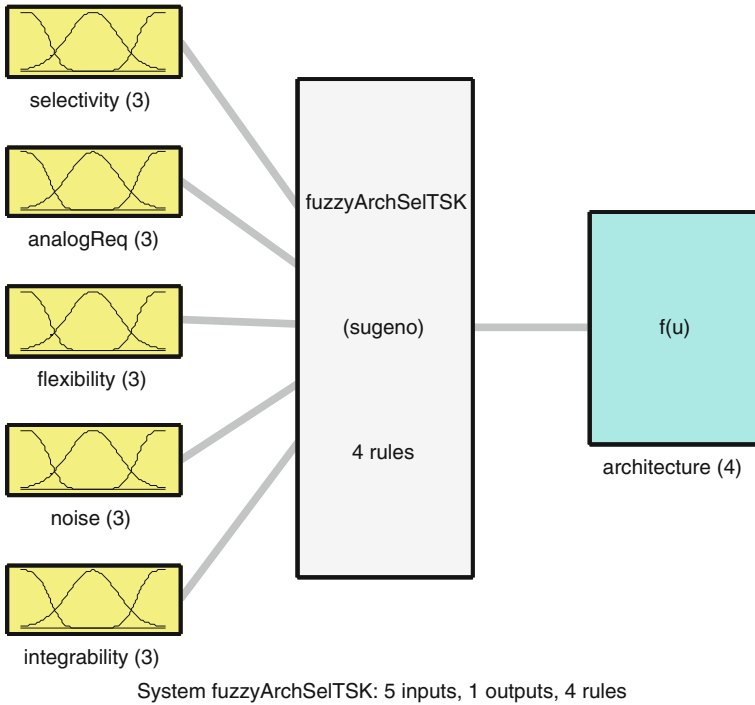


Fig. 15.3 Fuzzy system diagram [8]

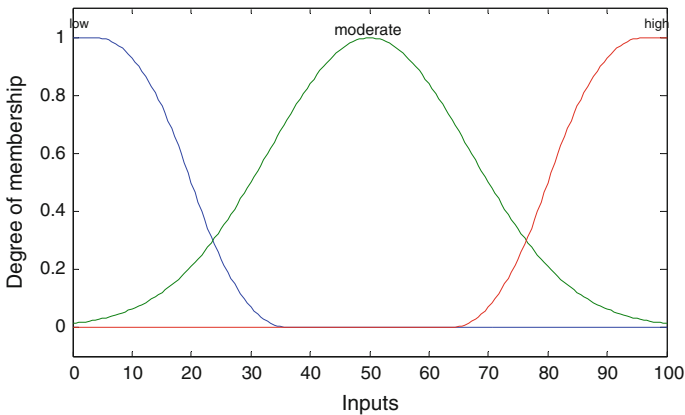


Fig. 15.4 Membership functions for input variables [8]

The rules, and implicitly the entire system, are based on the previously discussed key features of receiver architectures. The four rules of the system are listed below [8]. All the rules have the implicit firing strength, 1.

1. If (selectivity is high) and (analogReq is high) and (flexibility is low) and (noise is low) and (integrability is low) then (architecture is superHet) (1)
2. If (selectivity is high) and (analogReq is moderate) and (flexibility is low) and (noise is moderate) and (integrability is high) then (architecture is zeroIF) (1)
3. If (selectivity is high) and (analogReq is low) and (flexibility is high) and (noise is low) and (integrability is high) then (architecture is lowIF) (1)
4. If (selectivity is high) and (analogReq is moderate) and (flexibility is high) and (noise is low) and (integrability is moderate) then (architecture is doubleConv) (1)

The values chosen to represent the output variable, namely the architecture type, are clearly irrelevant as numbers or order, as the system's type is TSK (denoted as Sugeno in MATLAB's fuzzy logic toolbox). Equally spaced values, between 0 and 1, were chosen to represent the four possible architectures. Again, the universe of discourse is entirely up to the designer's liking. Moreover, the result provided at the output represents the activation degree for each rule, which is an indicator of the possible proper architecture. The final choice remains to be made by the user.

### 15.2.4 Results

The system was tested for four possible input values arrays, chosen so that they each activate one of the rules; so, a value of 10 was associated to *low*, 50 to *moderate*, and 90 to *high*. The test arrays and the results are listed in Table 15.2 [8].

The ruleviewer of the system, with input values that activate the first rule, is depicted in Fig. 15.5 [8]. The results in Table 15.2 are not surprising, because the input values were carefully selected, and so the final choice for the receiver architecture is more than obvious, in each case.

However, because the rule base is not complete, the system may return misleading results, when the input values do not activate any of the rules. Such an example is given in Fig. 15.6 [8].

**Table 15.2** Input values and results [8]

Input	Results
[90 90 10 10 10]	[0.6945 0 0 0]
[90 50 10 50 90]	[0 0.8035 0 0]
[90 10 90 10 90]	[0 0 0.6945 0.032]
[90 50 90 10 50]	[0 0 0 0.8035]

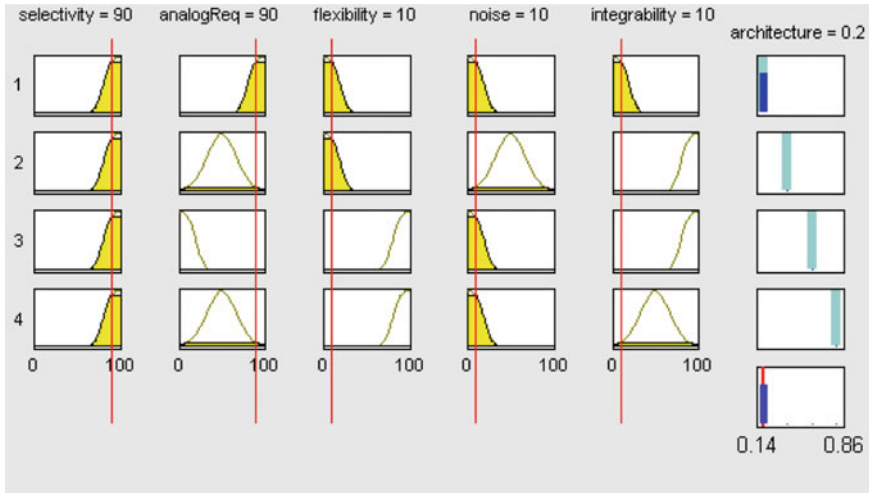


Fig. 15.5 The ruleviewer window—activation of the first rule [8]

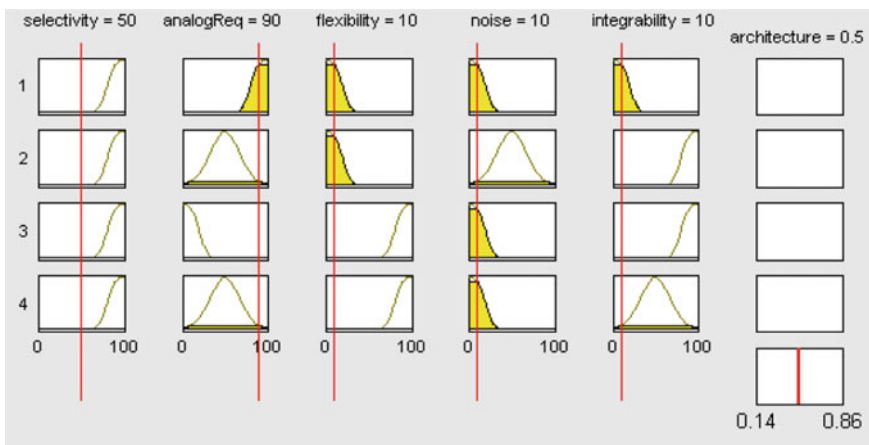


Fig. 15.6 The ruleviewer window—no rule is activated [8]

### 15.3 Computational Intelligence-Based Approach for Parameter Distribution in Receiver Chains

This section first briefly presents the main metrics (parameters) used in the design and performance evaluation of RF receivers chains. The solution proposed by the authors combines fuzzy logic with GA, in order to obtain an optimal parameter distribution over the blocks of the receiver chain. The implementation procedure, as

well as the results of testing this method for wideband code division multiple access (WCDMA) and wireless local area network (WLAN) standard specifications, is presented and discussed.

### 15.3.1 Parameters of RF Receiver Chains

In order to successfully complete a system- or circuit-level design task, one must know the means of performance evaluation, namely the most important parameters of the system or circuit. For RF receivers, the most important parameters are the noise factor and noise figure, the 1-dB compression point, and the second- and third-order intercept points [10].

- Noise factor and noise figure

The noise factor is a parameter that shows the degrading of a system's signal-to-noise ratio (SNR), when a signal is applied at its input [13].

$$nf = \frac{SNR_{in}}{SNR_{out}} \quad (15.1)$$

The noise figure is the decibel value of the noise factor:

$$NF = 10 \log(nf) \quad (15.2)$$

The noise figure can also be expressed as a linear combination between the receiver's sensitivity (minimum detectable signal level)— $P_{in,min}$ , the occupied bandwidth,  $B$ , and the minimum SNR accepted at the output,  $SNR_{min}$  [13]:

$$NF[dB] = P_{in,min}[dBm] - SNR_{min}[dB] + 174 \frac{dBm}{Hz} - 10 \log B \quad (15.3)$$

The global noise factor of a cascaded n-blocks system is calculated using Friis' equation:

$$nf = nf_1 + \frac{nf_2 - 1}{G_1} + \frac{nf_3 - 1}{G_1 G_2} + \dots + \frac{nf_n - 1}{G_1 G_2 \dots G_{n-1}} \quad (15.4)$$

where  $nf_i, i = \overline{1, n}$  and  $G_i, i = \overline{1, n}$  are the noise factor and gain of block  $i$ . The first block has a crucial influence over the entire system, so that a careful distribution of the noise factor and gain of the blocks is necessary. A possible solution is to choose a low noise factor and high gain for the first block.



- Nonlinearity parameters

The 1-dB compression point, together with second- and third-order intercept points (IP2 and IP3), is the parameter that outlines the nonlinearity of the system and is a direct effect of the nonlinearities of the RF devices. Intermodulation products and/or superior order harmonics occur, which will add to the output signal.

IIP3 is computed as follows [13]:

$$\text{IIP}_3[\text{dBm}] = P_{\text{in}}[\text{dBm}] + \Delta P/2[\text{dB}] \quad (15.5)$$

where  $P_{\text{in}}$  is the input signal's power and  $\Delta P$  is

$$\Delta P[\text{dB}] = P_{\text{in}}[\text{dBm}] - \text{IM}_3[\text{dBm}] \quad (15.6)$$

where  $\text{IM}_3$  is the third-order intermodulation product, with a power level given by:

$$\text{IM}_3[\text{dBm}] = P_{\text{in,min}}[\text{dBm}] - \text{SNR}_{\text{min}}[\text{dB}] - M[\text{dB}] \quad (15.7)$$

where  $P_{\text{in,min}}$  is the sensitivity of the receiver and  $M$  is the margin.

A similar computational flow is available for computing IIP2 [13]:

$$\text{IIP}_2[\text{dBm}] = P_{\text{in}}[\text{dBm}] + \Delta P[\text{dB}] \quad (15.8)$$

where  $P_{\text{in}}$  is the input signal's power and  $\Delta P$  is

$$\Delta P[\text{dB}] = P_{\text{in}}[\text{dBm}] - \text{IM}_2[\text{dBm}] \quad (15.9)$$

where  $\text{IM}_2$  is the second-order intermodulation product, with a power level given by:

$$\text{IM}_2[\text{dBm}] = P_{\text{in,min}}[\text{dBm}] - \text{SNR}_{\text{min}}[\text{dB}] - M[\text{dB}] \quad (15.10)$$

where  $P_{\text{in,min}}$  is the sensitivity of the receiver and  $M$  is the margin.

When computing the global IIP2 and IIP3 for cascaded systems, the following equation is used [13]:

$$\frac{1}{\text{IIP}_k^2} = \frac{1}{\text{IIP}_{k,1}^2} + \frac{G_1}{\text{IIP}_{k,2}^2} + \dots + \frac{G_1 G_2 \dots G_{n-1}}{\text{IIP}_{k,n}^2} \quad (15.11)$$

where  $\text{IIP}_{k,i}$ ,  $i = \overline{1, n}$  and  $G_i$ ,  $i = \overline{1, n}$  are the  $k$ -order input intercept point, respectively, the gain of block  $i$ , and  $k$  is 2 or 3.

### 15.3.2 *Methods and Instruments for System-Level Design and Testing of RF Receivers*

The complete, end-to-end design of an RF receiver FE, after architecture selection (discussed in Sect. 15.2), implies the distribution of parameters over each block and finally, circuit-level design, and layout.

One of the most tedious tasks of this process is an optimal distribution of gain, noise figure, and nonlinearity parameters along the receiver chain. Next, the selection of the intermediate frequency/frequencies in the case of a low-IF or double-conversion architecture is another choice that has to be made by the designer.

Current trends in system-level design converge toward developing tools that automatically generate the parameters of each block, based in global specifications. An example of how these global specifications are defined for the wideband code division multiple access (WCDMA) standard, is presented in Table 15.3 [9].

RF FE blocks can be modeled using two approaches:

- behavioral/black box modeling—does not require knowledge about how the block works, is suited for system-level analysis
- physical modeling—requires knowledge about how the block works, is suitable for circuit-level analysis.

### 15.3.3 *The Hybrid Fuzzy-GA Solution to Parameter Distribution*

The block diagram of the parameter distribution system (PDS) is presented in Fig. 15.7 [7]. The system was developed in MATLAB, using the predefined functions for fuzzy sets and GA.

The inputs are the global values for gain (dB), noise figure (dB), IIP3 (dBm), IIP2 (dBm). The system is currently designed for five blocks, but can easily be extended to accept any number of blocks. PDS returns the distributed values of the parameters across the blocks. As each block is defined by four parameters, the output array will be  $4 * no. \text{ of blocks}$  long. In order to associate proper values for

**Table 15.3** Receiver specifications example—WCDMA [9]

Specification	Value
Noise figure (NF)	$\leq 9$ dB
IIP2 (@10 MHz)	$\geq -16$ dBm
IIP2 (@15 MHz)	$\geq 8$ dBm
IIP3 (@10/20 MHz)	$\geq -17$ dBm
Image rejection (@ >85 MHz)	$> 84$ dB

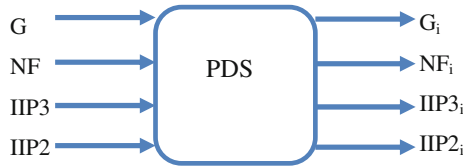
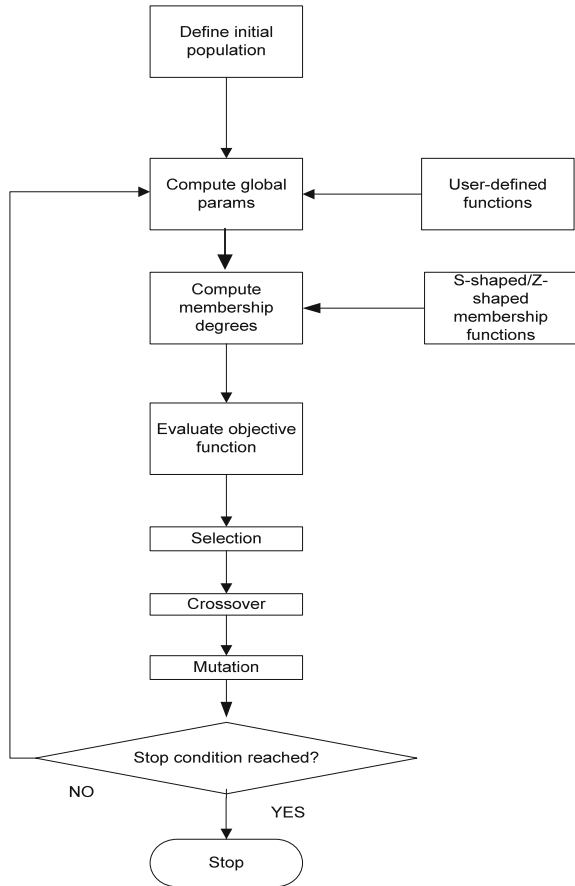


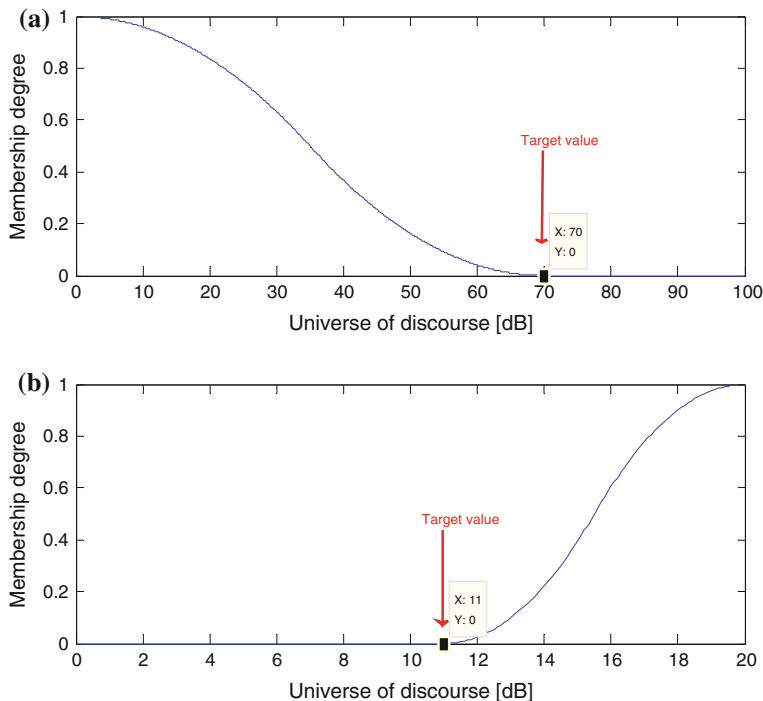
Fig. 15.7 Block diagram of PDS [7]

Fig. 15.8 PDS workflow [7]



gain, noise figure, IIP3, and IIP2 for each block, the system uses the workflow in Fig. 15.8 [7].

Since the global values of the parameters are defined as *larger than* (>) or *smaller than* (<), the handiest way to translate these conditions into fitness function material for the GA was to use fuzzy sets. As a numerical example, let us assume that gain should be above 70 dB, so 70 dB is the target value. Every value higher than 70 dB is suitable, which leads us to using a Z-shaped membership function



**Fig. 15.9** a Z-shaped membership function, b S-shaped membership function [7]

(Fig. 15.9a) [7]. Everything less than 70 dB has a membership degree higher than 0, whereas all values over 70 dB have a 0 membership degree, which is what we are looking for. For parameters that are described as *smaller than*, S-shaped membership functions are used (Fig. 15.9b) [7]; each point below the target value has a null membership degree.

Computing the global values of the parameters is achieved using several user-defined MATLAB functions and uses linear domain values, so the conversion from dB/dBm to linear scale and vice versa was performed in each of the functions.

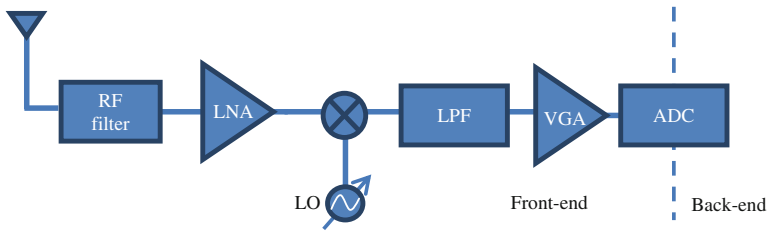
The GA tries to minimize the objective function described as follows [7]:

$$f(x) = \mu_{\text{Gain}}(x) + \mu_{\text{NF}}(x) + \mu_{\text{IIP3}}(x) + \mu_{\text{GHP2}}(x) \tag{15.12}$$

where  $\mu_{\text{Param}}(x)$  is the membership degree of point  $x$  to the fuzzy set Param, where Param = {Gain, NF, IIP2, IIP3}. Each individual of the population consists of 20 variables [7]:

$$[G_{1..5} \text{NF}_{1..5} \text{IIP3}_{1..5} \text{IIP2}_{1..5}] \tag{15.13}$$

The first 5 genes are the gains for each block; the next 5 are the noise figures, followed by 5 values for IIP3 and 5 values for IIP2.



**Fig. 15.10** RF receiver simplified model—WCDMA/WLAN [7]

Several linear inequality constraints are defined, considering that the five blocks in the receiver FE simplified model are, in order RF filter, LNA, mixer, baseband (BB) filter, and VGA (Fig. 15.10) [7]. These constraints should be changed accordingly when using the blocks in a different order.

The fitness value associated to each individual is the result of the objective function, which should be as close to 0 as possible. The individual is as fit as the value of the objective function is low. Its chances of survival in the next generation are thus increased.

Although the proposed method does not take into account parameters such as the dynamic range or effective number of bits of the ADC, it represents a solid starting point in the complete design of a receiver FE.

### 15.3.4 Simulation Results

The system was tested using the specifications of two well-known communications standards, WCDMA, and WLAN (Table 15.4). 60 dB was considered a proper minimum value for the global gain, in both cases.

Figure 15.11 [7] depicts the membership functions defined for the global gain, noise figure, IIP3, and IIP2, for WCDMA. Z-shaped membership functions were used for gain, IIP3, and IIP2, whereas for noise figure, given as *smaller than*, an S-shaped membership function was defined. The universe of discourse and parameters for each membership function are listed in Table 15.5 [7].

A population of 100 individuals was used. The algorithm is set to stop if it reaches 150 runs or the value of the objective function does not change significantly for several consecutive runs.

**Table 15.4** Design specifications—WCDMA and WLAN

Specification	WCDMA	WLAN
Gain [dB]	–	–
NF [dB]	<9	<11
IIP3 [dBm]	>–17	>–5
IIP2 [dBm]	>14	>23

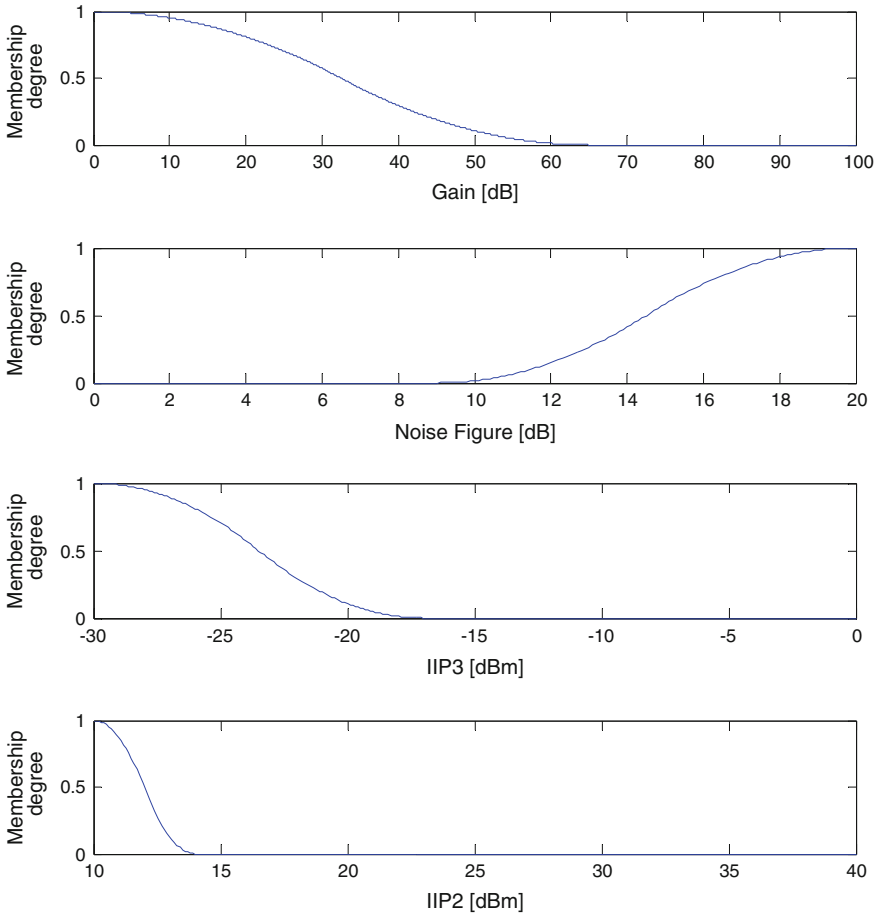
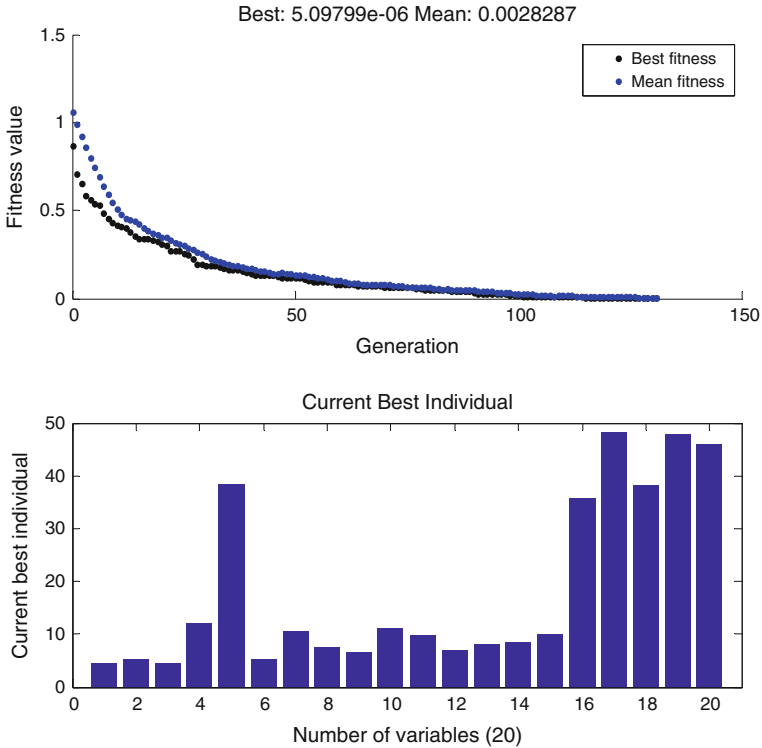


Fig. 15.11 Membership functions—WCDMA [7]

Table 15.5 Definitions of membership functions—WCDMA [7]

Name	Type	Univ. of discourse	Params
Gain	Z-shaped	[0...100]	[0 60]
NF	S-shaped	[0...20]	[9 20]
IIP3	Z-shaped	[-30...0]	[-30 -17]
IIP2	Z-shaped	[10...40]	[10 14]

The evolution of the best and mean values of the objective function and a bar chart of the current best individual for WCDMA is depicted in Fig. 15.12 [7] for WCDMA and Fig. 15.13 [7] for WLAN. Several runs showed that 150 runs for WCDMA and 250 runs for WLAN are enough for the GA to converge.



**Fig. 15.12** Best/mean fitness function evolution; current best—WCMDA [7]

The final values for the distributed parameters are listed in Table 15.6 [7]. Based on the bar charts in Figs. 15.12 and 15.13 and on Table 15.6, it can be observed that the gain of the fifth block, the VGA, is the highest, while the gain of the mixer is the lowest and very close to the gain of the RF filter, in both cases. The values of IIP2 are very high, for each of the five blocks, as expected.

Table 15.7 [7] compares the global values to the specified ones, pointing out that the system provides very good results, in both cases. Also, it is worth mentioning that the entire process lasts only a few seconds.

### 15.3.5 Genetic Algorithms-Based Techniques in Analog Circuits Design

This section is dedicated to circuit-level contributions. The method proposed for the design of analog circuits [5] uses multiobjective optimization with GA and includes the call to a circuit simulator, in the optimization loop. The solution is tested on the

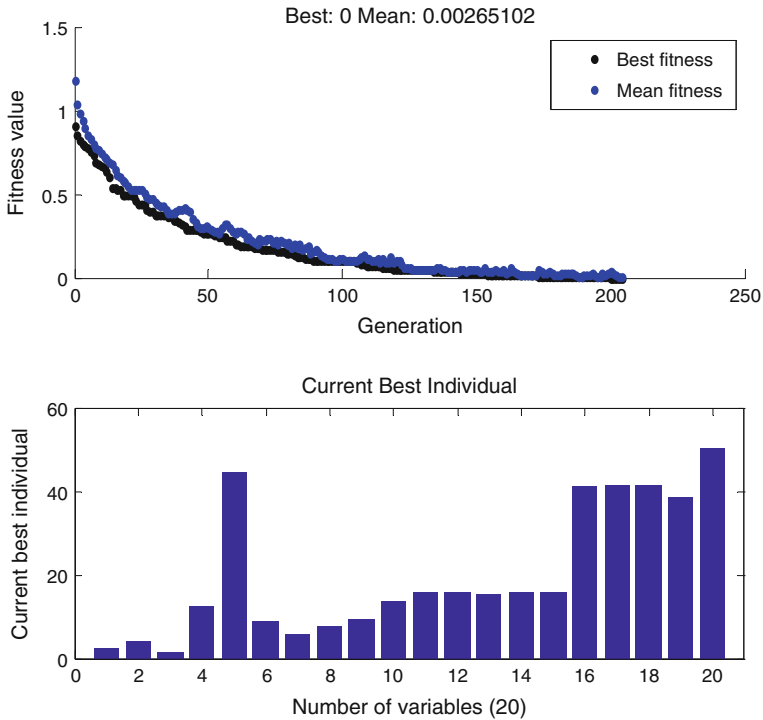


Fig. 15.13 Best/mean fitness function evolution; current best—WLAN [7]

Table 15.6 Final values of the distributed parameters [7]

Param	WCDMA					WLAN				
	RF filter	LNA	Mixer	LPF	VGA	RF filter	LNA	Mixer	LPF	VGA
Gain [dB]	4.57	5.15	4.49	12.18	38.50	2.54	4.17	1.31	12.36	44.68
NF [dB]	5.34	10.49	7.42	6.38	11.14	9.06	5.97	7.65	9.32	13.74
IIP3 [dBm]	9.76	7.14	8.02	8.62	9.96	15.66	15.69	15.56	15.75	15.98
IIP2 [dBm]	35.93	48.41	38.15	47.85	45.92	41.14	41.38	41.33	38.67	50.15

design of a symmetric operational transconductance amplifier (OTA); specific implementation issues are also described.

The process of designing analog integrated circuits is a very difficult and complex task, given the high number of requirements that need to be satisfied and the conflicts that occur between them. Therefore, designers have to choose which objectives to fully accomplish or leave aside, which leads to a permanent trade-off between specifications.



**Table 15.7** Design specifications and results [7]

Specification	WCDMA		WLAN	
	Specs.	Result	Specs.	Result
Gain [dB]	–	64.9	–	64.4
NF [dB]	<9	8.81	<11	10.9
IIP3 [dBm]	>–17	–16	>–5	–4.8
IIP2 [dBm]	>14	18.75	>23	25.5

Manual circuit design is usually based solely on the designer’s previous experience and knowledge. When dealing with complex circuits, with a lot of requirements, manual design becomes difficult and obsolete.

### 15.3.6 The Proposed Design Optimization Method

This section proposes an automated design optimization method for an analog circuit. The MATLAB environment is used both to control the entire process and to run the GA. The evaluation of each individual is performed by simulation, using an external (industrial) simulator in the optimization loop. The simulator is called in each iteration for every individual.

The flowchart in Fig. 15.14 summarizes the operation of the design optimization process, which makes use of both MATLAB and SPICE environments, in order to find the best solution [5]. At first, the initial population is generated. Next, for each individual, MATLAB will create a netlist file, which is used by SPICE to simulate the circuit. The simulator is launched from MATLAB for each individual.

The results of the simulation are written in the output files, which are read from MATLAB. From these files, the circuit performance is extracted and used to compute the objective functions. The selection automatically includes the elite individuals, in the sense that it uses multiobjective ranking, by ranking the individuals in one non-dominated category and one or more dominated categories (ranks).

Tournament selection and intermediate crossover are used together with mutation to generate the new population. The algorithm stops if the stop condition is reached (maximum number of generations or nonsignificant improvement in the fitness function value, over a certain number of generations), or else, it goes back to the first step.

The result consists in the non-inferior (non-dominated) solution points, also called Pareto optima. The user has the freedom to make the final decision choosing one solution from the optimal Pareto front.

The additional functions used to create the netlist file, launch the SPICE simulator, and extract the necessary data for the algorithm to work were defined in the MATLAB environment and are external to the multiobjective GA optimization tool.

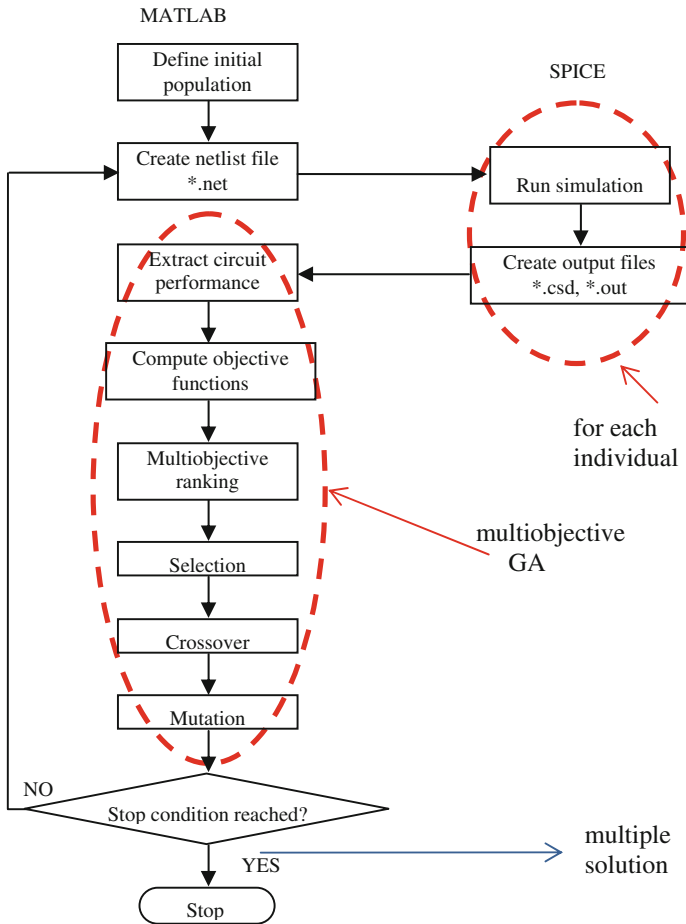


Fig. 15.14 Flowchart of the design optimization method [5]

### 15.3.7 Brief Circuit Description of the Symmetric OTA

The method described above is tested for the design optimization of a symmetric CMOS OTA (Fig. 15.15) [14], operating on a capacitive load  $C_L$ .

A symmetric OTA consists of one differential pair ( $M_1, M_2$ ) and three current mirrors ( $M_7$ – $M_8, M_3$ – $M_5$ , and  $M_4$ – $M_6$ ). The input differential pair is loaded with two identical current mirrors, which provide a current gain  $B$ .

Generally speaking, the design parameters for the circuit are the channel size ( $W$  and  $L$ ) of all transistors, the biasing current  $I$  and the gain of the current mirror,  $B$ . However, for this circuit, some simplifications appear [5]. The sizing process of the transistors consists in finding appropriate values for their widths:  $W_1 = W_2$  s the

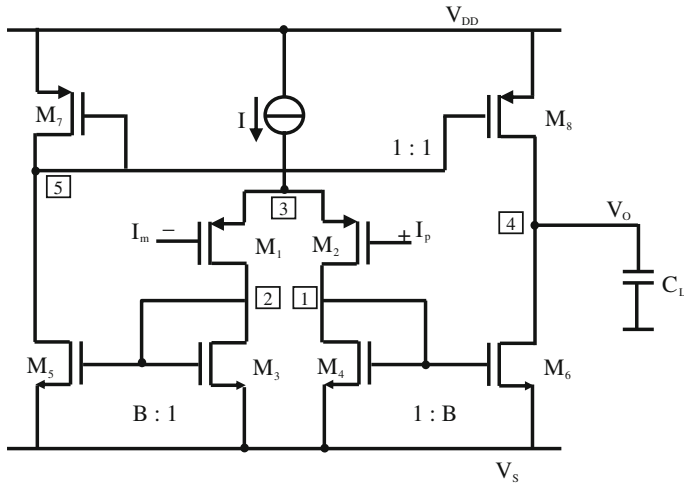


Fig. 15.15 Symmetric operational transconductance amplifier [14]

width of the transistors  $M_1$  and  $M_2$ ,  $W_3 = W_4$  is the width of the transistors  $M_3$  and  $M_4$ , and finally  $W_7 = W_8$  is the width of the transistors  $M_7$  and  $M_8$ .

The gain at low frequencies is computed using:

$$A_{v0} = \frac{v_o}{v_i} = \frac{i_6 R_{N4}}{v_i} = \frac{B i_2 R_{N4}}{v_i} = \frac{B g_{m1} v_i R_{N4}}{v_i} = g_{m1} B R_{N4} \tag{15.14}$$

where  $R_{N4}$  is the equivalent output resistance at node 4.

$$A_{v0} = 2\sqrt{K_p} \frac{V_{Ep} L_8 V_{En} L_6}{V_{Ep} L_8 + V_{En} L_6} \sqrt{\left(\frac{W}{L}\right)_1} \frac{1}{\sqrt{I}} \tag{15.15}$$

The circuit has only one high-resistance node (node 4) where the gain is large, the swing is large, and ultimately where the dominant pole is formed, giving the bandwidth (BW) of the amplifier [14].

$$BW = \frac{1}{2\pi R_{N4} C_L} \tag{15.16}$$

$$BW = \frac{1}{4\pi C_L} \frac{V_{Ep} L_8 + V_{En} L_6}{V_{Ep} L_8 V_{En} L_6} I B \tag{15.17}$$

The layout area can be simply computed using the formula:

$$\text{Area} = \sum_{i=1}^N W_i L_i \quad (15.18)$$

where  $W_i$  and  $L_i$  are the width and the length of the  $i$ th MOS transistor.  $N$  is the number of transistor, in this design  $N = 8$ .

The power dissipation can be computed as follows:

$$P = (V_{DD} - V_{SS})(I + BI) = (V_{DD} - V_{SS})(1 + B)I. \quad (15.19)$$

### 15.3.8 Simulations and Results

Four design parameters were considered: width of the transistors  $M_1$  and  $M_2$  ( $W_1 = W_2$ ),  $M_3$  and  $M_4$  ( $W_3 = W_4$ ),  $M_7$  and  $M_8$  ( $W_7 = W_8$ ) and the biasing current  $I$ :

$$[W_1 W_3 W_7 I]$$

The design optimization is subject to the specifications in Table 15.8 [5].

The design considers the nanometric 180-nm process. The objective functions represent the absolute errors, with respect to the desired values, with a minimum value of 0, if the specification is fulfilled. The channel length is set to  $L_1 = L_2 = 1 \mu\text{m}$  to be able to obtain some gain.

For the current mirror ( $M_4$ – $M_6$ ,  $M_3$ – $M_5$ ), the mirrored current should be as close as possible to the value  $B$  times the reference current. For the design,  $L_3 = L_4 = L_5 = L_6 = 10 \mu\text{m}$  was set.

In order to keep the biasing current source operational, a linear inequality constraint between two design parameters was found [5]:

$$\left(\frac{W}{L}\right)_1 > 0.036 \times (\text{numerical value of } I[\mu\text{A}]) \quad (15.20)$$

The algorithm was tested for a population size of 60 individuals, over 100 generations. Other GA settings include lower and upper boundaries for the design parameters and specific values for crossover and selection sizes.

The final Pareto front consists of 21 individuals [5] (Table 15.9). Given that there are four objectives to be met, a trade-off between them is inherent, when choosing the best solution.

Considering the gain and bandwidth specifications, there is one individual that satisfies the former objective (no. 7), and four that meet the latter (no. 2, 9, 15, and 21). From those four, the one with the highest gain is individual no. 9.

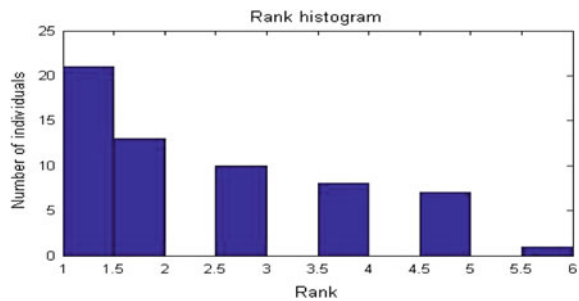
**Table 15.8** Design specifications [5]

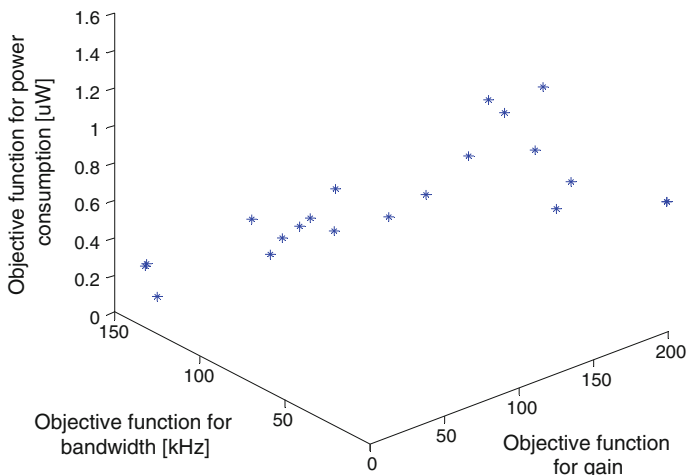
No.	Performance	Desired value
1	Gain	>200
2	Bandwidth	>150 kHz
3	Layout area	Minimized
4	Power consumption	Minimized

**Table 15.9** Individuals of the final Pareto front—values and specifications [5]

Indiv.	W <sub>1</sub> [μm]	W <sub>3</sub> [μm]	W <sub>7</sub> [μm]	I [μm]	Gain	Bandwidth [kHz]	Layout area [μm <sup>2</sup> ]	Power consumption [μW]
1	1	10	1	1	97.70	1.54	604	9.19
2	6.21	10.38	1	86.85	0.88	102329299.23	637.41	698
3	3.24	601.17	13.38	56.39	23.36	104.71	36103.75	517
4	45.57	154.93	7.23	151.51	89.48	123.02	9401.83	1380
5	47.73	789.65	35.01	27.78	115.12	23.98	47544.68	256
6	37.39	618.22	26.24	97.9	72.12	95.49	37220.98	897
7	44.36	27.41	1.01	37.05	204.04	18.19	1735.62	335
8	47.18	223.88	34.39	56.63	95.33	69.18	13596.14	519
9	49.69	96.80	9.82	171.71	83.63	158.48	5927.61	1560
10	43.53	360.82	12.20	140.47	78.55	123.02	21760.86	1280
11	27.32	128.28	1.46	50.15	141.44	28.84	7754.59	455
12	46.14	282.22	22.88	124.93	75.21	138.03	17071.52	1140
13	47.22	669.75	21.51	37.43	112.40	31.62	40322.78	344
14	49.58	801.48	15.09	63.87	104.51	45.70	48218.48	586
15	3.87	10.01	1	86.51	0.676	102329299.23	610.55	695
16	22.27	476.32	32.36	27.19	88.84	31.62	28688.76	250
17	43.32	856.66	1.18	31.22	191.11	10.96	51488.91	284
18	11.74	926.22	24.96	47.02	51.61	52.48	55646.91	432
19	49.41	1000	38.21	40.49	105.74	32.35	60175.26	372
20	41.68	10.09	1	1	173.13	1.12	691.26	9.19
21	44.37	254.57	99.99	108.51	64.76	158.48	15563.19	994

**Fig. 15.16** Rank histogram for the optimal Pareto front [5]





**Fig. 15.17** 3D plot for gain, bandwidth, and power consumption [5]

The plot in Fig. 15.16 [5] displays the ranking of the individuals, in the final iteration. The Pareto front consists of the 21 non-dominated individuals, that is, individuals that have rank 1.

In order to display the results considering all four objectives, a 4D plot is needed. Since this is not possible, a 3D plot (Fig. 15.17) [5] is chosen to exemplify the compromise between three objectives, namely gain, bandwidth, and power consumption.

## 15.4 Conclusions

Technological advances of the past decade have shown that traditional (manual) circuit design has become obsolete. These days, automated processes are more and more used in the design loop, making the entire process fast and less faulty. However, in many cases, the intervention of a human expert is still needed, for example, when the design process comes up with more than one possible solution.

The final choice remains a question of human intervention. Computational intelligence techniques, such as GA, fuzzy logic, or neural networks, are widely employed in modern circuit and system design.

Section 1 describes the implementation of a fuzzy expert system, able to indicate the possible receiver architecture, given some minimal information at the input.

The developed system is a rule-based expert system, according to Turban's classification. It represents a step forward into bringing the theoretical fundamentals of receiver architectures closer to the inexperienced user. Although the system is

only in prototype phase, it comes as a solid foundation for an expert system capable to fully design a receiver, together with the methods to be described in Sect. 15.2.

The fuzzy expert system for architecture selection can be seen as a powerful educational tool, as it brings the theoretical fundamentals of receiver architectures closer to the unexperimented user. The system can be further developed by adding a graphical user interface, in order to improve the user–system interaction.

Section 15.2 describes the implementation of a hybrid fuzzy-genetic algorithm for parameter distribution in receiver chains. The fact that the typical global parameters of a receiver chain (gain, noise figure, intercept points) are stated as *smaller than* or *larger than*, leads us to use fuzzy sets in defining these parameters. GA come as a natural solution to minimizing an objective function, given as a sum of membership degrees. The presented solution achieves a very fast parameter distribution of the global metrics mentioned before.

Although the system is currently designed to work with five blocks, it can easily be adjusted to accept a user-given number of blocks. Experimental runs using global parameter values for WCDMA and WLAN proved that PDS is able to solve the minimization problem in a very short time.

The proposed solution can be used as a strong starting point in the process of completely designing a receiver chain. Future developments can address the flexibility of the system (e.g., introducing user-defined constraints) or its use in correlation with some other tool that shows whether the values of the distributed parameters are achievable in real implementations (e.g., connecting the system to a circuit simulator).

Section 15.3 presents the implementation of a multiobjective optimization method based on the genetic algorithm, and the results for applying it for the design optimization of a symmetric OTA. The method was used with two types of design specifications: “greater than,” for gain and bandwidth, and minimization, for layout area and power consumption. The MATLAB toolbox implemented for this method has the ability to control the whole process, to run the GA, to create the circuit netlist, to run the external simulator and to post-process the simulation results, in order to evaluate the objective functions. The solutions provided by the algorithm are located on the Pareto front, revealing the trade-offs that occur between the conflicting design specifications. The designer is given the possibility to choose the final solution, from the Pareto optimal set.

The method proved to be time-consuming (approximately 10 h, on several computers) due to the large number of calls to the external simulator. Nevertheless, this time is not prohibitive for a real design task.

Given its modular structure, the method for multiobjective optimization proposed in this section can easily be adjusted to fit other design specifications or to automate the design process of a different and more complex circuit.

**Acknowledgments** This paper was supported by the project “Improvement of the doctoral studies quality in engineering science for development of the knowledge based society-QDOC” contract no. POSDRU/107/1.5/S/78534, project co-funded by the European Social Fund through the Sectorial Operational Program Human Resources 2007–2013.

## References

1. Circa, R.: Study on Resistive mixer circuits in reconfigurable mobile communications systems. Ph.D. thesis, Technische Universitat Berlin, Germany (2008)
2. de Llera Gonzalez, D.R.: Methodologies and tools for the design and optimization of multi-standard radio receivers. Ph.D. thesis, KTH School of Information and Communication Technology, Stockholm, Sweden (2008)
3. Crols, J., Steyaert, M.: Low-IF topologies for high-performance analog front ends of fully integrated receivers. *IEEE Trans. Circuits Syst. II Analog Digital Signal Proces.* **45**(3), 269–282 (1998)
4. Duenas, S.A.R.: Multi-band multi-standard CMOS Receiver front-ends for 4G mobile applications. Ph.D. thesis, KTH Royal Institute of Technology, Stockholm, Sweden (2009)
5. Eberhart, R., Shi, Y.: Computational Intelligence. Concepts to Implementations. Elsevier, Morgan Kaufman Publisher, Los Altos, CA (2007)
6. Ivanciu, L., Oltean, G. et al.: Design illustration of a symmetric OTA using multiobjective genetic algorithms. In: Springer Lecture Notes in Computer Science, vol. 6883, pp. 443–452. Springer, Berlin (2011)
7. Ivanciu, L.: A Survey on computational intelligence techniques used in LNA design. *IOSR-JECE* **8**(3), 41–46 (2013)
8. Ivanciu, L., Oltean, G.: Parameter distribution in receiver chains: a hybrid fuzzy-genetic algorithm approach. *ACTA Technica Napocensis Electron Telecommun.* **54**(4), 28–33 (2013)
9. Ivanciu, L.: Contributions to the design of telecommunications receivers using computational intelligence techniques. Ph.D. thesis, Technical University of Cluj-Napoca, Romania (2014)
10. Jensen, O.K., Kolding, T.E., et al.: RF Receiver Requirements for 3G WCDMA Mobile Equipment—Technical Feature. Aalborg University, RISC Group, Aalborg, Denmark (2000)
11. Kundert, K.: Accurate and Rapid Measurement of IP2 and IP3. The Designer's Guide Community (2009)
12. Mikkelsen, J.M.: Front-End Architectures for CMOS Radio Receivers, Division of Telecommunications. Aalborg University, Denmark (1996). [http://www.inst.bnl.gov/~poc/wireless/rf\\_architectures.pdf](http://www.inst.bnl.gov/~poc/wireless/rf_architectures.pdf). Accessed 22 June 2014
13. Razavi, B.: RF Microelectronics, 2nd edn. Prentice-Hall, Englewood Cliffs, NJ (2011)
14. Sedra, A.S., Smith, K.C.: Microelectronic Circuits, 6th edn. Oxford University Press, Oxford (2011)
15. Whitley, D.: An overview of evolutionary algorithms: practical issues and common pitfalls. *J. Inf. Softw. Technol.* **43**, 817–831 (2001)



# Chapter 16

## Enhancing Automation in RF Design Using Hardware Abstraction

Sabeur Lafi, Ammar Kouki and Jean Belzile

**Abstract** This chapter presents advances in automating RF design through the adoption of a framework that tackles primarily the issues of automation, complexity reduction, and design collaboration. The proposed framework consists of a design cycle along with a comprehensive RF hardware abstraction strategy. Being a model-centric framework, it captures each RF system using an appropriate model that corresponds to a given abstraction level and expresses a given design perspective. It also defines a set of mechanisms for the transition between the models defined at different abstraction levels, which contributes to the automation of design stages. The combination of an intensive modeling activity and a clear hardware abstraction strategy through a flexible design cycle introduces intelligence, enabling higher design automation and agility.

### 16.1 Introduction

The need for radio systems is in constant growth due to the particular success of consumer communication services. The wide adoption of cellular and wireless systems in the last decades is particularly driving the ICT market, giving birth to new applications and services (e.g., machine-to-machine and over-the-top services) and fueling the increasing convergence between fixed- and mobile-broadband communications. Naturally, end-user expectations in terms of quality of service are evolving. At affordable costs, it is expected that future radio systems provide higher data rates and lower power consumption in increasingly harsher radio environments where spectrum is getting more crowded and regulations are becoming tougher.

---

S. Lafi (✉) · A. Kouki  
LACIME, ÉTS, Montréal, Canada  
e-mail: sabeur.lafi.1@ens.etsmtl.ca

J. Belzile  
MEIE, Québec, Canada

In order to keep pace with the emerging requirements, the challenges that should be addressed are related to implementation technology and radio design flows. On technology level, most future radios will be built with multi-standard, multi-band, and multi-mode transceivers to provide a seamless connectivity to various mobile and wireless networks. This requires higher processing capability for baseband stages and more robust radiofrequency (RF) front ends in order to support multiple communication standards and accommodate various radio transmission scenarios. Higher levels of miniaturization and integration are also needed to keep the form factor within an acceptable range for consumers. In addition, all these should have a very-low-energy-consumption profile. Remarkable efforts are being deployed in both industry and academia in order to come up with relevant solutions that effectively address these issues. But, is this enough to leverage the encountered challenges? While a myriad of new technologies are being developed to enhance radio systems capability (i.e., the “what-to-do”), less interest is dedicated to design approaches and tools (i.e., the “how-to-do”).

On radio design level, there are particularly wide disparities between digital baseband and RF front-end design cycles. In digital design, it is possible to integrate very complex circuits during a reasonable time frame. Digital designers have adopted a structured design approach that is backed by a set of tools allowing the automation of most design steps from concept to prototype. This approach builds up the circuit hierarchically: it is considered as a collection of modules. Each module is a collection of cells, and each cell is composed of some transistors and lumped components. Each module or cell implements a logical functionality and can be reused as much as required. Thus, the design effort is reduced. The main concept behind this useful representation is hardware abstraction. Every component is used as black-box model. At each abstraction level, the designer deals only with the models available at that level. Given enough data about their functionality, the designer can use these models without knowing their internal structure. The characteristics of their underlying components are virtually masked. Complexity is thus reduced and mastered. These paradigms led to the implementation of mature digital design tools, which played a key role in rising design productivity via modeling and automation.

On the contrary, the classic RF design scheme still starts at circuit level and is mostly manual and very technology dependent. It presents various discontinuities between design stages and lacks formal communication rules between the different developers involved in the same RF design project. Consequently, the exchange of data and collaboration abilities are still limited. Actually, the conventional design flow is too costly, long, and not amenable for easy technology insertion. Design reuse is also limited. The changes and corrections of the design according to new specifications are often expensive and time-consuming. Final system integration is tedious, risky, and slow particularly when different technologies are involved in the system architecture. Despite recent notable advances, most RF tools do what they are best at. There is a lack of tools able to carry out system-level analyses, tackle growing design complexity, support multiple technologies, allow cost-effective co-design especially in mixed-signal context, and ensure reliable formal verification

at the different design stages. The absence of clear abstraction levels and coherent functional modeling is a major hindrance to current RF design practice.

In light of these observations, this chapter presents advances in automating RF design through the adoption of a framework that tackles primarily the issues of automation, complexity reduction, and design collaboration. The proposed framework consists of a design cycle along with a comprehensive RF hardware abstraction strategy. Being a model-centric framework, it captures each RF system using an appropriate model that corresponds to a given abstraction level and expresses a given design perspective. It also defines a set of mechanisms for the transition between the models defined at different abstraction levels, which contributes to the automation of design stages. The combination of an intensive modeling activity and a clear hardware abstraction strategy through a flexible design cycle brings enough intelligence enabling higher design automation and agility. The chapter is concluded by a design example, which illustrates how all the presented concepts can be practically applied to RF design.

## 16.2 Overview of Modern RF Design Practice

To better understand the motivation for a different approach in RF design, a quick overview of specificities of RF domain is presented followed by a brief study of common RF design approaches to highlight the limitations and shortcomings of modern RF design practice.

### 16.2.1 *Particularities of RF Design*

Radio frequencies refer to alternating current (AC) signals whose frequencies are ranging from 30 MHz up to 300 MHz. Microwaves refer to those with frequencies ranging from 300 MHz to 300 GHz. Generally, RF engineering covers the design of radio front ends that use radio waves whose frequencies lay in RF spectrum [1]. However, RF and microwave design are very similar in terms of design approaches and tools. In wireless and mobile radios, RF front ends make the link between the digital baseband and the immediate radio environment (e.g., base stations, hot spots, and other radios). While the design of the RF front end is a minor portion of the whole communication system design, it nonetheless poses significant challenges at various levels due to the specificities of RF domain [2].

Successful RF design is frequently the result of mastering a variety of disciplines [3, 4]. It often requires advanced knowledge and good skills in various topics including radio environment analysis, basic communication theory, design flows, and tools as well as standards and regulations. It is a field where a single technology can rarely be used alone. In fact, different RF components often built in different technologies are required to build a working front end. The number and type of

these components vary depending on the selected architecture. Each component accomplishes a given functionality (e.g., filtering, signal routing, or amplification). The resulting mix of technologies is challenging because it is the source of many design problems (e.g., integration, heat dissipation, and shielding). For example, some components such as surface-mount devices (SMD) cannot be built on the same substrate on which the baseband circuitry is implemented. This may turn into a serious design bottleneck especially when a small form factor is required.

Furthermore, RF systems are highly sensitive to impairments and nonlinearities. They are also very sensitive to noise, signal distortion, and inherent radio impairments more than digital circuits. Each RF component adds an amount of noise to the signal. The cumulative value of added noise depends on the way the RF components are mounted. It reduces the front end's signal-to-noise ratio that impacts negatively the data rate. Signal distortion occurs due to the imperfections and the nonlinearity of RF devices. The latter produces both inband and out-of-band unwanted signals (i.e., intermodulation products) that may interfere with desired signals and contribute to environment interferences. Reducing the impact of noise and mitigating the effect of inherent radio impairments are essential to come up with a successful RF design.

In addition, it is a discipline of compromises and trade-offs. Designers are continuously asked to find out compromises between antagonistic design variables. The "RF Design Hexagon" given in [3] depicts six among the mutually dependent variables (i.e., frequency, power, DC supply voltage, gain, noise, and linearity) that impact the most RF design.

## ***16.2.2 Common Design Approaches: Basics and Limitations***

For decades, designers attempted to use several design approaches in order to alleviate RF domain constraints and come up with good front ends. Despite the notable progress in design approaches and tools, various limitations continue to hinder the evolution of radio design.

### (a) Traditional design approaches

Historically, the design approaches of RF systems did not remarkably change. The following two main RF design flows (and some of their variants) were prevalent:

- *Bottom-up design flow*: The bottom-up design flow starts with the design of individual blocks that are assembled to form a more complex block. These blocks are gradually combined to form the final system. In practice, each individual block is implemented all the way to the lowest available circuit level (e.g., transistor level) according to a given set of specifications. Next, the block is separately tested and verified. Finally, all blocks are gathered, assembled, and

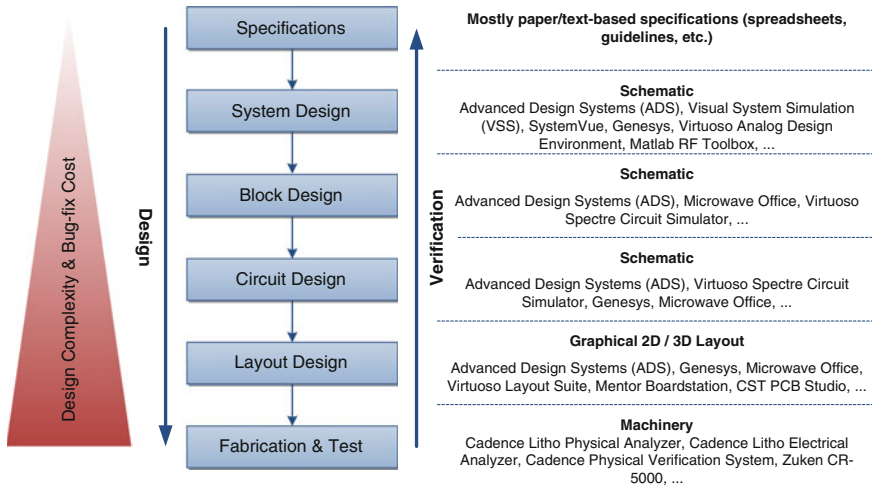
verified together. The entire system ends up generally with the lowest abstraction level representation [5].

- *Top-down design flow*: The top-down design flow starts with the whole system concept and then recursively breaks it down into smaller pieces, easy to implement, test, and validate. The design level at which this approach starts is referred to as the system level [6]. At top level, the architecture is defined as a block diagram that is iteratively refined and optimized to meet the specifications. The specifications of the underlying blocks are then derived from system-level architecture simulation. Once all blocks and sub-blocks are individually designed and verified, the overall system is assembled and verified against the original requirements.
- *Other approaches*: The previous design approaches suffer from some limitations [5, 7]. To combine the strengths of both, various mix were suggested. For example, [6] proposed a hybrid design flow commonly known as the “V” diagram, which combines a top-down design flow along with a bottom-up verification process. The top-down flow proceeds with the design from the system level through to the transistor level while the bottom-up verification process starts at the layout level and proceeds up to the highest levels. For performance-critical applications, the performance-driven design was proposed [8]. This approach consists of the alternation of a top-down flow, used in design, and a bottom-up flow, used in verification at all design stages. At each level, the system is subdivided into sub-blocks to be implemented (i.e., topology selection). These sub-blocks are sized, optimized, and verified against the performance specifications (i.e., specification translation). Once the sub-blocks are assembled at the same level, the new assembly is verified (i.e., layout generation and extraction). This approach ensures that the designed sub-blocks meet always the performance constraints before going farther in the design flow.

## (b) Typical design flows

RF design is an iterative process whose purpose is building up a RF front end that meets certain specifications. In general, a RF designer begins with the search of a system-level solution that meets the requirements. This process is commonly referred to as “Design Space Exploration.”

At system level, a candidate solution is evaluated based on its performance. Metrics such as noise, frequency response, and matching are used for this purpose. Once an initial solution is selected, the designer proceeds step-by-step according to a design scheme in order to optimize that solution and implement a prototype. To illustrate this philosophy, Fig. 16.1 shows a typical RF design scheme. In fact, the candidate solution is subdivided into smaller blocks. Each block is composed of a single or a collection of components. These blocks are often implemented separately by different teams. If required, these blocks might also be broken down into smaller pieces for more optimization. The specifications of each block are derived from the system-level initial requirements. In most cases, there are no formal



**Fig. 16.1** Typical RF design scheme and some EDA tools in use for each design stage

methods to validate a priori this kind of specifications [4]. Once the specifications of each block (generally in text or spreadsheet formats) are known, every designer proceeds with the implementation of a given block at the circuit level. At this step, circuits are captured using a schematic capture tool. The circuits consist of the interconnection of RF components (e.g., lumped components and transmission lines). Each component is generally defined by either its electrical or its physical model that might be layout-, equation-, or file based. Various simulations are carried out in order to evaluate each component's performance. For this purpose, one or many electronic design automation (EDA) tools might be used at each step.

Then, designers use layout tools in order to create the block's corresponding layout. Generally, most optimizations and adjustments take place at this step where various simulation tools might be used to verify the final design performance (e.g., signal reflection, gain, noise, shielding properties, and radiation).

The next step consists of final integration, prototype manufacturing, and testing. Simulated and measured performances rarely match due to the inaccuracies in device modeling and simulation tools as well as the variability of fabrication processes. For this reason, most designers consider a margin to compensate any eventual degradation particularly in critical performance applications. Furthermore, the verification process is also iterative. Once an error is detected, the previous step is revisited. These re-spins are time-consuming. The cost of error correction may be prohibitively expensive particularly at advanced design stages such as final integration [4].

### (c) Limitations and shortcomings

Modern RF design practice is suffering from concrete limitations and weaknesses at the level of both design flows and tools.

On the one hand, the first drawback of modern RF design flows is their high technology-dependence. The lack of effective system-level models pushes designers to work instead with those available at circuit level, which makes technology the actual guide of the design process. This situation results particularly from the absence of higher levels of abstraction, hinders the effort of efficient design space exploration, reduces automation capability, and limits design reuse capability. In addition, the prevalence of circuit-level design imposes the use of a bottom-up design approach where tasks are carried out serially [7]. This added to the manual interfacing between design stages increases design flaws and errors which may be excessively expensive in time and money (particularly at the end of the design cycle). This approach is also known for the discontinuities it causes within the design cycle. In this context, designers tend to work in isolation from each other, and communication is generally poor. Furthermore, there is a lack in design flows enabling co-design of digital and analog/RF circuits. But this lack is not limited to mixed-signal design and verification, and numerous RF fabrication processes (e.g., SiP integration) have also no complete end-to-end design flows.

On the other hand, the EDA tools used at the different design stages are limited in both performance and design features. In fact, specific technologies (e.g., silicon-on-insulator) lack specialized tools [9]. Most of tools are deficient in multi-technology support especially at system level [10]. Limited modeling accuracy and simulation issues in most tools impact design quality particularly for complex systems. Additionally, essential features for fast RF design such as specifications' validation and functional-level verification are almost absent in most tools. On the contrary, physical-level verification is available in some tools, but for a limited set of technologies only. The interaction between tools and the exchange of both models and design data is limited as well [11].

All these weaknesses and shortcomings in both design flows and EDA tools in common use in RF design tend to seriously hamper the ability of automating the design process and decrease overall productivity.

### **16.3 Hardware-Abstraction-Based RF Design Frameworks for Automation and Productivity Enhancement**

To address the weaknesses of modern RF design practice, a different design philosophy becomes a requirement. In addition to make automation and productivity enhancement its central goal, specific issues such as prevalent technology predominance and sustained design complexity increase should also be effectively addressed. Similar to developments in other domains (e.g., digital design), the introduction of higher levels of abstraction is a promising response to these issues. For this purpose, a framework that consists of a refurbished five-step design cycle which is backed by a hardware abstraction strategy for RF domain is presented in the following.

### 16.3.1 A Five-step Design Cycle

Unlike the traditional RF design flow where handling technology often starts at the early design steps, the design flow of Fig. 16.2 was thought to bridge this gap. It consists of five main design steps. The first, namely “Functional Description”, is the design starting point in which the system’s specifications are captured in functional models. In the “Analysis” step, the resulting models are checked out for coherence. Then, various system-level analyses may take place. In this phase, the designer’s aim is to find out a design solution that meets the specifications. If an initial solution is found, it is thoroughly investigated, optimized, and implemented in the “Synthesis” step considering a relevant technology input. “Synthesis” is meant to be automated as much as possible. It results in a ready-to-manufacture design that is fabricated in the next step, namely “Manufacturing.” Next, the prototype is submitted to final validation in the last design step, namely “Tests and Measurements.” In the heart of this design cycle lays the Q-matrix, a multi-dimensional structure that holds electrical design data throughout the design process.

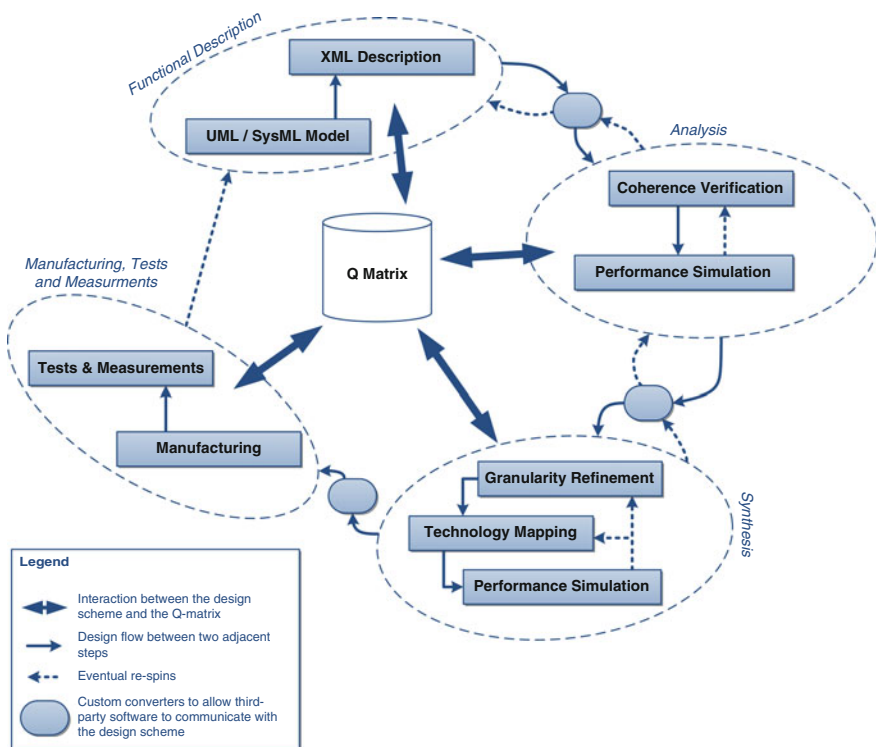


Fig. 16.2 The five-step framework for RF design



This said, this design cycle defines several mechanisms and artifacts whose primary aim is enabling higher automation, improved design collaboration, and less technology prevalence.

(a) Functional description

In traditional RF design, specifications are text-based and cannot often be validated using specialized tools. This explains in part why designers start by tweaking the physical details of their circuits looking for a suitable design solution to begin with. This task, mostly manual, may take a valuable time with no satisfactory results because the specifications may be neither coherent nor realistic. “Functional Description” addresses these issues using high-level modeling. Specifications are captured in comprehensive models, which uncouples the RF functionality from the underlying physical structure and properties. Functional models are meant to be expressed using human-readable languages in order to make them easily understandable by designers. In addition, these models can be handled by design tools for various automated operations (e.g., storage, exchange, and processing).

- Modeling RF systems using standard modeling languages

“Functional Description” turns specifications into models that capture system’s functionality and properties as well as the related design requirements and constraints. To make this effective, mechanisms enabling human-readable representations for functional description, easy storage, and exchange are required.

Traditionally, RF components are modeled in most commercial tools using graphical blocks to capture their properties and emulate their frequency response (e.g., ADS, SystemVue). In some cases, it is possible to use programming languages (e.g., C++, SystemC) and hardware description languages (e.g., VHDL, Verilog) for this purpose. Other EDA tools enable the use of script-, equation-, and file-based models.

However, all these modeling techniques are limited. Most system-level blocks are predefined and cannot be altered by the designer, which limits their accuracy and usefulness. In addition, the model readability becomes quickly poor as the complexity of the model (especially algorithm based) grows. It is also difficult to capture the system requirements (mostly text based) using algorithm-, equation-, or block-based models.

To enhance system modeling and specification capture, [12] suggested the use of standard modeling languages proven very useful in other engineering domains in order to cope with modeling issues in RF design.

In this regard, it was established in [12] that Systems Modeling Language (SysML) is a good candidate for “Functional Description” of RF systems. SysML is a general-purpose modeling language standardized by the Object Management Group (OMG) for the specification, analysis, design, and verification of systems in a broad range of engineering fields [13]. It provides graphical representations with flexible and expressive semantics allowing the design of complex engineering systems. SysML defines five structural and four behavioral diagrams. The former are dedicated for the capture of both internal and external structures of an engineering system.

The latter are used for the description of dynamic and behavioral artifacts (e.g., interactions) within that system. Two additional diagrams, namely requirement and parametric diagrams, are used, respectively, for the capture of requirements into visual representations and the definition of constraints and rules of the engineering system. In addition, SysML introduces the concept of views allowing the modeling of the same system from different viewpoints [13]. Furthermore, SysML is compliant with many standardized data interchange formats (e.g., XML, XMI, and AP-233) which makes models, data, and metadata exchange easy [13]. Considering these advantages of SysML, [12] attempted to use SysML in order to model a UMTS transceiver using the different diagrams and concepts provided within the language. The conclusion was that SysML has many assets that make it suitable for RF system's modeling. For in-depth information about SysML, the reader may refer to [13–17].

- Exchanging models via standard markup languages

The use of SysML for the modeling of RF systems enables effective “Functional Description” because it allows multi-level technology-independent representation. It also endorses design collaboration and communication between designers because it provides designers with visual graphical representations. However, graphical notations are human readable, but not relevant for machines. To improve tools interaction and ensure automated processing of functional descriptions, models should be amenable for storage and exchange. To do so, the Extensible Markup Language (XML), a simple and widely adopted markup language that is already supported by SysML, is used [18]. This standard text format is flexible and easy to use. It is both human- and machine-readable and designed for data automated processing, storage, and exchange. It is worth noting that the framework uses XML not only to facilitate the storage and exchange of SysML models but also to define the file format used by the Q-matrix.

#### (b) Analysis and Synthesis

Specifications captured in SysML models and exchanged as XML files during “Functional Description” stage are submitted to the following step, namely “Analysis.” At this level, functional models are validated against coherence rules in order to detect any errors. If no errors are identified, these models are used to carry out system-level analyses in order to figure out an initial solution that meets, in part or in whole, the specifications. The following step, namely “Synthesis,” uses this initial solution as an ingredient to find out a final one. To this end, the starting solution may be subject to additional granularity refinement. Given the appropriate technology input, further optimizations may be required at the light of an iterative series of performance simulations.

- Validating the specifications through “Coherence Verification”

The process of “Coherence Verification” is based on a set of coherence rules that are used to validate the consistency of the models developed during the “Functional Description” stage. These rules may be embedded within SysML models or provided as a separate input.

The framework defines two levels of consistency checks as follows:

- *Warnings*: This level indicates that a coherence rule has detected a minor issue within the provided functional description, which can impact the design consistency at later stages.
- *Errors*: A coherence rule results in an error when it detects a major inconsistency that makes the design practically not feasible.

To illustrate the difference between warnings and errors, Table 16.1 shows two rules that can be used for the coherence verification of a RF filter functional description.

- Enhancing system-level analysis

One of the goals of functional description is enhancing design space exploration. Given the appropriate tools and depending on the target functionality, functional models can be submitted to a myriad of system-level analyses. Being already validated after the coherence verification test, these analyses are more likely to result in a relevant initial design solution. This solution is the input to the following design stage (i.e., “Synthesis”). This said, system-level analyses are thought to be automated as much as possible. If necessary, the designer can always intervene to guide these analyses. In addition, this design stage is intended to be independent from technology details in order to allow more flexibility in the design process.

- Changing the implementation viewpoint using granularity refinement

The design requires often to be partitioned into smaller blocks and sub-blocks that can be implemented and verified separately. This technique allows to minimizing design time throughout higher design concurrency. The number of elements to which the system is partitioned is called granularity level. A coarse-grained partitioning corresponds to a low granularity level and results in fewer but larger parts. However, a fine-grained partitioning corresponds to a high level of granularity and results in more detailed subdivision. Within the “Synthesis” design stage, the step of “Granularity Refinement” allows the designer to decide about the adequate partitioning of the system under design. Changing the level of granularity allows to modify the design considerations. This results in more control of the design process, which enhances the quality of the final solution (e.g., faster optimization and multi-level simulations).

**Table 16.1** An example of coherence rules to validate a RF filter functional description

Rule no.	Rule description	Consistency level
1	For a Chebyshev type I filter whose order ( $N$ ) is an even value, the output impedance ( $Z_{out}$ ) is different from the reference one ( $Z_0$ )	Warning
2	The specified filter passband attenuation ( $A_p$ ) should be equivalent to a value ( $r$ ) less or equal to the specified ripple ( $r_s$ ) $r _{A_{ps}} \leq r_s \quad (16.1)$	Error

- Enhancing technology support via technology mapping

High-level functional description uncouples the functionality from the underlying technology artifacts. Coherence verification and system-level analyses enable early specifications' validation as well as better design space exploration. This results in a good candidate solution, but mostly idealistic. To ensure the solution's feasibility in terms of not only performance but also physical implementation, technology details should be taken into account. For this reason, the designer provides the necessary technology information during the step of "Technology Mapping." For example, ideal lumped components are transformed into real passives using appropriate device characterization libraries. Thus, the parasitic effects present in a resistor (e.g., thin film resistor model [19]), a capacitor (e.g., metal-insulator-metal capacitor model [20]), and an inductor (e.g., general inductor model [21]) are included in the design and considered in performance simulation. In distributed lines, the designer should include the substrate properties and indicate the relevant transmission line models (e.g., microstrip, stripline, and coaxial cable) along with the related physical properties.

- Validating implementation requirements using performance simulation

Once the technology input is considered in the design, the system is designed at circuit and/or physical levels. Repetitive performance simulations and analyses are carried out in order to assess the design frequency response and check whether it meets the initial requirements. The accuracy of technology models and the quality of EDA tools have a direct impact on the accuracy of the obtained results. To prevent a performance less than planned, a reasonable margin may be considered to suppress the effect of eventual inaccuracies in both models and simulations. When a satisfactory solution is achieved, a ready-to-manufacture layout is produced and submitted for fabrication and testing.

### (c) Q-matrix

The design scheme of Fig. 16.2 provides mechanisms to capture functional models, validate, process, and exchange them between the different design stages. However, it still lacks an efficient tool to exchange design data. So, it is important to endorse this design scheme with such tool because current techniques suffer from various limitations. In fact, modern EDA packages use matrix representation for the storage and exchange of electrical design data (e.g., scattering parameters). First, RF components are of different natures (e.g., linear/nonlinear, passive/active). However, there is no unique matrix form to capture their electrical response in a uniform way. Then, some matrix representations fail sometimes to capture the electrical response of all RF systems. For instance, mixers cannot be characterized using scattering parameters because they required to have the same frequency at all ports. In addition, there are many data file formats. The design of a RF system requires numerous files because almost each type of simulation requires a specific data file format. So, design data become fragmented and difficult to handle throughout the design process. This is due to the absence of a centralized data

structure enabling easy data capture, storage, and exchange. In addition, most existing file formats do not capture efficiently the circuit environment configuration.

Considering all these limitations in existing tools and file formats, the five-step design scheme is augmented by tool-neutral multi-dimensional data structure, namely the Q-matrix that captures the electrical behavior of each RF system regardless of its nature, number or types of ports, etc. The Q-matrix is placed in the heart of the design framework in order to link all the design stages and steps.

The Q-matrix is mathematically constructed as a superset encompassing and extending the traditional scattering parameters (see definition in [1]). As stated in Eq. (16.2), the Q-matrix is composed by  $N \times N$  elements ( $q_{ij}$ ) for a  $N$ -port network. When the frequency at all ports is the same and not equal to zero, the Q parameters are the same as the scattering parameters.

$$q_{ij} = \frac{b_j|_{f=f_i}}{a_i|_{f=f_i}} \quad (16.2)$$

where

- $b_j$  The reflected wave at the  $j$ th port
- $a_i$  The incident wave at the  $i$ th port
- $f_i$  Frequency of the signal entering the  $i$ th port
- $f_j$  Frequency of the signal leaving the  $j$ th port

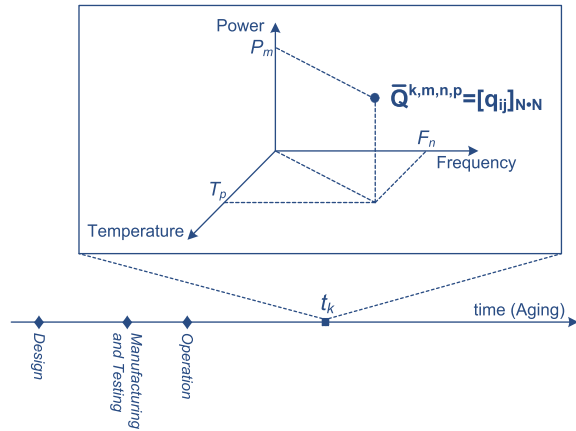
The definition given in Eq. (16.2) generalizes the traditional incident and reflected wave ratios regardless of the operating frequency at each port. However, this is not enough to cover more than one independent variable. For this reason, a broader definition, that is  $\bar{Q}^{t,T,P,F}$ , is introduced by the Eq. (16.3). It defines the Q-matrix  $N^2$  basic  $q_{ij}$  elements in function of four independent variables (i.e., frequency, power, temperature, and aging time). Figure 16.3 attempts to graphically depict the relationship between the  $q_{ij}$  elements and the extended data structure  $\bar{Q}^{t,T,P,F}$ .

$$\bar{Q}^{t,T,P,F} = \left[ [q_{ij}]_{N \times N} \right]_{N_t \times N_T \times N_P \times N_F} \quad (16.3)$$

where

- $t$  Time (aging)
- $T$  Temperature
- $P$  Power
- $F$  Frequency
- $N$  Total number of ports
- $N_t$  Number of time steps
- $N_T$  Number of temperature points
- $N_P$  Number of power points
- $N_F$  Number of frequency points

**Fig. 16.3** Generalized Q-matrix is a function of frequency, power, temperature, and time



A XML-based data file format was developed to hold Q-matrix electrical design data and a selection of environment setup information. The file is conceived for the storage of data from different sources (e.g., design cycle, test and measurements, and operation phase). It is a multi-page file that is composed of similar data blocks. During design cycle, data from different design stages, iterations, or designers are gathered into the same data file and shared among the design team.

### 16.3.2 Hardware Abstraction Strategy for RF Design

To improve design automation, the design framework presented in the previous section introduced mechanisms for high-level functional description and model-based analyses and synthesis. These concepts provide a certain level of abstraction. However, from practical standpoint, some issues remain unresolved. How high-level models for RF systems may be developed and used? And how to make these models technology independent?

To answer these questions, a generic model for RF systems should be defined. In addition, clear abstraction levels that differentiate technology-independent functional models from technology-dependent implementations are also needed. To do so, this section discusses a hardware abstraction strategy for RF design that attempts to endorse the design framework with practical answers.

#### (a) Definition and advantages

Abstraction is defined by Dictionary.com as “*the act of considering something as a general quality or characteristic, apart from concrete realities, specific objects, or actual instances.*” The psychologists Goldstone and Barsalou assert in [22] that “*to abstract is to distill the essence from its superficial trappings.*” The sixth edition of Oxford Dictionary of Computing defines abstraction as “*the principle of ignoring those aspects of a subject that are not relevant to the current purpose in order to*

*concentrate more fully on those that are*” [23]. So, all these definitions suggest that abstraction is the act of uncoupling the fundamental characteristics of an object from the details of its construction. Thus, its main purpose is presenting a simplified view of a complex reality by hiding its unnecessary attributes and aspects. This is why abstractions can be of various types due to the complexity of the real world.

The major benefits of abstraction can be summarized as follows [24]:

- *Simplicity*: Abstraction reduces complexity by producing an object view as simple as possible.
- *Relevance and information control*: Abstraction is meant to capture only the relevant aspects of an object. This includes the control of the information amount needed to describe that object at a given level of abstraction.
- *Granularity*: An object can be described at different levels of detail. Consequently, capturing fewer details results in more abstract representations and vice versa.
- *Uncoupling abstract from concrete views*: Object abstraction provides a distance with its concrete status. The most the concrete status is hidden, the highest the abstraction level is.
- *Naming*: At a given abstraction level, the object name becomes the synonym of the object properties and attributes accessible at that level of abstraction. This strong semantics allow easier understandability of a complex system.
- *Reformulation*: An abstraction is not unique, which makes it easy to reformulate the same object properties in different formal ways that can be suitable for different scenarios.

In summary, the major advantage of abstraction remains its ability to manage complexity by hiding unnecessary details and aspects. For this reason, some computer architects consider that “*abstraction is probably the most powerful tool available to managing complexity*” [25].

Besides, most engineering literature uses frequently the term “*Hardware Abstraction Layer*” (HAL) rather than “*Abstraction*”, and “*Hardware Abstraction*” (e.g., sensor networks [26] and system-on-chip [27]) to refer to the adopted abstraction strategies in certain domains. This is typically related to the layered nature of the systems using these HALs. For instance, HAL is well known and predominant in computer systems because it serves as a logical division that makes the link between software and hardware layers. As far as the RF domain is concerned, the term adopted henceforth is “*Hardware Abstraction*” and can be defined as “*the act of masking physical details of hardware, allowing the designer to focus on the effects rather than the details resulting of manipulating directly the hardware*”. The term “*Hardware*” emphasizes that the main goal of abstraction in RF domain is the control of technology-dependent aspects throughout the design process.

## (b) RF Hardware Abstraction Strategy

In this section, a basic model to functionally describe a RF system is presented. Then, a hierarchy of abstraction levels and their corresponding design perspectives

applicable to RF design are detailed. To link both, adequate mechanisms for the manipulation of models and transition between the abstraction levels are defined.

- Basic Definitions

To foster better understanding of the proposed abstraction strategy, these alphabetically ordered terms and definitions apply in the following:

- *Abstraction*: the process of representing a RF system at a given level of detail and with respect to a given design viewpoint.
- *Abstraction view*: an extent in which models result from an association of an abstraction level and an abstraction viewpoint.
- *Abstraction level*: a reasonable characterization related to the complexity and details in a representation of a RF system.
- *Abstraction viewpoint*: a representation of a RF system from a design perspective that focuses on particular concerns within that system.
- *Coherence*: the quality of a model, a specification or a functionality of being composed of mutually consistent and non-contradictory elements (and/or attributes).
- *Formal model*: a model that is semantically consistent in a sense that it complies with the semantic rules of a given modeling language.
- *Functionality*: a field of operation related to a RF system which can be modeled using a given response function (and/or physical or logical description).
- *Granularity*: a specification that characterizes the number of parts composing a RF system with the guarantee that each part among them represents a coherent functionality.
- *Independence*: the quality of a model of being usable without specifying any technology information or platform-related attributes.
- *Model*: a formal representation of the functionality, structure, behavior, and/or physics of a RF system.
- *Platform*: a set of RF technologies (or physical infrastructure) that is either required or can be used to implement a given functionality.
- *Platform-independent model (PIM)*: a RF system model that does not specify any technology information or platform-related attributes.
- *Platform-specific model*: a RF system model that includes technology information or platform-related attributes to be utilized in the implementation of that system.
- *Platform model (implementation)*: a RF system model that describes the physics and/or the detailed implementation of that system.
- *Refinement*: the process of adding more details to an existing model.
- *System*: a RF entity that is characterized by a coherent (i.e., deterministic) and identifiable functionality and a set of ports (or interfaces) enabling the interaction with its environment.
- *Technology mapping*: the process of associating a physical platform (i.e., technology data or information) to a platform-specific RF system model.



– *Transformation*: a process that translates a source model to another one given eventually a set of specifications, rules, flows, specific data, or tools.

- Modeling of RF systems

From functional perspective, a RF system is an entity whose primary purpose is fulfilling a given functionality. To do so, it interacts with the immediate environment using input and/or output ports. In practice, this functionality can be captured in various ways (e.g., mathematical formalism and data-based). The ports may be of different types and serve for distinct functions (e.g., control, biasing, and signal routing). Consequently, a RF system can be considered as a black box that is defined by a response function (i.e., functionality) and a set of input and/or output interfaces (see Fig. 16.4).

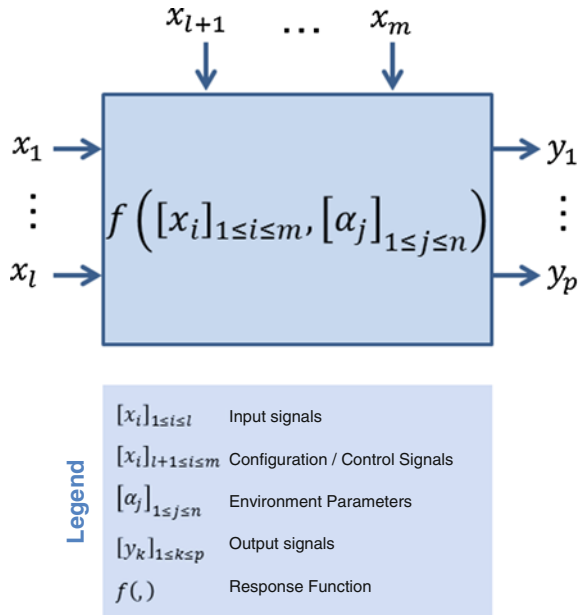
(i) System inputs and outputs

A RF system uses input and/or output ports that may be of different signal types (e.g., DC, AC, and RF). The black-box model represents these ports as interfaces to which some properties are attached (e.g., direction and signal type).

In this regard, two types of inputs are defined as follows:

1. *Regular inputs*: They represent a typical AC or RF signal and
2. *Control inputs*: They represent a signal (typically DC) that is used to drive the RF system (e.g., input voltage in a voltage-controlled oscillator and an automatic gain control).

**Fig. 16.4** A black-box view considers a RF system as an entity, which is defined by its inputs/outputs and its response function



In addition to inputs, output interfaces are characterized by regular outputs, which are similar to regular inputs.

For simplification, it may be sometimes useful, from abstraction perspective, to ignore some inputs (respectively outputs) when modeling some RF systems. In this case, the “hidden” input (respectively output) is not considered in the model. For example in some cases, biasing input voltages may be ignored in oscillators and amplifiers.

#### (ii) System functionality

A black-box model defines a response function that characterizes the RF system’s functionality. In practice, the response function may be of four types as follows:

- *Mathematical transfer function*: It uses an elaborated mathematical formalism,
- *Data-based response function*: It uses a data file resulting from either simulations or measurements,
- *Expression-based response function*: It uses mathematical equations along with datasets to characterize the system’s functionality, and
- *Hybrid response function*: It uses a mix of the previous types to model the system’s response.

In addition to inputs, the response function takes into consideration a set of environment parameters. These parameters represent variables that are not regular signal inputs to the system but characterize the environment and the context of operation (e.g., frequency, reference impedance, temperature, and time).

#### (iii) Modeling black box using SysML

As previously mentioned, SysML was retained for the development of functional description models. Can SysML be used for the capture of the black-box model?

Being a language for systems engineering, SysML allows the capture of not only behavioral and structural aspects but also any data or mathematical formalisms related to the system. As shown in Fig. 16.5, a black-box model of a RF system can be captured using a SysML block definition diagram (bdd). It can be described by a «block» construct, namely “RF system,” whose environment parameters are captured in the “*values*” section and its response function is captured in the “*constraints*” section. The latter is composed of other  $m$  «block» constructs, namely “input port” and  $p$  «block» constructs, namely “output port.” The environment parameters related to each port can be captured within its own section “*values*.”

- Abstraction Levels, Viewpoints, and Views

Real-world RF systems (e.g., receivers and transmitters) are not monolithic entities that are always described as single blocks. Designers need frequently to architect the internals of such systems at different levels of granularity and from different design perspectives (e.g., electrical, mechanical, and thermal). Besides the fact that SysML can be used to capture all these aspects, which abstraction levels, design perspectives and corresponding models may be considered to enable the definition of RF systems’ artifacts?

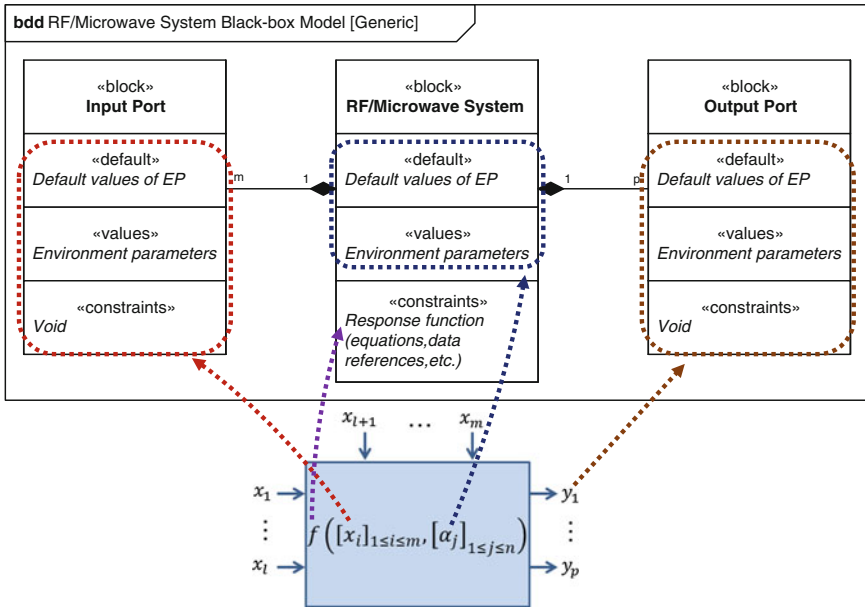


Fig. 16.5 RF system black-box model can be captured using a SysML bdd

(i) Abstraction Viewpoints

A viewpoint is a design perspective that focuses on particular concerns within the RF system. Generally, five abstraction viewpoints from which the latter can be observed and designed are as follows:

1. *Physical*: From this viewpoint, the system is described exclusively by its physical attributes. For example, a transistor is seen as a device that is defined by its dimensions (e.g., length and width), shape, the number of its metal layers as well as the type of materials and substrates used for its implementation, etc. In complex systems, other physical information may be added to this description (e.g., system’s layout, devices placement, interconnections, and signals routing).
2. *Electrical*: The system is regarded from this viewpoint as a circuit that is exclusively specified by its electrical characteristics. For instance, a resistor is defined by the electrical voltage standing between its ends, the current flowing through it, and its characteristic impedance. In complex circuits, interconnections and wiring plan (e.g., netlist) and reference nodes (e.g., ground, sources, and loads) are also specified, which allows the application of electrical laws (e.g., Kirchoff law and superposition law).
3. *Structural*: From this viewpoint, the designer’s interest is neither physical nor electrical. The focus is on how to arrange basic parts (e.g., circuits and devices)

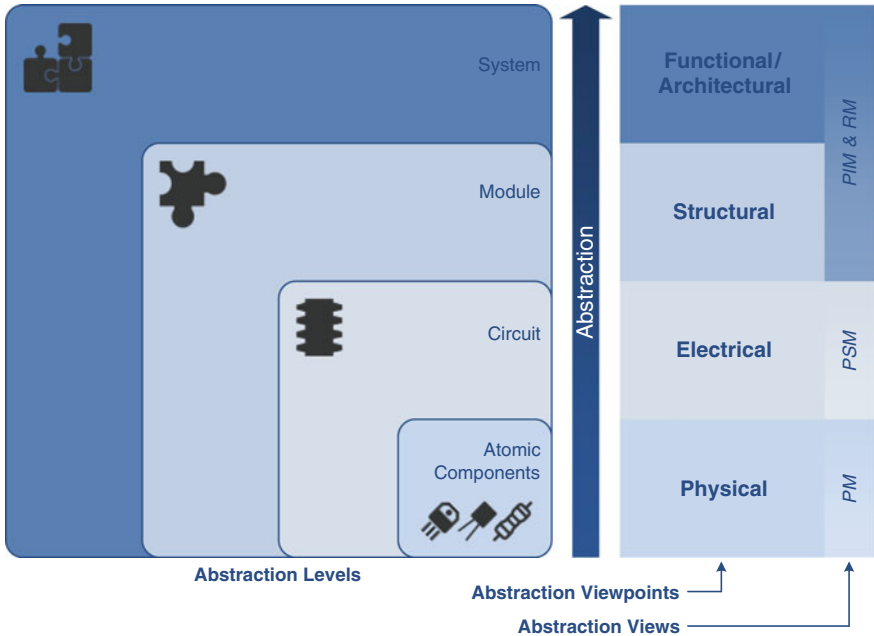
in order to build a system's topology. These parts may or may not be self-contained and interchangeable. For example, a phased-locked loop is seen from this viewpoint as a control system where phase detectors, filters, dividers, voltage-controlled oscillators, and other parts are placed in a certain arrangement in both main and loop paths. It is worth noting that the structural viewpoint is not purely mechanical, as the term "*structural*" may suggest, but denotes especially system assembly and integration aspects.

4. *Architectural*: From this viewpoint, the designer is interested in how a system is structured using self-contained and interchangeable parts. This viewpoint expresses generally a contractual architecture that is predefined by the high-level specifications. For example, a receiver may be architected in compliance with a given reference architecture (e.g., super-heterodyne). In some cases, this viewpoint is not considered because the system's complexity is reduced in a way that the structural viewpoint is enough to characterize its structure/topology.
5. *Functional*: This viewpoint is mostly interested in the system's functionality and operation. It underlines the purpose of the RF system and how it is expected to work. Thus, it defines for example its operation constraints as well as the specific role of each actor or building part within that system (if a particular reference architecture is already contracted).

## (ii) Abstraction levels

As shown in Fig. 16.6, four distinct abstraction levels are defined for RF domain as follows:

1. *Atomic Layer*: Contrary to digital design where the lowest abstraction level (i.e., physical) is represented by a "*device*" (i.e., typically a silicon-based transistor), RF systems' physical implementation cannot be represented by a single device due to the predominant mix of technologies. For this reason, the lowest abstraction level (from physical design perspective) considered in RF system is a "*layer*" of atomic components. The term "*layer*" denotes that many individual devices may represent the physical design perspective of RF systems. The term "*atomic*" indicates that each component of this layer cannot be subdivided further from any RF design perspective. In other words, such component can no longer be subdivided into elements that might be captured using a black-box model. In a hierarchical tree view representation of a RF system, atomic components lie down at the "*leaf*" level. Among the components satisfying these conditions, at least four groups are identified: transmission lines, lumped components, nonlinear devices, and sources.
2. *Circuit*: A physical assembly of atomic components can also be regarded from electrical viewpoint. Accordingly, the physical details are ignored while the electrical properties are emphasized. This viewpoint corresponds to a higher level of abstraction, namely "*circuit*." In this regard, a circuit is simply a network composed of electrical elements that are connected by a media through



**Fig. 16.6** Five abstraction viewpoints are associated to four abstraction levels and four abstraction views

which an electrical signal flows (e.g., current, wave). At the circuit level, atomic components (or any assembly of them) lose virtually their physical properties. They become represented by their electrical properties and functions as well as their respective input/output flows (e.g., current and voltage).

3. *Module*: One or many circuits can be assembled in a given topology to construct a self-contained entity. Therefore, the design perspective is no longer electrical but structural. The corresponding abstraction level is “*module*.” This entity can be defined as an individual, independent, and interchangeable unit that can be used to build more complex structures. At this level of abstraction, the internal electrical properties of a module are hidden. Instead, it is defined by its functionality and inputs/outputs. Both are defined from structural viewpoint.
4. *System*: As interchangeable units, modules can be used to construct complex structures. From this design viewpoint, the internal structure of each module is not the primary concern of the designer who focuses more on how to organize the modules to achieve a specified system-level functionality regardless of how the internals of each module were structured. Thus, the design perspective is more architectural/functional rather than structural. This corresponds to a new abstraction level, namely “*system*.” It is the highest and the least complex abstraction level.

Figure 16.6 shows how the four abstraction levels are nested and associate them to the corresponding viewpoints. Each abstraction level hides those lying underneath it. The atomic layer (i.e., composed of atomic components) is the lowest abstraction level. The circuit level lies on the top of the atomic layer while the module level lies between the latter and the system level. This makes the system is the highest abstraction level. “*Abstraction*” consists of moving up from a lower to higher abstraction level. Inversely, it is “*refinement*.”

Applied to the front end example shown in Fig. 16.7, the system-level abstraction consists of considering the direct-conversion receiver as a whole (from antenna to baseband input). At module level, individual components (e.g., filters, amplifiers, and oscillator) and their arrangement are considered. At circuit level, the internals of each module are exposed from an electrical viewpoint. Devices constituting each circuit are considered as physical layouts at atomic layer level.

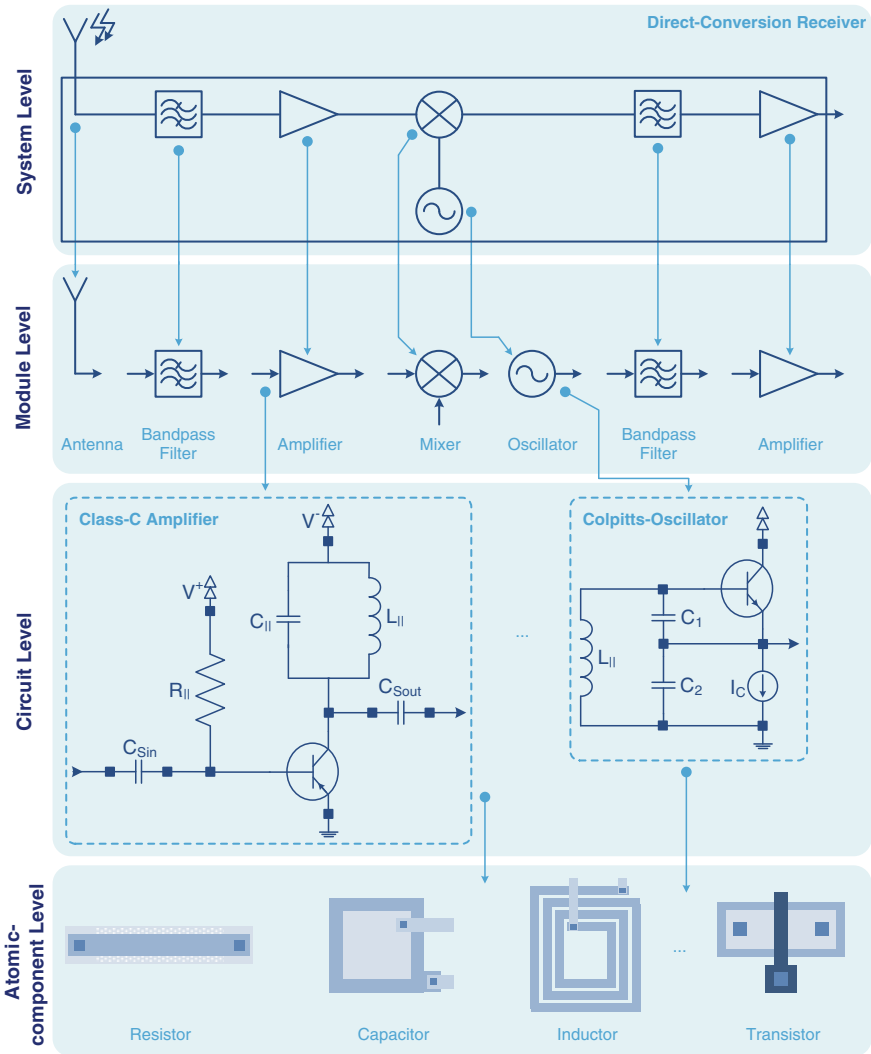
### (iii) Abstraction Views

As shown in Fig. 16.6, an abstraction view corresponds to the association of an abstraction level and an abstraction viewpoint. It is simply a construction (i.e., image) that captures from a specific viewpoint the system properties with reference to a particular level of abstraction. Four types of models are defined to express abstraction views as follows:

1. *Platform Model (PM)*: It is an atomic-layer-level representation which is developed from physical viewpoint. The PM expresses physical specificities of system implementation which may include (but not limited to) layout, ports, interconnections, substrates, fabrication materials, etc. For example, physical properties of devices (e.g., dimensions, shape, and layers) are captured using a PM.
2. *Platform-specific Model (PSM)*: It describes the system from an electrical viewpoint. A PSM expresses platform-specific artifacts using electrical abstractions. For example, lumped components may be represented using standard schematics (e.g., amplifier circuit in Fig. 16.7).
3. *PIM*: It describes the aspects pertaining to how the system should be structured and architected. These aspects include structural and/or architectural guidelines, design and operation constraints, etc. This model remains independent from the implementation technology because it is not intended to embed any physical or electrical information that is specific to a particular platform.
4. *Requirement Model (RM)*: It captures the system specifications from functional viewpoint. It expresses the requirements that are related to functional design concerns. This model is not only used to express system-level specifications. It may also serve to map the system requirements to the other models for traceability, validation, and verification purposes.

- Transition between Abstraction Levels

In the previous section, we defined four types of views to model RF systems from different design perspectives and at different levels of abstraction. The next step is



**Fig. 16.7** An example of how the different abstraction levels are considered in a typical direct-conversion receiver

the definition of adequate mechanisms for the transition between the different abstraction levels and viewpoints. It is important that these mechanisms of the model-centric abstraction views are to improve automation.

For this purpose, a model-to-model transformation approach is adopted. Accordingly, two types of transformations are considered. The first, namely *cross-view transformation*, derives a view model into another distinct one (e.g.,

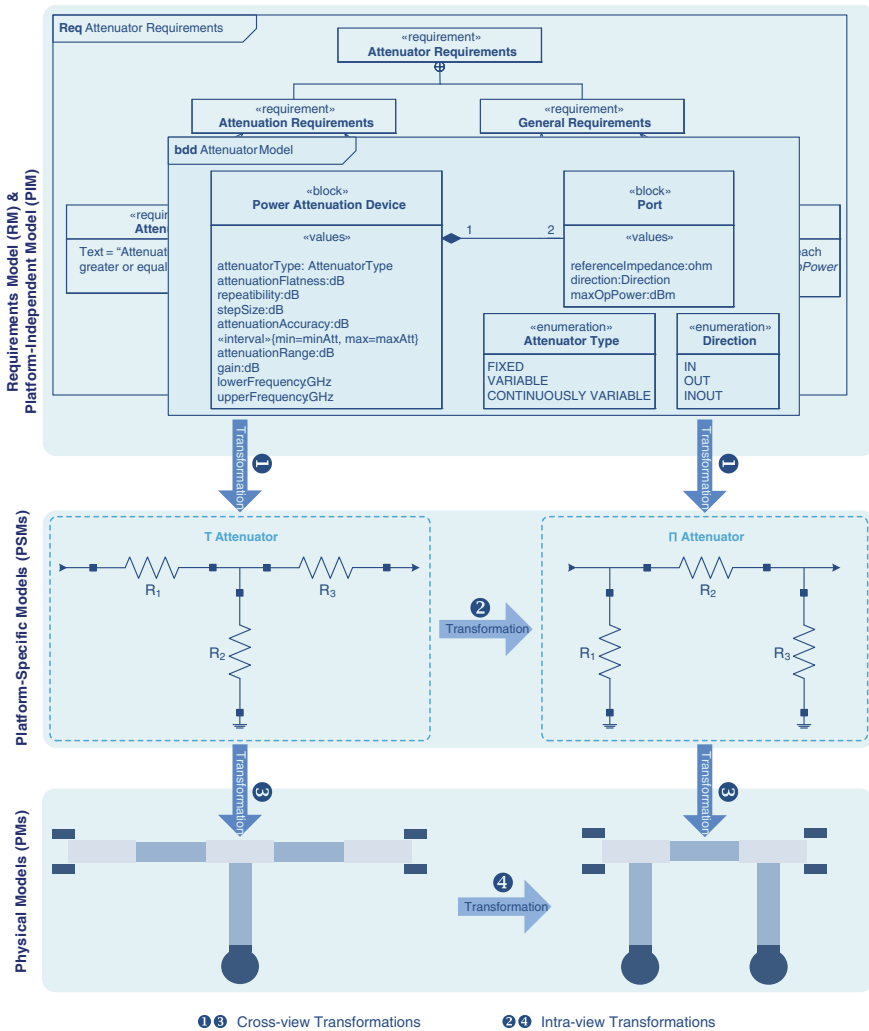


Fig. 16.8 Cross- and intra-view transformations

transformations ① and ③ in Fig. 16.8). The second, namely *intra-view transformation*, derives a view model into another one of the same type (e.g., transformations ② and ④ in Fig. 16.8).

(i) Cross-view transformations

A cross-view transformation converts a view model (i.e., RM, PIM, PSM, and PM) into another one corresponding to a different abstraction level and viewpoint.

As shown in Figs. 16.8 and 16.9, a first cross-view transformation may be used to translate a PIM (and/or RMs) into one or many PSMs. The second translates a



PSM into one or many PMs. For simplicity, there is no cross-view transformation to convert a requirement model into a PIM. That is mainly because RMs are used in RF domain as a complement to PIMs to enable specifications’ validation and enhance traceability of requirements within the design cycle.

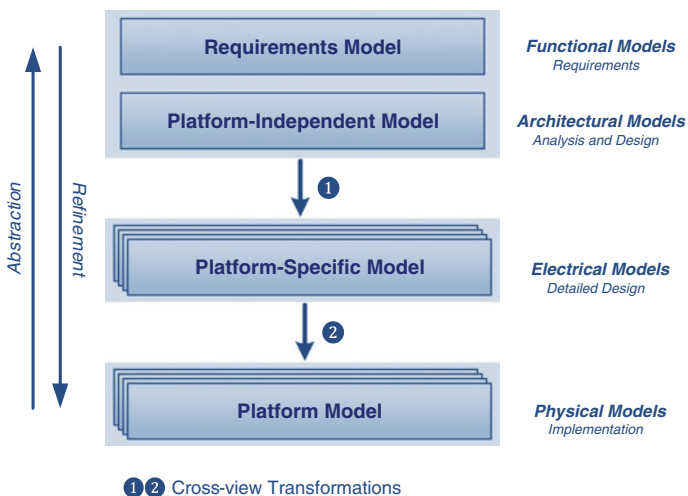
(ii) Intra-view transformations

The second type of transformations converts a model to another model of the same kind. This means that both the source and target models share the same level of abstraction and are developed from the same abstraction viewpoint. Theoretically, it is possible to use intra-view transformations to convert all types of view models. In practice, this type of transformations is more interesting to use with concrete models (i.e., PIMs, PSMs, PMs) than abstract ones (i.e., RMs). For illustration, transformation ② in Fig. 16.8 is intended to translate a T-pad attenuator PSM into a  $\pi$ -pad attenuator PSM while transformation ④ in Fig. 16.8 does the same between T- and  $\pi$ -pad attenuator PMs.

Depending on the granularity level within the considered abstraction view, intra-view transformations may be classified into two main categories: (i) view-model bridges and (ii) granularity refinement transformations. The former carries out model transformation without changing the granularity level of the original view model while the latter does not.

- Integration of RF Hardware Abstraction Strategy in the Design Framework

For the design framework to be streamlined along with the concepts adopted in the RF hardware abstraction strategy, the first task is to delimit its various stages in accordance with the four abstraction views of Fig. 16.9. To this end, the Q-matrix is



**Fig. 16.9** Two transformations are defined to move from system requirements throughout implementation

still considered in a central position accessible at various steps be they in the RM/PIM, the PSM, or PM (i.e., implementation) domains. The resulting mapping of the design scheme to the different abstraction models is captured in Fig. 16.10.

Under this scheme, the RM/PIM domain covers the functional description of the system, the coherence verification, and system-level performance simulation. In this domain, the system is presented at a level that is totally independent from any technology details or platform particularities. At this level, the abstraction is very high in a way that even an unrealistic system may be functionally described but rejected through coherence verification and/or performance simulation. Next, the PSM domain may include system simulation and covers the steps of the synthesis process, which is composed of three sub-steps, namely granularity refinement, technology mapping, and performance simulation. In this domain, the system model is enriched with technology details and the abstraction level is lowered in order to take into consideration the physical constraints and information related to the implementation platform. On the first hand, technology limitations, if any, that may prevent the realization of the stated specifications are generally discovered and feedback to the previous stages can be given so the design process may be restarted or reiterated. On the other hand, if no technology limitations are met, then the

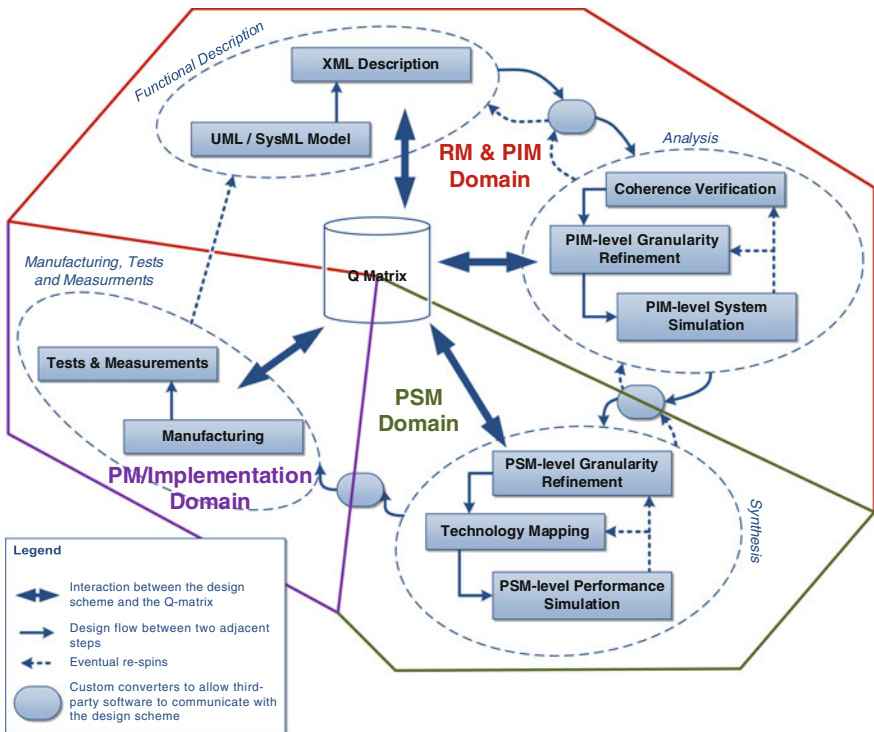


Fig. 16.10 Streamlining the design framework with the RF abstraction strategy

design will be feasible and can be moved on to the PM/implementation domain, which encompasses the manufacturing and testing steps.

It is worth noting that the border between RM/PIM and PSM domains is floating. This is due to the fact that some system-level performance simulations may result in some cases, in a circuit model that can be used also as a PSM. This said, specialized tools and/or algorithms may be used to implement cross-view transformations from RM/PIM to PSM domains as well as from the latter to PM/implementation domain. In practice, cross-view transformations do not impose any changes of the design scheme since it is always possible to move from a domain to another without facing any discontinuities in the design flow. However, intra-view transformations may need a change of granularity when applied within the same domain. Additionally, changing the granularity level during the steps of “Functional Description” and “Synthesis” is possible due to modeling flexibility in the former and the “Granularity Refinement” step in the latter. This is not the case in “Analysis.” For this reason, a new sub-step is added to this design stage in order to allow granularity refinement when required. Accordingly, some sub-steps in both “Analysis” and “Synthesis” design stages were renamed in order to emphasize the abstraction strategy being adopted.

## 16.4 Application to RF Design

The previous sections presented a design framework combined to a hardware abstraction strategy, which addresses the issues of poor automation capability and the notable technology dominance in modern RF design. Now, it is worthy to ask how the concepts and mechanisms brought by this framework (e.g., model-centric design, model-to-model transformations, and abstraction levels) may be applicable in real-world design. To thoroughly illustrate how this works, [28] proposed a complete case study that details a step-by-step tutorial for the design of RF band-pass filters using this framework. Lafi et al. [29] has also introduced how operational amplifiers can be implemented throughout the same design process.

To summarize how the framework works in real design cases, it is important to learn first how its concepts are implemented in practice. In fact, functional description models (i.e., RMs and PIMs) are handled by modeling tools supporting SysML. These models are exchanged using a dedicated XML-based file format. Traditional electrical circuit schematics (or equivalent netlists) are retained as platform-specific models while layout artworks are used as PMs. Transformations can be implemented using algorithms or dedicated EDA tools. Design rules serving as input for coherence verification and transformation tools are captured using the SysML parametric diagram and exchanged through XML or specific tools. Technology information is provided as classical design libraries or in other custom formats (e.g., XML). The Q-matrix is also represented by a dedicated XML-based file format.

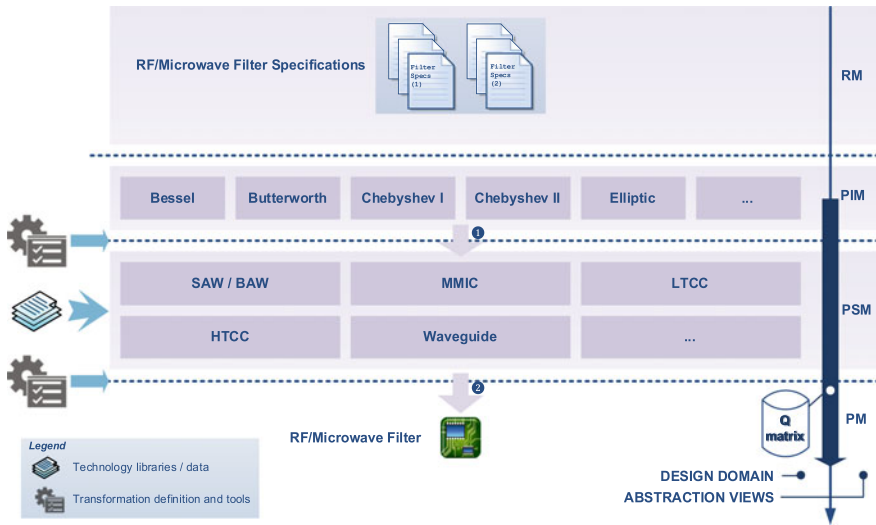


Fig. 16.11 The four abstraction views mapped to the design process of passive RF filters

If the framework is applied to RF filters’ design, the different abstraction models (i.e., views) can be mapped to the design domain major stages as shown in Fig. 16.11. Filter specifications traditionally text-based are captured using SysML requirement models. The corresponding PIM may be inspired from either a traditional filter prototype (e.g., Chebyshev, maximally flat, elliptic, and Bessel) or a custom filter model. Given the transformations’ definition (i.e., rules) and tools, the PIM is submitted first to coherence verification. Based on technology input including target platform (e.g., waveguides, lumped components, LTCC, and distributed lines) and technology libraries/data, the PIM to PSM transformation (no. 1) derives an electrical circuit schematic for the desired filter, that is, the platform-specific model. It is possible that the same transformation derives multiple PSMs for one or many platforms from a single PIM. At this step, repetitive optimizations and changes (using granularity refinement and performance simulation tools) may take place to enhance the quality of the design solution.

When a satisfactory filter design is achieved, the PSM to PM transformation (no. 2) is used in order to generate the filter’s final layout. This may take place using an EDA tool for layout generation and edition. The layout artwork can be subject to further optimizations and design rules checks to prevent performance degradation. Finally, the final layout is submitted for manufacturing and testing.

As shown in Fig. 16.11, the Q-matrix life cycle starts when the PIM is created. All the filter’s electrical data obtained in all the design steps (including tests and measurements) are stored in it. Among the advantages of the Q-matrix is the possibility to use it even after the design process is completed. Electrical data resulting from tests and measurements can be delivered with the operational filter. During the operation phase, the component performance can be monitored.

Its electrical data can always be compared to those obtained during the design and testing phases. This allows to enhance technology models, for example, thanks to better estimation of performance drifts through the filter's aging.

## 16.5 Concluding Remarks

The emerging requirements in wireless and mobile devices, especially in terms of performance, energy consumption, and form factor, put a significant amount of pressure on RF front-end designers. To meet these requirements, the response is not only the development of new RF technologies but also the renovation and sophistication of design approaches and EDA tools. This is because the modern RF design practice is suffering from numerous limitations and shortcomings, which significantly hinder productivity and design quality. The framework presented in this chapter attempts to tackle these weaknesses using a model-centric and abstraction-based design philosophy.

On the one hand, models are used not only to capture system's specifications but also to define basic elements that can be used to automate as much as possible the design process. RF systems' modeling is no longer based on traditional techniques such as predefined inaccurate and monolithic blocks. Henceforth, it is based on the use of standardized modeling languages (such as SysML), which allow more flexibility and customization. Moreover, the framework remains open to the use of domain-specific languages as well. To prevent errors from the beginning, models are submitted to an early phase of specifications' validation (i.e., coherence verification). Validated models are used at both system and circuit levels in order to select, optimize, and implement the best solution possible.

On the other hand, the framework adopts a hardware abstraction strategy in the purpose of addressing the issues of technology prevalence and limited automation in modern RF design practice. The key principle is rising the abstraction level to uncouple system functionality from technology considerations. Consequently, the basic modeling entity for RF systems is the black-box model. Four abstraction levels (i.e., atomic layer, circuit, module, and system) are established to conduct design from five distinct viewpoints (i.e., physical, electrical, structural, architectural, and functional). This gives birth to four types of abstraction models (i.e., views), which can be used to represent a RF system at a given level of abstraction and from a specific design viewpoint. Particularly, requirements and PIMs express the system's specifications independently from any technology details. A PSM depicts a system representation that is specific to a given platform. A platform model expresses that system implementation at physical level. In addition to these models, two types of model-to-model transformations are defined. The first, namely cross-view transformations, allows the generation of a target model that is different from the source one. This mechanism is used to push forward the design process from one stage to another. The second, that is intra-view transformations, allows the generation of a target model that is similar to the source one. It is used to endorse

design space exploration and lead quickly to an optimized design solution. Besides models and depending on the design stage, these transformations require other types of inputs. For instance, coherence rules are required in “Analysis” while technology information is required during the “Synthesis” stage.

In addition to all these concepts and mechanisms, the framework defines a new multi-dimensional structure, namely the Q-matrix. It is located in the heart of the design cycle and holds design data not only during the design process but also during testing and operation phases. This structure centralizes design data and provides a unique repository that is shared by all the designers throughout the design process.

Several case studies concluded the usefulness of this framework. Given the appropriate tools, it is expected that this emerging design philosophy changes not only the way RF front ends are designed but also their quality.

## References

1. Pozar, D.M.: *Microwave Engineering*, 4th edn. Wiley, New York (2012)
2. Kevenaar, T.A.M., Ter Maten, E.J.W.: RF IC simulation: state-of-the-art and future trends. In: *International Conference on Simulation of Semiconductor Processes and Devices*, pp. 7–10 (1999)
3. Razavi, B.: *RF Microelectronics*. 2nd edn. Prentice Hall, New Jersey (1997)
4. Thompson, M.T.: *Intuitive Analog Circuit Design*. Newnes, London (2010)
5. Kundert, K.: Principles of Top-Down Mixed-Signal Design, pp. 1–31. *Designer’s Guide Consulting Inc.*, Los Altos (2006)
6. Frevert, R., et al.: *Modeling and Simulation for RF System Design*. Springer, Berlin (2005)
7. Kundert, K., Chang, H.: Top-Down Design and Verification of Mixed-Signal Integrated Circuits, pp. 1–8. *Designer’s Guide Consulting Inc.*, London (2005)
8. Gielen, G., Rutenbar, R.: Computer-aided design of analog and mixed-signal integrated circuits. *IEEE Proc* **88**, 1825–1852 (2000)
9. McMahon, J.: *Design Tools, Flows and Methodologies for RF and Mixed-Signal ICs*. SOI Industry Consortium Design Clinic (2009)
10. Park, J., Hartung, J., Dudek, H.: Complete front-to-back RF SiP design implementation flow. In: *Proceedings of 57th Electronic Components and Technology Conference*, pp. 986–991 (2007)
11. Dunham, W., et al.: RF Module Design: Requirements and Issues, pp. 30–37. *RF Design Magazine*, June 2003
12. Lafi, S., Champagne, R., Kouki, A.B., Belzile, J.: Modeling radio-frequency front-ends using SysML: a case study of a UMTS transceiver. *First international workshop on model based architecting and construction of embedded systems* (2008)
13. Object Management Group: *Systems Modeling Language Specifications—Version 1.3*. <http://www.omg.org/spec/SysML/1.3/>. Retrieved on 24 Mar 2014
14. Holt, J., Perry, S.: *SysML for Systems Engineering*. The Institution of Engineering and Technology, Stevenage (2008)
15. Delligatti, L.: *SysML Distilled: A Brief Guide to the Systems Modeling Language*. Addison-Wesley, Reading, MA (2014)
16. Weilkiens, T.: *Systems Engineering with SysML/UML: Modeling, Analysis, Design*. Morgan Kaufmann, Los Altos (2008)

17. Friedenthal, S., Moore, A., Steiner, R.: *A Practical Guide to SysML: The Systems Modeling Language*. Morgan Kaufmann, Los Altos (2008)
18. The World Wide Web Consortium: Extensible Markup Language (XML). <http://www.w3.org/XML/>. Retrieved on 25 Mar 2014
19. Vishay Intertechnology: Frequency Response of Thin Film Chip Resistors. Technical Note (2009)
20. Gruner, D., Shang, Z., Subramanian, V., Korndorfer, F.: Lumped element MIM capacitor model for Si-RFICs. In: *IEEE MTT-S International Microwave and Optoelectronics Conference*, pp. 149–152 (2007)
21. Green, L.: RF-inductor Modeling for the 21st Century. *EDN Magazine*, pp. 67–74 (2001)
22. Goldstone, R.L., Barsalou, L.W.: Reuniting perception and conception. *Cognition* **65**(2), 231–262 (1998)
23. Rozenberg, G., Vaandrager, F.: *Lectures on Embedded Systems*. Springer, Berlin (1998)
24. Saïta, L., Zucker, J.-D.: *Abstraction in Artificial Intelligence and Complex Systems*. Springer, Berlin (2013)
25. Archer, N., Head, M.M., Yuan, Y.: Patterns in Information search for decision making: the effects of information abstraction. *Int. J. Human Comput. Stud.* **45**, 599–616 (1996)
26. Handziski, V., et al.: Flexible hardware abstraction for wireless sensor networks. In: *Proceedings of the Second European Workshop on Wireless Sensor Networks*, pp. 145–157 (2005)
27. Yoo, S., Jerraya, A.A.: Introduction to hardware abstraction layers for SoC. In: *Design, Automation and Test in Europe Conference and Exhibition* (2003)
28. Lafi, S., Elzayat, A., Kouki, A.B., Belzile, J.: A RF hardware abstraction-based methodology for front-end design in software-defined radios. In: *European Conference on Communications Technologies and Software-defined Radios* (2011)
29. Lafi, S., Kouki, A.B., Belzile, J.: A new hardware abstraction-based framework to cope with analog design challenges. In: *IEEE 23rd International Conference on Microelectronics* (2011)

# Chapter 17

## Optimization Methodology Based on IC Parameter for the Design of Radio-Frequency Circuits in CMOS Technology

Abdellah Idrissi Ouali, Ahmed El Oualkadi, Mohamed Moussaoui and Yassin Laaziz

**Abstract** This chapter presents a computational methodology for the design optimization of ultra-low-power CMOS radio-frequency front-end blocks. The methodology allows us to explore MOS transistors in all regions of inversion. The power level is set as an input parameter before we begin the computational process involving other aspects of the design performance. The approach consists of trade-offs between power consumption and other radio-frequency performance parameters. This can help designers to seek quickly and accurately the initial sizing of the radio-frequency building blocks while maintaining low levels of power consumption. A design example shows that the best trade-offs between the most important low-power radio-frequency performances occur in the moderate inversion region.

### 17.1 Introduction

The design of low-power, low-cost wireless transceivers has become more significant due to the explosion of portable and ubiquitous wireless applications such as personal area networks and wireless sensor networks. These applications need to limit power consumption at microwatt level [1]. The IEEE 802.15.4 standard is introduced to satisfy this specification of low-power, low-cost, and short-range wireless communications but has relative flexibility in terms of noise, linearity, and bandwidth requirements [2].

The design of high-performance radio-frequency (RF) analog CMOS integrated circuit is still a complicated activity. Many reasons contribute to this complicity:

---

A.I. Ouali · A.E. Oualkadi (✉) · M. Moussaoui · Y. Laaziz  
LabTIC, National School of Applied Sciences of Tangier, Abdelmalek Essaadi University,  
ENSA Tanger, Route Ziaten, Tanger principale, BP 1818 Tangier, Morocco  
e-mail: eloualkadi@gmail.com



- The design specifications are varied and numerous. In addition to the power consumption and bandwidth, there are some concepts which come from the circuit analysis such as gain, image rejection, signal distortion, signal-to-noise ratio (SNR), phase margin, and input/output impedance.
- The performances of the analog circuit depend on the physical phenomena of the transistor and the passive components (resistors, capacitances and inductors) particularly in the RF domain, such as the channel length modulation, nonlinearity, and noise. The issue is the ability of transistor models to accurately express actual electrical characteristics.

These problems justify the difficulty for developing an efficient design methodology. The most used methodology is based on an intuitive approach for the behavior of the circuit and a simple model of the transistor biased in the weak or strong inversion region, while the moderate inversion is taking a high importance particularly for low-power wireless applications. On the other hand, other methodologies do not care about the physical behavior and are based on powerful computing software to reach the design specifications after multiple attempts and simulations with different transistor parameters.

Many questions remain open for these methodologies, namely uniqueness, the quality of the found solution, the choice of the defaults values, the computing time, and the accuracy of the used MOS transistor models. In these conditions, the experience of the designer is the key to success for such methodologies.

This chapter is organized as follows: Sect. 17.2 will present some related works concerning low-power design techniques. The approach and the design description of the proposed computational design methodology will be presented in Sects. 17.3 and 17.4. To validate our approach, Sect. 17.5 will describe the design steps to follow using the proposed methodology for the design of a low-noise amplifier (LNA) at 2.4-GHz operating frequency with two different CMOS technologies (0.13 and 0.18  $\mu\text{m}$ ). Finally, Sect. 17.6 will present the conclusion.

## 17.2 CMOS Low-Power Design Methodologies and Techniques Review

Thanks to the development of CMOS technology, it is possible to implement gigahertz RF and microwave circuits with submicron technologies. The CMOS technology has a merit to be combined with digital circuits. The gate length, which is directly related to the effective channel length, is a main feature controlling the MOSFET performance. As CMOS technology is scaled into the nanometer range, the transit frequency ( $f_T$ ) and the maximum frequency of oscillation ( $f_{\text{max}}$ ) of transistors have been increased. Currently, from 50- to 100-nm gate-length MOSFETs, the transit frequency can reach 200 GHz, which allows the design of integrated circuits to operate at up to 20 GHz [3, 4].

In the literature, many researches try to reduce power consumption for different RF building blocks of the transceiver. In this context, different methods and techniques have been proposed [5–9]. In the most case, transistors for high-frequency applications are operating in strong inversion to take advantage of the high device transit frequency ( $f_T$ ) in this regime. Subthreshold operation is one of the low-power design approaches available. In this region, the gate source voltage is below the threshold voltage leading to a lower saturation voltages ( $\approx 100$  mV), which allows the use of low supply voltage. Another advantage of this region is the high gain obtained compared to the strong inversion. However, there are some unwanted issues which make this region not attractive:

- The tight bandwidth.
- The high noise at the output.
- The large transistor size increases the parasitic components which degrades some device performances such as linearity and noise.

In the design level, there are no structured or computational methodologies which can help designers during the design process when working in this region. The most published works show tentative to optimize power consumption using a large transistor width and low bias voltage. However, the rest of circuit performances are adjusted through intuitive experience or tuning process until attain some acceptable values.

In the past few years, the  $g_m/I_d$  method has been developed to explore the MOS transistor in all regions of operation [5]. This design method takes into account the transconductance and drain current ratio  $g_m/I_d$  and the normalized current  $I_n = I_d/(W/L)$  as the basic design parameters [6]. The value of  $g_m/I_d$  ratio is maximum in the weak inversion. The main advantage of this method is that the  $g_m/I_d$  versus  $I_n$  curve is technology independent, which reduce the number of electrical device parameters related to the used process. In spite of the relevance of this method, it is less used for the design of RF building blocks.

### 17.3 Approach of the Computational Design Methodology

The design complexity of RF circuits always requires a trade-off between the different parameters that come into play in the design in order to achieve the necessary performances. On the architectural level, the current-reused technique is utilized to overcome the limitation on the supply voltage and transistor overdrive [7]. Other techniques combine RF microelectronic mechanical systems (MEMS) technology and weak inversion standard CMOS to reduce power consumption and increase the integration level [8].

Reducing power consumption while maintaining acceptable performances remains a challenge for CMOS RF circuits. Designers use their own experience to achieve these objectives by carrying out several simulation exercises and

optimizations. Consequently, developing an efficient design methodology for CMOS RF building blocks has become necessary. The concept of inversion coefficient has been studied more in detail in the few last years [9]. It is a pertinent tool which provides a link between design intuition and simulation. The challenge of the designer consists in evaluating the available trade-offs in order to found the optimal circuit. The inversion coefficient method provides several simulations data which present all the device performances in all inversion regions. These data help designers to compare and make the right decision for a specific case.

The general principle of the proposed design methodology consists of four main steps:

1. Fixing the three freedom parameters: the inversion coefficient (IC), the drain current  $I_d$ , and channel length  $L$ , as described in Ref. [9], to find the optimum biasing point for all used MOS transistors in the circuit.
2. Plotting all transistor parameters ( $C_{gs}$ ,  $C_{gd}$ ,  $f_T$ ,  $g_m/I_d$ ,  $I_d$ ,  $g_{ds}$ ,  $V_{gs}$ ) as a function of IC to determine the performance of all used transistors at the fixed point. Where  $C_{gs}$  is the gate source capacitance,  $C_{gd}$  is the gate drain capacitance,  $f_T$  is the transistor bandwidth,  $g_m/I_d$  is the transconductance efficiency,  $g_{ds}$  represents the output conductance, and  $V_{gs}$  is the gate source voltage.
3. Extracting the analytical equations from the circuit which defines the performance key parameters (expressions of gain, matching condition, bandwidth, linearity, noise figure, etc.). In this step, the parasitic elements of passive components must be also taken into account in order to have the most accurate results.
4. In parallel with these steps, a trade-off between the power consumption and the rest of the performance key parameters can be found depending on the target application and the wireless standard requirements. This trade-off can be reached by creating a design flow relating RF block equations, target specifications, transistors parameters, and the final decision.

## 17.4 Hand Calculation and Automated Process Combination for Design Optimization

The design steps, developed before, can be summarized in three phases during the design process:

- The first phase consists of SPICE simulation, of the used PMOS or NMOS transistor, with a CAD tool performed by sweeping the bias voltage  $V_{gs}$ . The objective is to show the performances of the device as a function of IC.
- The second phase consists of an analytical study of the whole circuit. Generally, the expression of the design performances such as gain, noise, distortion, and matching conditions can be determined through a small-signal equivalent

circuit. This last takes into account the most parasitic components. This method compensates the inaccuracy which is not supported in the SPICE model used by the CAD simulation generated in the first phase. Short-channel effect, gate resistance, and other parasitic elements in MOS model or passives components (i.e., series resistance of integrated inductor) can be introduced in the extracted equations. This methodology provides the possibility to determine the design performances by hand calculation, and some developed automated program or combination between the two ways. The calculated results will be compared to the fixed specifications and objectives. This choice between hand calculation or automated tool depends on the nature of the studied RF block. For a simple RF block where only one node and a few number of passive components are integrated, hand calculation can be an optimal choice. For other RF blocks, which use a complicated architecture with many nodes and passives component, a generic program is helpful to increase productivity and saving time. In some cases, the combination of two methods can be a solution, and designer can calculate one performance (i.e., gain) with hand calculation and other figures of merit with automated process (i.e., distortion).

- The third phase consists of the creation of a design flow for the studied RF block to determine sequentially the figure of merit priority. It is the first step of the trade-off.

Depending on the role of the RF block, the priority can be determined. For a LNA, the gain and the noise figure are the most important figures of merit because they affect the entire receiver. According to our approach, the power consumption is the high priority. The drain current  $I_d$  appears as a predefined parameter in the design flow. The simulation of the RF block can be started with the defaults calculated sizing values. In the most cases, a small tuning adjustment must be done to eliminate some deviations. Figure 17.1 illustrates the approach of the proposed design methodology.

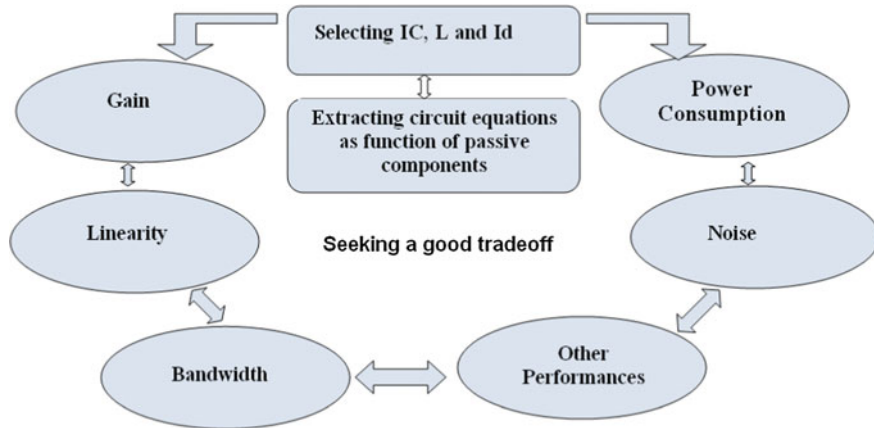


Fig. 17.1 Approach of the proposed design methodology

As previously reported, this method defines three degrees of design freedom: the inversion coefficient, the channel length, and the drain current. By selecting these three parameters, channel width  $W$  is easily found. In addition to channel width, the passive components and the architecture of the RF circuit affect also the key performances such as gain, bandwidth, linearity, and noise. Combining the inversion coefficient and the extracted circuit equations, an optimum trade-off can be found especially for ultra-low-power design. To show the effectiveness of the proposed methodology, the design of an important RF building block is demonstrated in the next section.

## 17.5 A Design Example: Ultra-Low-Power LNA

As described earlier, the first step in this methodology is the simulation of the MOS transistor parameters as a function of IC. For this study, two CMOS processes based on the BSIM3 model (TSMC RF 0.18  $\mu\text{m}$  and TSMC RF 0.13  $\mu\text{m}$ ) are used in all regions of transistor operation: weak inversion for  $IC < 0.1$ , moderate inversion for  $1 < IC < 10$ , and strong inversion for  $IC > 10$ . The objective of this comparison is to find the adequate regions and subregions of inversion, which satisfies the fixed low-power RF design specifications.

The inversion coefficient is a normalized measure of MOS inversion independent of technology parameters [9]. Equation (17.1) gives the expression of the IC:

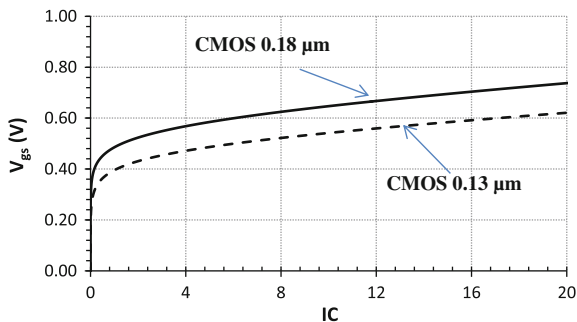
$$IC = \frac{I_d}{2n_0\mu_0C_{ox}U_T^2\left(\frac{W}{L}\right)} = \frac{I_d}{I_0\left(\frac{W}{L}\right)} \quad (17.1)$$

where  $\mu_0$  is the low-field mobility,  $n_0$  is the substrate factor,  $C_{ox}$  is the gate oxide capacitance,  $U_T$  is the thermal voltage, and  $I_0$  is a process-dependent current equal to  $2n_0\mu_0C_{ox}U_T^2$ . From Eq. (17.1), it is shown that once  $I_0$  is known, the IC can be easily extracted from the bias current and  $W/L$  ratio.

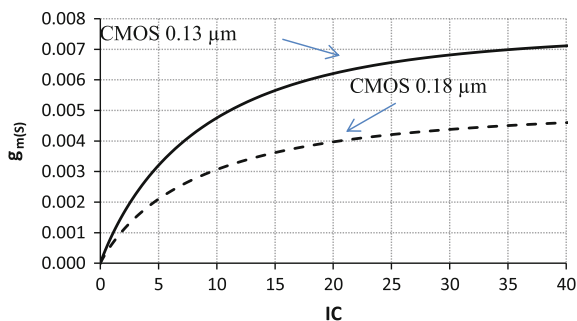
### 17.5.1 MOS Performances Versus IC

The evolution of the gate source voltage  $V_{gs}$  versus IC is very interesting. Indeed, this voltage must be maintained at sufficiently low values to ensure bias compliance, especially in low-voltage designs.

Figure 17.2 shows the variation of the gate source voltage  $V_{gs}$  versus IC for both used CMOS 0.13 and 0.18- $\mu\text{m}$  technologies. This curve is very important since it defines the voltage across each region. In the weak inversion,  $V_{gs}$  is very low (65 mV below the threshold voltage  $V_T$ ). In the moderate region ( $0.1 < IC < 10$ ), this value varies from 60 mV below  $V_T$  and 200 mV above  $V_T$ . In weak and



**Fig. 17.2**  $V_{gs}$  versus  $IC$  for CMOS 0.13  $\mu\text{m}$  ( $L = 0.13 \mu\text{m}$ ,  $W = 10 \mu\text{m}$ ) and CMOS 0.18  $\mu\text{m}$  ( $L = 0.18 \mu\text{m}$ ,  $W = 10 \mu\text{m}$ ) technologies



**Fig. 17.3** Transconductance versus  $IC$  for CMOS 0.13  $\mu\text{m}$  ( $L = 0.13 \mu\text{m}$ ,  $W = 10 \mu\text{m}$ ) and CMOS 0.18  $\mu\text{m}$  ( $L = 0.18 \mu\text{m}$ ,  $W = 10 \mu\text{m}$ ) technologies (for these curves  $V_{ds} = V_{gs}$ )

moderate inversions, the drain–source saturation voltage  $V_{DSat}$  is very low, which allows the use of low-voltage design. In strong inversion,  $V_{gs}$  is 210 mV above  $V_T$  and  $V_{ds}$  is high.

The transconductance represents the variation in the drain current  $I_d$  divided by the small variation in the gate–source voltage  $V_{gs}$  with a constant drain–source voltage  $V_{ds}$  ( $\partial V_{gs} / \partial V_{ds}$ ). Figure 17.3 shows the simulation of the transconductance  $g_m$  versus  $IC$  for the two used technologies. This parameter is very important to determine the intrinsic voltage gain, the bandwidth, and the transconductance efficiency  $g_m / I_d$ .

Another important MOS performance parameter is the transconductance efficiency. It is the quality factor describing the production of desired transconductance for a given level of drain bias current. It is almost process-independent except for the substrate factor  $n$ .

The bandwidth is the frequency where gate-to-drain current  $I_d$  gain is unity. Figure 17.5 shows  $f_T$  as function of  $IC$ . Equation (17.2) gives the expression of  $f_T$ :

$$f_T = \frac{g_m}{2\pi(C_{gs} + C_{gd})} \tag{17.2}$$

where  $C_{gs}$  is the gate source capacitance and  $C_{gd}$  is the gate drain capacitance. The  $C_{gs}$  can also be evaluated in terms of IC. The expression of the gate–source capacitance in all regions of operation is given by [9]:

$$C_{gs} = \frac{2-x}{3}C_{ox} \text{ and } x = \frac{(\sqrt{IC + 0.25} + 0.5) + 1}{(\sqrt{IC + 0.25} + 0.5)^2} \tag{17.3}$$

where  $C_{ox}$  is the gate oxide capacitance.

The transconductance is maximum in weak inversion, and it decreases modestly in moderate inversion and drops in strong inversion as shown in Fig. 17.4.

The expression of  $g_m/I_d$  in all regions of operation is given by:

$$\frac{g_m}{I_d} = \frac{1}{nU_T(\sqrt{IC + 0.25} + 0.5)} \tag{17.4}$$

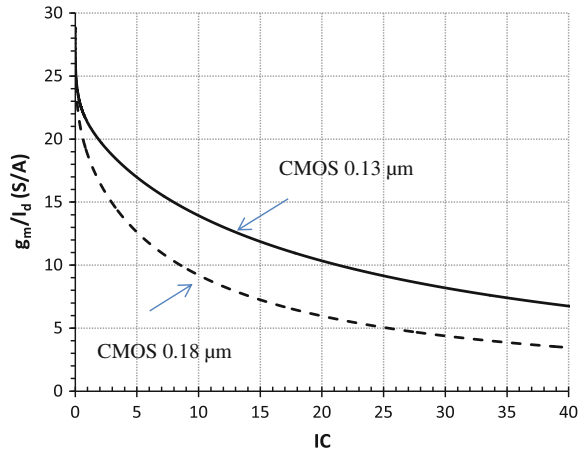
where  $U_T$  is equal to 25.9 mV.

In the weak inversion, the drain current is proportional to the exponential of the effective gate–source ( $V_{eff} = V_{gs} - V_T$ ):

$$I_d = 2n\mu C_{ox} U_T^2 \left(\frac{W}{L}\right) e^{\frac{V_{gs}-V_T}{nU_T}} \tag{17.5}$$

$$g_m = \frac{\partial I_d}{\partial V_{gs}} = \frac{I_d}{nU_T} \tag{17.6}$$

**Fig. 17.4**  $g_m/I_d$  versus IC for CMOS 0.13  $\mu\text{m}$  ( $L = 0.13 \mu\text{m}$ ,  $W = 10 \mu\text{m}$ ) and CMOS 0.18  $\mu\text{m}$  ( $L = 0.18 \mu\text{m}$ ,  $W = 10 \mu\text{m}$ ) technologies (for these curves  $V_{ds} = V_{gs}$ )



Then

$$\frac{g_m}{I_d} = \frac{1}{nU_T} \tag{17.7}$$

Using Eq. (17.5), the substrate factor  $n$  can be easily calculated and used for extracting the technology current  $I_0$ .

A maximum bandwidth is obtained when operating in the strong inversion. This bandwidth decreases in the moderate inversion and drops in the weak inversion. The curves in Fig. 17.5 shows also the increase of the bandwidth by technology downscaling from CMOS 0.18  $\mu\text{m}$  to CMOS 0.13  $\mu\text{m}$ .

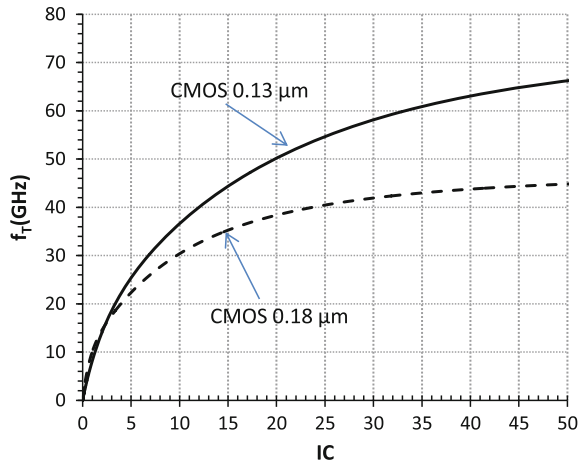
The intrinsic voltage gain is defined as the ratio of transconductance  $g_m$  and drain–source conductance  $g_{ds}$ :

$$A_v = \frac{g_m}{g_{ds}} \tag{17.8}$$

Figure 17.6 shows the simulated voltage gain versus the coefficient inversion IC. This parameter is maximum in weak inversion ( $IC < 0.1$ ) and decreases as the inversion coefficient increases.

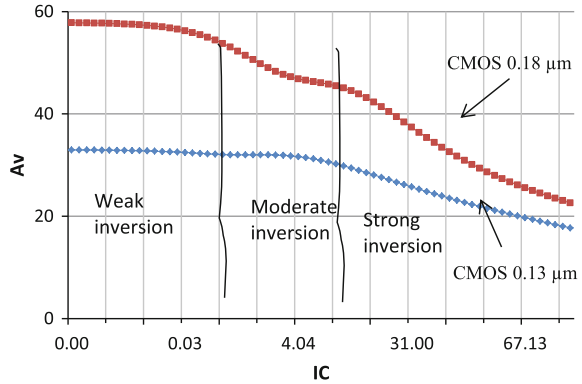
From the previous results, it is interesting to note that the weak inversion ( $IC < 0.1$ ) is characterized by a low power consumption, a good gain and a maximum transconductance efficiency, but it suffers from a low bandwidth. The region of strong inversion ( $IC > 10$ ) is characterized by high power consumption, a low  $g_m/I_d$ , a low gain, and an excellent bandwidth. However, the moderate inversion region ( $0.1 < IC < 10$ ) is characterized by a low power consumption, a good gain, a good transconductance efficiency, and a moderate bandwidth, this will allow a low-voltage design. This last region is attractive choice for the design of ultra-low-power RF circuits.

**Fig. 17.5**  $f_T$  versus IC for CMOS 0.13  $\mu\text{m}$  ( $L = 0.13 \mu\text{m}$ ,  $W = 10 \mu\text{m}$ ) and CMOS 0.18  $\mu\text{m}$  ( $L = 0.18 \mu\text{m}$ ,  $W = 10 \mu\text{m}$ ) technologies ( $V_{ds} = V_{gs}$ )





**Fig. 17.6** Intrinsic voltage gain  $A_v$  versus IC for CMOS  $0.13\ \mu\text{m}$  ( $L = 0.13\ \mu\text{m}$ ,  $W = 10\ \mu\text{m}$ ) and CMOS  $0.18\ \mu\text{m}$  ( $L = 0.18\ \mu\text{m}$ ,  $W = 10\ \mu\text{m}$ ),  $V_{ds} = 1\ \text{V}$



### 17.5.2 LNA Architecture Study

The most used techniques to optimize the power consumption for CMOS LNA design are the subthreshold and current reuse. In Refs. [10–12], authors use the subthreshold region to design a low-power LNA with inductive degeneration. Despite the good reached performances in terms of power consumption, the design process is different for all three references. In Ref. [11], the transistors are biased in the subthreshold region with a low supply voltage (0.6 V). In Ref. [10], the degeneration and load inductors are removed to reduce chip area. However, in Ref. [12], the authors use an unrestrained bias technique to improve linearity and gain. The unique used transistor is biased in the weak inversion with a very large width of  $600\ \mu\text{m}$  which increases the parasitic capacitance  $C_{gs}$ . Another disadvantage of the proposed design in [12] is the use of three big inductors which increase the chip area. The current reuse technique is used in Refs. [13, 14] to allow a low-voltage design. The  $g_m/I_d$  method is explored in Ref. [15] to design a low-power LNA at 2.4 GHz. The optimum biasing point is found by using computational routines to obtain numerically the optimum noise figure for the available range of  $g_m/I_d$  versus a wide range of  $I_d$ .

In this chapter, the proposed design methodology has been used for the design of a low-noise amplifier. This interesting RF building block is the first stage of a receiver; its main function is to provide enough gain to overcome the noise of subsequent stages (such as mixers). The LNA should provide a good linearity and should also present specific impedance, such as  $50\ \Omega$ , both to the input source and to the output load. Besides, the LNA should provide low power consumption especially when it is used for wireless and mobile communication systems. Moreover, the LNA must have a good reverse isolation to prevent self-mixing.

Figure 17.7 shows the most popular topology of the LNA with inductive degeneration. This architecture has the advantage to achieve good input matching with power gain and noise for minimum power consumption [16]. Figure 17.8 shows the small-signal equivalent circuit of the proposed topology which will be used to calculate the LNA performances.

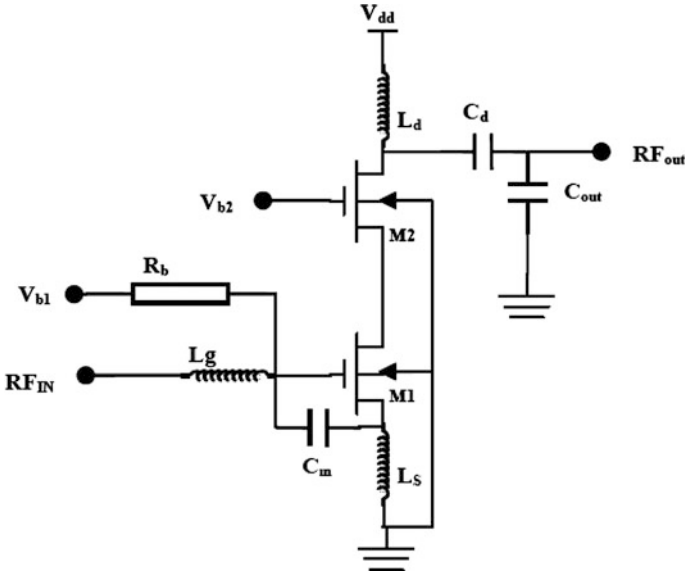


Fig. 17.7 Topology of the studied LNA

As can be seen, the selected topology uses three integrated inductors. Those inductors affect the performance of the LNA especially the noise factor and the input impedance due to losses caused by the parasitic series resistances, the substrate capacitance, and the substrate resistances. To overcome this problem, a high-quality factor integrated inductor will be used. Figure 17.9 shows the inductance value and the quality factor for the used model from the design kit TSCM RF CMOS 0.13  $\mu\text{m}$ .

Based on the small-signal equivalent circuit in Fig. 17.8, the expression of the input impedance can be determined by:

$$Z_{in} = j\omega(L_s + L_g) + R_g + R_{L_g} + \frac{1}{j\omega C_{gs}} + \frac{g_m}{C_{gs}} L_s \tag{17.9}$$

$$Z_{in}|_{\omega=\omega_0} = R_g + R_{L_g} + \frac{g_m}{C_{gs}} L_s$$

where  $R_g$  in Eq. (17.9) represents the gate resistance. It depends on the layout of the transistor M1 and the sheet resistance of the polysilicon.  $R_{L_g}$  is the series resistance of the gate inductor  $L_g$ .

Equation (17.10) shows the effective transconductance of the input stage:

$$G_{meff} = \frac{I_{in}}{V_{in}} = g_m Q_{in} = \frac{\omega T}{2\omega_0 R_s} \tag{17.10}$$

where the quality factor of the input stage  $Q_{in}$  is given by:

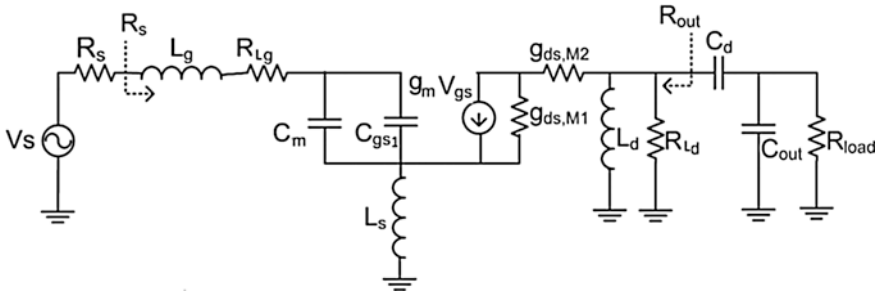


Fig. 17.8 Small-signal equivalent circuit of the proposed topology [15]

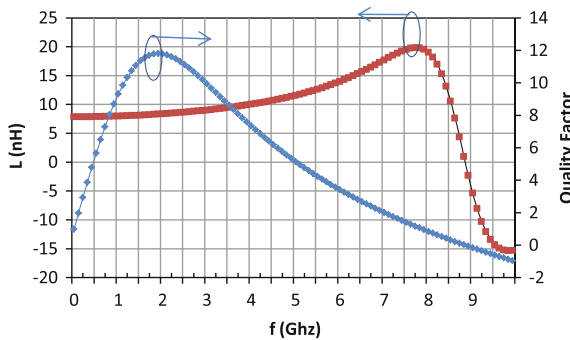


Fig. 17.9 Inductance value and quality factor versus frequency for the used gate spiral inductor (width = 3 μm, turn = 5.5, and radius = 65 μm)

$$Q_{in} = \frac{\omega_0(L_s + L_g)}{R_s} = \frac{1}{\omega_0 R_s C_{gs}} \tag{17.11}$$

where

$$\omega_0^2 = \frac{1}{(L_g + L_s) C_{gs}} \tag{17.12}$$

And  $C_{gs} = C_{gs1} + C_m$  where  $C_{gs1}$  is the gate–source capacitance of transistor M1. The output load is an LC resonator at the operating frequency  $f_0$ .

Taking into account all noise sources, the expression of noise figure is given by [17, 18]:

$$F = 1 + \frac{R_{Lg}}{R_s} + \frac{R_g}{R_s} + \frac{\gamma}{\alpha} \frac{\chi}{Q_{Lg}} \left( \frac{\omega_0}{\omega_t} \right) \tag{17.13}$$

$$\chi = \varphi + \kappa = 1 + 2|c|Q\sqrt{\frac{\delta\alpha^2}{5\gamma}} + \frac{\delta\alpha^2}{5\gamma}\left(1 + Q_{L_g}^2\right)$$

where  $\omega_f = \frac{g_m}{C_{gs1} + C_{gd}}$ ,  $Q_{L_g}$  is the quality factor of the gate inductor  $L_g$ ,  $\gamma$  is the thermal noise coefficient, and  $\delta$  is the gate noise coefficient. This expression takes into account the channel gate resistance  $R_g$  and the parasitic gate inductor resistance  $R_{L_g}$ . These two last parameters affect directly the noise and matching performance of the LNA.

However, Eq. (17.14) presents the expression of the LNA gain [10]:

$$G = \frac{V_{out}}{V_{in}} = G_{eff} \frac{R_{out}}{2} \quad (17.14)$$

The nonlinear behavior of the MOS transistor contributes to the degradation of the quality of the signal transmission in the transceiver. Nonlinearity enables the generation of new frequencies in form of harmonics which can be mixed with the fundamental frequency causing intermodulation and distortion. There are two important performance parameters to evaluate the linearity of a RF system.

The first parameter is the 1-dB compression point which is the point where the gain falls by 1 dB. The second parameter is the third-order intercept point which measures the effect of the intermodulation. The main source of nonlinearity, but not exclusive, in the studied LNA architecture is the transistor in the first stage. The power series of the transconduction of the MOS transistor is given as follows [19]:

$$i_d(V_{gs}) = I_{DC} + g_1 V_{gs} + g_2 V_{gs}^2 + g_3 V_{gs}^3 + \dots \quad (17.15)$$

where

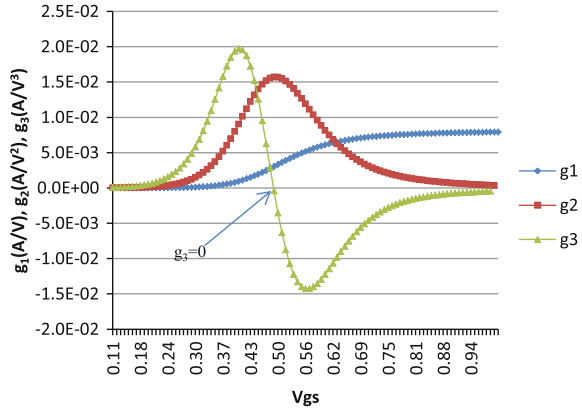
$$g_1 = \frac{\partial i_d}{\partial V_{gs}}, \quad g_2 = \frac{1}{2!} \frac{\partial^2 i_d}{\partial V_{gs}^2}, \quad \text{and} \quad g_3 = \frac{1}{3!} \frac{\partial^3 i_d}{\partial V_{gs}^3} \quad (17.16)$$

According to many investigations concerning the harmonic distortion, the coefficient  $g_3$  in Eq. (17.15) determines the value of IIP3:

$$A_{IIP3} = \sqrt{\frac{4}{3} \left| \frac{g_1}{g_3} \right|} \quad (17.17)$$

Equations (17.15) and (17.17) suppose that the MOS transistor is working in the strong inversion regime, and the variation of the drain–source voltage is neglected. In other inversion regimes, the voltage  $V_{ds}$  must be taken into account. In our proposed methodology, the parameter IIP3 must be kept near the specified value. The performance of IIP3 must be evaluated during the trade-off phase. Many techniques are

**Fig. 17.10** Power series coefficient for 0.13- $\mu\text{m}$  CMOS ( $V_{gs} = V_{ds}$ ) and ( $W = 10 \mu\text{m}$ ,  $L = 0.13 \mu\text{m}$ ) IIP3 is maximum for  $g_3 = 0$



investigated to improve the linearity of the LNA. Some techniques propose changes of the used LNA architecture which is not suitable for our case. The gate biasing technique tries to improve the linearity by controlling the  $V_{gs}$  and biasing the MOS transistor in the moderate inversion [20]. This technique is helpful while the transistor M1 will be biased in the moderate inversion near the center  $IC = 1$ . Figure 17.10 shows the variation of the three power series coefficients  $g_1$ ,  $g_2$ , and  $g_3$  versus the biasing voltage  $V_{gs}$ . The technique reported in [20] is based on the polarization of the input transistor of the LNA in the region of moderate inversion where IIP3 maximum is reached. Thus, to optimize the linearity means reducing  $g_3$  to a minimum value.

On the other hand, another parameter which affects the linearity in the inductive degeneration topology is the inductive feedback caused by the source inductor [21].

### 17.5.3 Step by Step LNA Design

To design a LNA using the proposed methodology for ultra-low-power applications, the designer should follow a procedure. Figure 17.11 shows the design flow.

All LNA performances parameters are computed with the help of small excel program using the extracted technology parameters of the BSIM3 model.

However, the descriptions of the different design steps of the LNA are explained thereafter:

**Step 1:** Choosing the optimum RF architecture for low-power design. This step is very important to avoid explicit power consumption drop. A LNA with inductive degeneration is chosen for this study.

**Step 2:** Fixing the desired performances:  $NF_{max}$  (maximum noise figure),  $G_{min}$  (minimum power gain),  $IIP3_{min}$  (minimum linearity), and  $P_{max}$  (maximum power consumption). Table 17.1 shows of the desired performances for the LNA design.

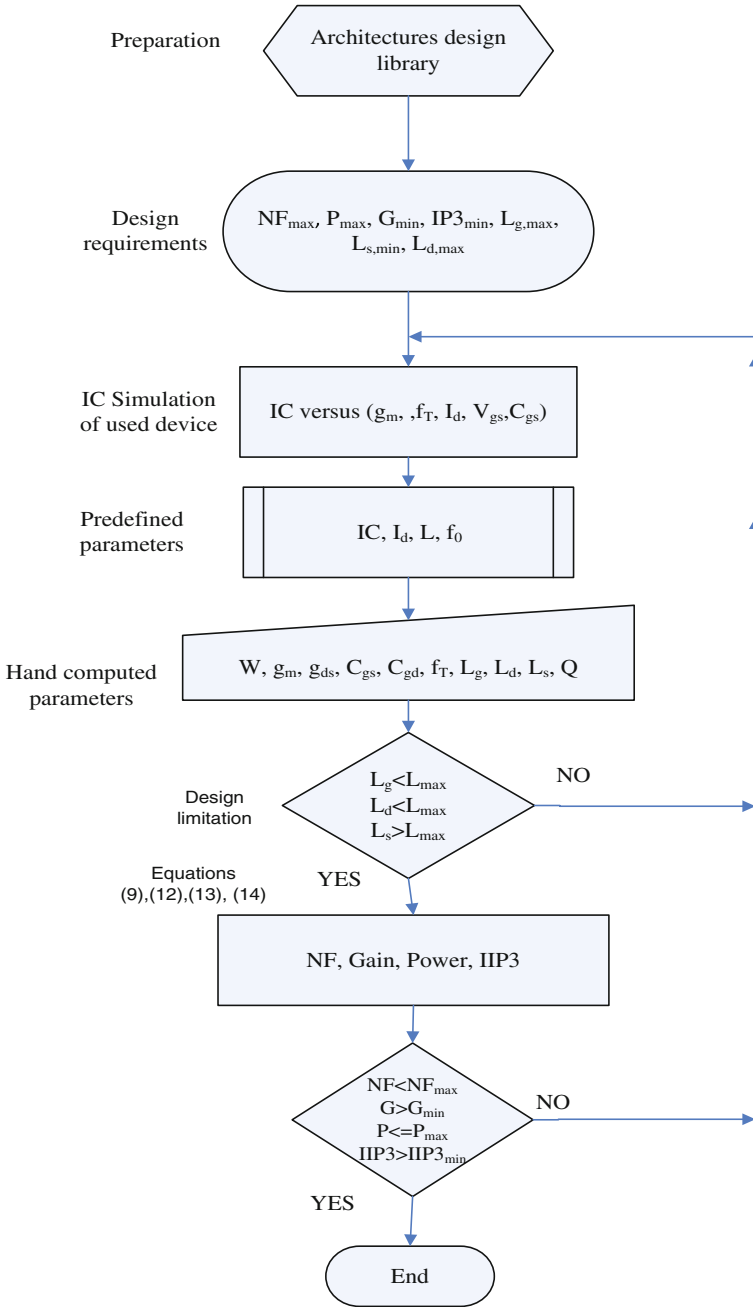


Fig. 17.11 LNA design flow

**Table 17.1** Required LNA performances

Parameters	$NF_{\max}$ (dB)	$G_{\min}$ (dB)	IIP3 <sub>min</sub> (dB)	$P_{\max}$ ( $\mu$ W)	$L_{\max}$ (nH)
Performances	3.5	10	-10	550	11

**Step 3:** Extracting the active components which affect directly the LNA power consumption. The contribution of the second stage (M2) in terms of power consumption is very low. For this reason, this transistor is biased in the strong inversion region. Only the biasing of the transistor M1 determines the dc drain current  $I_d$ .

**Step 4:** Extracting the passive components which affect directly the LNA performances. For the inductive degeneration architecture, the integrated inductors and particularly the series resistances of the inductors affect directly the noise figure and the  $50 \Omega$  matching input impedance of the LNA. The use of a high-quality factor  $Q_{L_g}$  of the inductor is required. However, the value of the inductance  $L$  should be kept smaller than the supported maximum value of the used technology (about 11 nH).

**Step 5:** Simulation of the IC and selection of the three design parameters: IC,  $I_d$ , and  $L$ . The values of the chosen parameters depend on the desired performances fixed in step 2. Two parameters are already known, the channel length  $L = L_{\min}$  to provide high  $f_T$  and  $I_d = P_{\max}/V_{dd}$ , where  $V_{dd}$  is the supply voltage. For  $V_{dd} = 1$  V, the current drain is equal to  $550 \mu$ A. From the performance of MOS transistor, the region near the center of the moderate inversion ( $IC = 1$ ) represents good trade-offs in power consumption, gain, noise figure, and bandwidth. For this reason, the IC is set equal to 1.

**Step 6:** Setting the values of the parameters fixed in step 5 in Eqs. (17.1), (17.9), (17.12), (17.13), and (17.17). The initial sizing of the LNA can be reached.

**Step 7:** Seeking for optimum trade-offs in power consumption and other LNA performances through a series of simulations of various IC near the selected value ( $IC = 1$ ). Table 17.2 shows the optimum sizing of the LNA for both used CMOS technologies  $0.13$  and  $0.18 \mu$ m.

### 17.5.4 Obtained Results

In order to show the importance of the moderate region in low-power design, the LNA is simulated in the three operation regions with the adequate passive components value for only  $550 \mu$ W of power consumption. Table 17.3 shows the performances of the LNA in different regions of operation using  $0.18\text{-}\mu$ m CMOS process. In the weak inversion region ( $IC < 0.1$ ), the power gain is dropped to 4.2 dB since the noise figure is equal to 4.2 dB. In strong inversion ( $IC > 10$ ), the gain reaches 8 dB with a noise figure of 3.5 dB since the supply is set to the

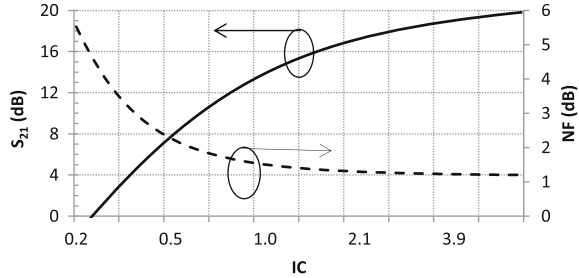
**Table 17.2** Sizing of the LNA for both 0.18- and 0.13- $\mu\text{m}$  CMOS technologies

Parameter	Value (0.18- $\mu\text{m}$ technology)	Value (0.13- $\mu\text{m}$ technology)
$W$ (M1)	170 $\mu\text{m}$	100 $\mu\text{m}$
$L$ (M1, M2)	0.18 $\mu\text{m}$	0.13 $\mu\text{m}$
$L_g$	11 nH	10 nH
$L_s$	1 nH	1.2 nH
$L_d$	7.5 nH	8 nH
$C_d$	0.52 pF	0.48 pF
$C_{\text{out}}$	1.5 pF	1.9 pF
$C_m$	0.120 pF	0.243 pF
$W$ (M2)	130 $\mu\text{m}$	90 $\mu\text{m}$
$V_{\text{dd}}$	1 V	1 V
$V_{b1}$	0.482 V	0.400 mV
$V_b$ (M2)	1 V	1 V

**Table 17.3** LNA performances in different regions of operation using 0.18- $\mu\text{m}$  CMOS process (for  $P = 550 \mu\text{W}$ )

	IC	$L$ ( $\mu\text{m}$ )	$W$ ( $\mu\text{m}$ )	$C_{\text{gs}}$ (fF)	$f_T$ (GHz)	$I_d$ ( $\mu\text{A}$ )	$S_{21}$ (dB)	NF (dB)
Weak inv.	0.1	0.18	900	355	1.5	556	4.2	4.2
Moderate inv.	1	0.18	180	106	10	582	13.5	1.5
Strong inv.	10	0.18	10	20	30	380	8	3.5

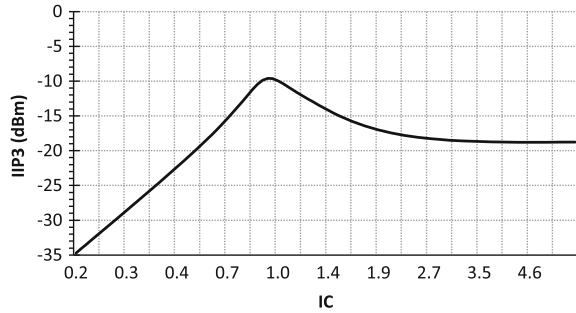
**Fig. 17.12**  $S_{21}$  parameter and noise figure versus IC for 0.18- $\mu\text{m}$  CMOS process



nominal value 1.8 V for 0.18  $\mu\text{m}$ . However, in the center of moderate inversion ( $IC = 1$ ), the power gain reaches 13.5 dB with 1.5 dB of noise figure. Figure 17.12 shows the simulation of the power gain and noise figure versus IC in moderate inversion region. In the subregion near the weak inversion, the power gain drops to 1 dB and the noise figure drops to 5 dB. The maximum power gain and minimum noise figure are reached in the subregion near the strong inversion. Figure 17.13 shows the performance of the linearity parameter IIP3 versus IC in the moderate



**Fig. 17.13** IIP3 versus IC for 0.18- $\mu\text{m}$  CMOS process



**Table 17.4** LNA performances for different IC values near IC = 1

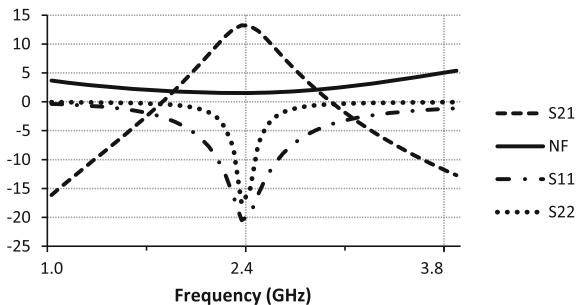
IC	$V_{gs}$ (V)	$I_d$ ( $\mu\text{A}$ )	$g_m/I_d$ ( $\text{V}^{-1}$ )	NF (dB)	Gain (dB)
0.60	0.46	360	20.16	1.89	10.7
0.73	0.47	430	19.6	1.66	12.12
0.98	0.48	525	19	1.55	13.35
1.1	0.49	629	18.5	1.46	14.42
1.3	0.5	750	17.9	1.4	15.34

inversion region. The maximum value is reached for IC = 0.9 which is compatible with the analysis done in Sect. 5.2.4.

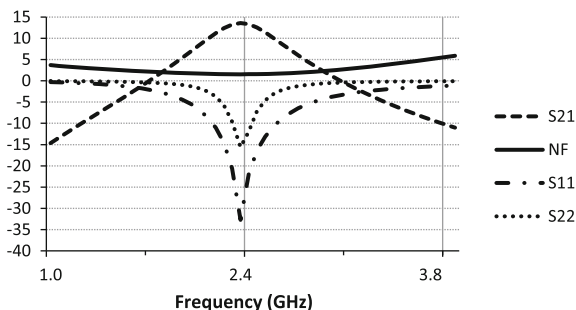
As it can be deduced from Table 17.3, the moderate inversion represents a good region for low-power design. Since the inversion coefficient IC varied from 0.1 to 10, the optimum value of IC could be determined by the simulation of IC as a function of the main required performances of the LNA. Table 17.4 shows the simulation results obtained by using 0.18- $\mu\text{m}$  CMOS process for different values of IC near the center of moderate inversion region. This table shows that the obtained gain reaches 10.7 dB with 1.89 dB of noise figure for only 360  $\mu\text{W}$  of power consumption, while the other performances such as the third-order input intercept point (IIP3) and inductance value are respected. These performances continue to improve by increasing the inversion coefficient value but for more power consumption. The power gain reaches 15.34 dB with 1.4 dB of noise figure for only 750  $\mu\text{W}$  of power consumption. Figure 17.14 shows the simulated S parameters of the LNA using 0.13- $\mu\text{m}$  CMOS process. The reflection coefficients  $S_{11}$  and  $S_{22}$  are both equal to  $-20$  dB at 2.4 GHz industrial, scientific, and medical (ISM) frequency band for 13.5 dB of power gain and 1.5 dB of noise figure.

Figure 17.15 shows the simulated S parameters using 0.18- $\mu\text{m}$  CMOS process. The power gain reaches 13.5 dB since the input matching reflection coefficient  $S_{11}$  is improved to  $-30$  dB and  $S_{22}$  is equal to  $-15$  dB. The simulated performance of the linearity defined by the 1-dB compression point ( $P_{1dB}$ ) is shown in Fig. 17.16 for 0.13- $\mu\text{m}$  CMOS technology. The LNA provides  $P_{1dB} = -18$  dBm and IIP3 =  $-9.5$  dBm for both used technologies.

**Fig. 17.14** S parameters of the LNA by using 0.13- $\mu\text{m}$  CMOS process



**Fig. 17.15** S parameters of the LNA by using 0.18- $\mu\text{m}$  CMOS process



**Fig. 17.16** 1-dB compression point by using 0.13- $\mu\text{m}$  CMOS process

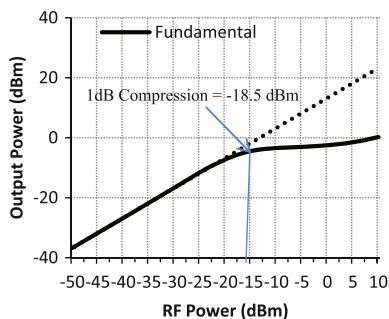


Table 17.5 shows the LNA obtained performances in 0.18- and 0.13- $\mu\text{m}$  CMOS processes compared to the state of the art. The LNA provides  $P_{1\text{dB}} = -18$  dBm and  $\text{IIP3} = -9$  dBm for both used technologies. In this work, interesting performances of noise figure and power gain are achieved for two different technologies by comparing with other researches using different low-power RF design techniques.

**Table 17.5** Comparison of LNA performances with the state of the art

Parameters	This work		[22] <sup>a</sup>	[23] <sup>a</sup>	[13] <sup>a</sup>	[15] <sup>a</sup>	[14] <sup>a</sup>
Year			2007	2009	2009	2011	2013
$f$ (GHz)	2.4	2.4	2.4	2.4	2.4	2.4	3.66
NF (dB)	1.52	1.5	2.8	1.6	2.2	3.85	2
Gain (dB)	13.5	13.5	22.7	14.4	14.4	10.7	14
Power (mW)	0.545	0.533	0.943	0.960	1.7	0.570	2.8
$S_{11}$ (dB)	-30	-20	-14	-18.1	-23	-	-10.6
$S_{22}$ (dB)	-15	-20	-	-12.7	-13.7	-	-
$V_{dd}$ (V)	1	1	1	0.9	0.4	1	0.8
IIP3 (dBm)	-9.2	-9.5	5.14	-9	-	-5	10.5
$P_{1dB}$ (dBm)	-18	-18.5	-10	-	-15.45	-	-
Low-power design technique	IC	IC	Moderate inversion	Moderate inversion	Current reuse	$g_m/I_d$	Current reuse
Technology	0.18	0.13	0.09	0.18	0.13	0.09	0.13

<sup>a</sup>Simulation results

## 17.6 Conclusion

In this chapter, a design methodology for ultra-low-power RF circuits has been described. The sizing of the circuit components is performed by the use of the inversion coefficient. The main advantage of the proposed methodology is the exploration of the MOS transistor in all regions of operation from weak to strong inversion. The studied example shows that the best trade-off between the most importance low-power RF performances occurs in the moderate region. This methodology contributes also for time reduction by seeking the initial sizing of the RF building blocks. The use of 0.13- and 0.18- $\mu\text{m}$  CMOS technologies proves that this methodology is process-independent and can be used for other CMOS standard technologies. The moderate inversion region represents an attractive choice especially the subregion around the center of the inversion level ( $IC = 1$ ) for the design of ultra-low-power RF circuits. The trade-off between power consumption and the most RF performances can be achieved in this region as it is demonstrated in this work through the example of LNA. The obtained simulation results are acceptable for low-power RF standards especially for the IEEE 802.15.4. Therefore, this methodology can be used in the design of other building blocks of a RF transceiver to ensure optimized power consumption without dropping other performances.

## References

1. Wang, A., Calhoun, B.H., Chandrakasan, A.P., Vittoz, E.A.: Sub-Threshold Design for Ultra Low-Power Systems, pp. 7–46. Springer, Berlin (2006)
2. Alliance: [www.zigbee.org](http://www.zigbee.org)

3. Liou, J.J., Schwierz, F.: RF MOSFET: recent advances, current status and future trends. *Solid State Electron.* **47**(11), 1881–1895 (2003)
4. International technology roadmap for semiconductors.: Radio frequency and analog/mixed-signal technologies summary (2013)
5. Silveira, F., Flandre, D., Jespers, P.G.A.: A gm/ID based methodology for the design of CMOS analog circuits and its application to the synthesis of a silicon-on-insulator micropower OTA. *IEEE J. Solid State Circuits* **31**(9), 1314–1319 (1996)
6. Jespers, P.: The gm/ID Methodology a Sizing Tool for Low-Voltage Analog CMOS Circuits: The Semi-Empirical and Compact Model Approaches, 1st edn. Springer, Berlin (2009)
7. Hsieh, H.-H., Lu, L.-H.: Design of ultra-low-voltage RF frontends with complementary current-reused architectures. *IEEE Trans. Microwave Theory Tech.* **55**(7), 1445 (2007)
8. Otis, B.P., Rabaey, J.: *Ultra-Low Power Wireless Technologies for Sensor Networks*. Springer, Berlin (2007)
9. Binkley, D.M.: *Tradeoffs and Optimization in Analog CMOS Design*. University of North Carolina at Charlotte, Wiley, USA (2008)
10. Do, A.V., Boon, C.C., Do, M.A., Yeo, K.S., Cabuk, A.: A subthreshold low-noise amplifier optimized for ultra-low-power applications in the ISM band. *IEEE Trans. Microwave Theory Tech.* **56**(2), 286 (2008)
11. Lee, H., Mohammadi, S.: A 3 GHz subthreshold CMOS low noise amplifier. In: *Proceedings of Radio Frequency Integrated Circuits (RFIC) Symposium*, Jun 2006
12. Perumana, B.G., Chakraborty, S., Lee, C.-H., Laskar, J.: A fully monolithic 260  $\mu$ W, 1 GHz subthreshold low noise amplifier. *IEEE Microw. Wireless Compon. Lett.* **15**(6), 428–430 (2005)
13. Cornetta, G., Santos, D.J.: Low-power multistage low noise amplifiers for wireless sensor networks. *Int. J. Electron.* **96**(1), 63–77 (2009)
14. Rastegar, H., Hakimi, A.: A high linearity CMOS low noise amplifier for 3.66 GHz applications using current-reused topology. *Microelectron. J.* **44**, 301–306 (2013)
15. Fiorelli, R.: All-inversion-region gm/ID based design methodology for radiofrequency blocks in CMOS nanometer technologies. University of Seville, Spain and Instituto de Microelectret de Sevilla, Spain, PhD thesis, p. 124 (2011)
16. Tiebout, M., Paporisto, E.: LNA design for a fully integrated CMOS single chip UMTS transceiver. In: *IEEE European Solid State Circuits Conference (ESSCIRC2002)*, Florence, Italia, pp. 825–828 (2002)
17. Sheaffer, D.K., Lee, T.H.: A 1.5-V, 1.5 GHz CMOS low noise amplifier. *IEEE J. Solid State Circuit* **32**, 745 (1997)
18. Shaeffer, D.K., Lee, T.H.: Corrections to a 1.5-V, 1.5-GHz CMOS low noise amplifier. *IEEE J. Solid State Circuits* **40**(6), 1397–1398 (2005)
19. Alvarado, U., Bistué, G., Adín, I.: *Low Power RF Circuit Design in Standard CMOS Technology*. Springer, Berlin (2011)
20. Aparin, V., Larson, L.E.: Linearization of CMOS LNA's via optimum gate biasing. In: *IEEE International Symposium on Integrated Circuit and System (ISCAS)*, pp. 748–751, Vancouver, CA (2004)
21. Toole, B., Plett, C., Cloutier, M.: RF circuit implications of moderate inversion enhanced linear region in MOSFETs. *IEEE Trans. Circuits Syst. I* **51**(2), 319–328 (2004)
22. Ho, D., Mirabbasi, S.: Design considerations for sub-mW RF CMOS low-noise amplifiers. In: *Vancouver Electrical and Computer Engineering*, British Columbia University (2007)
23. Baimei, L.I.U., Wang, C., Minglin, M.A., Shengqiang, G.U.O.: An ultra-low voltage and ultra-low-power 2.4 GHz LNA design. *Radioeng.* **18**(4), 527 (2009)