PHD

Symbolic tolerance and sensitivity analysis of large scale electronic circuits

Eberhardt, Friedemann

Award date:
1999

Awarding institution:
University of Bath

Symbolic Tolerance and Sensitivity Analysis of Large Scale Electronic Circuits

Submitted by Friedemann Eberhardt

for the degree of PhD

of the University of Bath

1999

Symbolic Tolerance and Sensitivity Analysis of Large Scale Electronic Circuits

Submitted by Friedemann Eberhardt

for the degree of PhD

of the University of Bath

1999

The work presented in this thesis was wholly undertaken while the candidate was registered as a part time student for the degree of doctor of philosophy at the University of Bath.

Friedemann Eberhardt

Abstract

The objective of this thesis is to develop design and characterisation techniques for analysing parametric deviation effects in large scale analogue circuits.

A general overview of issues involved in testing digital, analogue and mixed-signal circuits is presented. The different testing techniques are examined concerning their potential for diagnosing parametric deviations in analogue circuits.

The Self Test procedure originally devised for diagnosis of small discrete circuits is combined with a hierarchical modelling strategy which results in the development of a novel Hierarchical Fault Diagnosis Algorithm (HFDA). Due to its low requirements concerning computing time and measurement nodes, the HFDA can be used to diagnose large analogue integrated circuits during characterisation.

Symbolic analysis methods for analogue circuits are studied to evaluate their applicability to sensitivity and tolerance analysis of large scale networks. The hierarchical Sequence Of Expressions (SOE) approach is appropriate to reduce the inherent computational complexity.

Two novel SOE sensitivity analysis techniques are introduced. The techniques have the benefit of significantly accelerated calculation combined with an all-parameter sensitivity analysis. This makes the methods best suited in the application to parametric optimization of large scale analogue circuits during the design process.

The SOE approach is then used to develop an efficient tolerance analysis method. Statistical examinations are traditionally based on the Monte Carlo technique which has the disadvantage of being slow and limited for large scale circuits. Introduced by Glesner, the Quantile Arithmetic speeds up the tolerance investigations but shows a lack of accuracy. By deriving a Modified Quantile Arithmetic (MQA) the precision of tolerance analysis is significantly improved and calculations are further accelerated. The experimental results indicate that MQA runs typically 20 times faster than the Monte Carlo analysis and yields in most cases similar results.

Acknowledgements

A lot of thanks to my supervisor Dr Peter Shepherd for the chance to do this work as a part time student in his group at the University of Bath. I sincerely appreciate his guidance and encouragement throughout this research project and the constructive discussions. I am very obliged to Dr Chung Kin Ho for the fruitful collaboration within our PhD projects which resulted in the development of the hierarchical diagnosis procedure.

Many thanks are due to my project leader at the Robert Bosch GmbH, Dr Wilfried Tenten, for providing the initial direction of this thesis, for the encouragement and successful collaboration in this research. Special thanks to Dr Robert Kainer for the discussions on symbolic analysis and for the help with Maple. The hints of my colleagues in the department K8/EIS were very useful, thanks to Davide Buro, Peter Friedrichsohn, Arnd Gangei, Wolfgang Hellwig, Martin Wrede, Gerhard Wucher and all the others. I am also grateful to the students Thomas Horn and Dieter Klein for their support in this project.

I wish to express my thanks for the chance to do this research work within the Robert Bosch GmbH in Reutlingen. A lot of thanks to the department leader Mr Kalkhof and the group leaders Mr Klose and Mr Poppe for the opportunity to participate in design and test projects and to use the facilities of the department.

Contents

List of Figures

List of Tables

List of Acronyms and Symbols

ASIC	Application Specific Integrated Circuit
ATE	Automatic Test Engine
ATG	Automatic Test Generation
BIST	Built In Self Test
BJT	Bipolar Junction Transistor
BPT	Binary Partition Tree
CAD	Computer Aided Design
CCM	Component Connection Model
CMOS	Complementary Metal Oxide Semiconductor
CMRR	Common Mode Rejection Ratio
CUT	Circuit Under Test
DAG	Directed Acyclic (Expression) Graph
DFT	Design For Testability
DSP	Digital Signal Processor
FDA	Fault Diagnosis Algorithm
FFT	Fast Fourier Transform
HFDA	Hierarchical Fault Diagnosis Algorithm
HOTG	Hierarchical Optimal Tree Generation
IC	Integrated Circuit
IFA	Inductive Fault Analysis
KCL	Kirchhoff's Current Law
MNA	Modified Nodal Analysis
MOST	Metal Oxide Semiconductor Transistor
MQA	Modified Quantile Arithmetic
MUX	Multiplexer
OTG	Optimal Tree Generation
opamp	Operational Amplifier
PSRR	Power Supply Rejetion Ratio
pcf	Probability Cumulative Function
pdf	Probability Density Function
QA	Quantile Arithmetic
QSOE	Quantile Sequence Of Expressions

RMNA	Reduced Modified Nodal Analysis
RSS	Root Sum Square
SAT	Simulation After Test
SBT	Simulation Before Test
SC	Switched Capacitor
SCAPP	Symbolic Circuit Analysis Program with Partitioning
SOE	Sequence Of Expressions
ST	Self Test
TDG	Top Down (Expression) Graph
VDD	Positive Supply Voltage
VHDL	Very high speed integrated circuit Hardware Description Language
VLSI	Very Large Scale Integration
VSS	Negative Supply Voltage

A	Amplification
$\mathbf{a}$	CCM input vector
a_i	Network Function Numerator Polynomial Coefficient
$\mathbf{b}$	CCM output vector
b_i	Network Function Denominator Polynomial Coefficient
$\mathrm{cov}(u, v)$	Covariance of Random Variables u and v
C	Capacitance
e	Edge Index
G	Conductance
H	Network Function
H_i	i^{th} SOE Expression
i	Current
K	Number of Clock Phases
k	Clock Phase Index
$\mathbf{L_{ij}}$	Connection Matrices
l_{DAG}	Length of a DAG Path
l_{BPT}	Length of a BPT Path
m	Clock Cycle Index (Chapter 3)

m	Number of Terminal Blocks (Chapter 5)
m	Number of QSOE Expressions (Chapter 6)
N	Number of Discretization Points
N_{adds}	Number of Additions
N_e	Number of Edges of the Expression Graph, DAG or TDG
N_i	Node i in the Expression Graph, DAG or TDG
N_n	Number of Nodes of the Expression Graph, DAG or TDG
N_{mults}	Number of Multiplications
N^{BPT}	Node of the BPT
N^{DAG}	Node of the DAG
N_X	Number of Circuit Parameters
n	Node Index (Chapter 3)
n	Number of Resistors in the Ladder Circuit (Chapter 4)
n	Number of Circuit Nodes (Chapters 5-6)
n_i	Node Weight in Expression Graph, DAG or TDG
P	Discrete Probability Density Function
$\tilde{P}$	Continuous Probability Density Function
$\tilde{P}_{cum}$	Continuous Probability Cumulative Function
Q	Charge (Chapter 3)
Q	Filter Q-value (Chapters 4-6)
$Q(w)$	Quantile Function
q_i	i^{th} Quantile
R	Resistance
ρ_{uv}	Linear Correlation Coefficient of Random Variables u and v
S	Number of Monte Carlo Samples
s	Laplace Frequency Variable
$sen(H, x)$	Sensitivity of Network Function H to parameter x
σ_u	Standard Deviation of Random Variable u
τ_i	Tolerance Decision Threshold of ith Element
$\mathbf{u}$	stimulus vector
v	Voltage
w	Probability
$w(i, j)$	Edge Weight in Expression Graph, DAG or TDG

X	Set of Circuit Parameters
X_0	Nominal Point in the Parameter Space
x_i	i^{th} Circuit Component Parameter
$\overline{x}$	Mean Value of x
Y_i	Output Admittance of Ladder Circuit with i Resistors (Chapter 4)
Y_i	i^{th} Discretization Point of Random Variable Y (Chapter 6)
Y_{yield}	Yield
$\mathbf{y}$	test point vector
y_{high}	Discretization High Cut-Off Point
y_{low}	Discretization Low Cut-Off Point
$\mathbf{Z}$	Component Transfer Matrix
z	z Transform Variable
z_i	i^{th} QSOE Expression
(i, j)	Edge from Node i to Node j in Expression Graph, DAG or TDG

Chapter 1

Introduction

Advances in process technology nowadays allow the realization of complex integrated circuits (ICs). Application-specific ICs (ASICs) have moved towards the integration of complete systems which include both digital and analogue parts on a single chip. These mixed-signal very large scale integration (VLSI) chips require computer-aided design (CAD) tools which help the designer to handle rising circuit complexity and thereby reduce design time and cost.

In the digital domain, there exist numerous efficient simulators supporting the designer in verifying their circuits. Simulations can be performed at different levels of hierarchy allowing the analysis of very complex networks. This development was encouraged by the definition of high level hardware description languages such as VHDL (Very high speed integrated circuit Hardware Description Language). The abstract circuit description within VHDL is not only useful for fast behavioural level simulation but also provides the possibility for a complete digital silicon compilation.

The situation in the analogue domain is totally different. Analogue simulation is potentially much more involved than digital simulation. This is mainly caused by the continuous nature of analogue signals and the diversity of analogue behaviour. Analysis

is mostly carried out at the transistor level of description which makes simulation slow and cumbersome. Currently, effort is put into the definition of an analogue/mixed-signal hardware description language to alleviate high level simulation of large analogue systems. However, due to the complexity of analogue behaviour and loose form of hierarchy it is very difficult to automate parts of the analogue design process and provide CAD tool support. As a consequence, analogue design is mostly performed manually which is time consuming and error prone.

A further challenge of analogue design is that parameter tolerances play a much more important role for the circuit behaviour than in the digital domain. The tolerances associated with all manufactured components cause performance variations of the mass-produced analogue circuits, sometimes to the extent that the specification of the customer will be violated. As the specifications of high performance applications become more aggressive and the IC structures are scaled down, the consideration of parametric variations during the design process turns out to be very important.

Tolerance analysis helps the designer tackle the parameter variation challenge by predicting the performance spreads of the circuit. This information can be used for yield estimation before fabrication is started. Yield is an important factor for product cost assessment and economic planning. Moreover, if the estimated yield turns out to be unacceptably low, the results of tolerance analysis are useful for optimizing the circuit with respect to reduced performance variations (tolerance design).

Unfortunately, traditional approaches to tolerance analysis are either very time-consuming or inaccurate. This situation is aggravated by increasing IC complexity which makes tolerance analysis more and more involved. In the case of large scale analogue systems where a single simulation may run hours or even days, performing a Monte Carlo analysis, for example, is clearly prohibitive. Consequently, tolerance considerations are

typically accomplished ad hoc by assuming boundary conditions based on the designer's experience. This process however, is extremely failure prone and critical.

As a result of the situation described above the product behaviour may be suspect when the chip is passed on to fabrication and test. The economic consequences of parametrically critical designs are manifold. Based on the experience of the author in the design and test departments of Bosch Microelectronics these consequences may be summarized as follows.

- **low yield:** devices which are not in accordance with customer's specification can not be sold and increase the cost of the product and decrease profit.

- **high test cost:** marginal and parametrically critical designs typically require a large amount of additional specification driven testing under many possible situations in order to guarantee the quality of the product. For automotive applications, for example, critical designs need to be tested under many different environmental conditions, e.g. different temperatures, resulting in production test programs which run much longer than in the case of uncritical designs.

- **risk for quality:** even when many specification tests are applied, it is possible that marginal behaviour remains unobserved during testing until discovered in the field. In the case where devices exhibit a non-linear temperature dependency, for example, it is difficult or even impossible to define the most critical temperatures for test. This causes a quality risk.

- **increased time to market:** when parametric problems are discovered during characterisation (or even later on), a redesign becomes necessary in many cases in order to increase yield and guarantee quality. A redesign causes a further loop through the design-, layout- and mask development process which severely affects time to market. This delay cannot be accepted in most applications where the time to market

is critical for profit.

- the situation described in the previous item is aggravated by the difficulty to find the reason for a paramteric problem during characterisation. This is mainly due to the low accessibility to circuit nodes of analogue ICs which makes the localization and diagnosis of the responsible circuit block and/or parameter cumbersome. Moreover, the complexity of large scale analogue circuits often requires several loops of redesigns before a problem is fixed which may cause a delay of several months.

This summary shows a strong need for tool support which helps the designer in

1. tolerance analysis to make their design robust against the unavoidable parametric variations of the manufacturing process.

2. test point selection and parametric fault diagnosis in order to fix parametric problems as fast as possible.

Despite this strong need, the current methods available are impractical for today's large scale analogue circuits, mostly due to their huge computing time requirements.

The main goal of this thesis is to propose and investigate techniques for effective tolerance analysis of analogue circuits. An important focus of this work is mainly to reduce the computational expense of the methods such that they become usable during the design process for large scale analogue networks.

In addition, a method for the diagnosis of parametric deviations in analogue ICs is examined. This method helps the designer to define suitable test points and alleviates the localization of parametric design problems during characterisation.

The layout of the thesis is as follows.

Diagnosis of parametric deviations is strongly related to test issues. For this reason, Chapter 2 presents an overview of testing digital, analogue and mixed-signal circuits. The

role of fault modelling within test preparation and evaluation is highlighted. Emphasis is drawn to the various techniques of analogue fault modelling and testing and to the difficulties related to analogue test. Based on the review of this Chapter, a method is chosen which can be developed for parametric fault diagnosis of large scale analogue ICs to help the designer in the characterisation process.

In Chapter 3, the parametric fault diagnosis approach which has been chosen is described. Its benefits and limitations are outlined. The limitations are mainly due to computing time and applicability to integrated circuits. By combining the fault diagnosis method with a hierarchical modelling strategy, the computational expense and the number of test points required are reduced. By this means, the *hierarchical fault diagnosis algorithm* developed becomes applicable to large scale analogue integrated circuits. Issues concerning the diagnosis of switched capacitor circuits are also considered. Finally, experimental results are presented to investigate the performance of the new fault diagnosis algorithm.

Having investigated parametric fault diagnosis, the following chapters of the thesis concentrate on tolerance analysis of large scale analogue circuits. Symbolic analysis has been chosen as the basis for the techniques developed in this thesis. The underlying reason is that tolerance analysis is a highly iterative task where typically many circuit simulations are required. The advantage of symbolic analysis is that only one simulation run is needed in which a symbolic expression for the circuit behaviour is generated. During tolerance analysis, successive evaluations of the symbolic expression replaces the necessity for any extra numerical iterations through the simulator. In this way, large savings in computing time are achievable.

The purpose of Chapter 4 is to review the current state of the art in symbolic analysis of analogue circuits. The capabilities and limitations of symbolic techniques are discussed.

A main focus of this review is on the applicability of symbolic methods to large scale networks. Herein an important criterion is the number of terms in the symbolic network function. A comparison of the currently available methods shows that symbolic hierarchical decomposition is best suited to handle large scale systems. This method produces a *sequence of expressions* in which the number of terms grows only approximately linearly with circuit size making the approach very attractive for tolerance analysis.

An important technique which helps in tolerance analysis and tolerance design is sensitivity analysis. In Chapter 5 the range of applicability of sensitivity analysis and the available numerical and symbolic sensitivity techniques are reviewed. The previous symbolic procedures which are based on the sequence of expression approach are discussed in detail. The drawback of the previous techniques is that they still require a large number of arithmetic operations when the sensitivities with respect to many or all parameters need to be determined. Effective multi-parameter sensitivity analysis, however, is essential for the application of sensitivity methods within tolerance analysis. Two novel sequence of expression methods for fast multi-parameter sensitivity are then developed: the *balanced* symbolic sensitivity analysis, and the *parallel* symbolic sensitivity analysis. The first technique reduces computational complexity with the aid of a hierarchical balanced partitioning strategy. The second one uses the sequence of expressions to calculate the sensitivities with respect to all parameters in parallel. Applications of the methods to large scale circuits are presented to demonstrate the effectiveness of both approaches.

Sensitivity analysis is useful to get a first insight into tolerance behaviour. However, to examine the effects of parameter variations more accurately, additional methods are required. For this purpose, Chapter 6 presents a novel symbolic tolerance analysis technique. The method is based on *Quantile Arithmetic* which computes circuit tolerances using discretization of random variables. The advantage of Quantile Arithmetic is that it

runs approximately one order of magnitude faster than the Monte Carlo analysis. However, the cost for this speed increase is a lack of accuracy. By defining a new *Modified Quantile Arithmetic* the accuracy is significantly enhanced and, at the same time, execution speed further increased. The Modified Quantile Arithmetic is then combined with the *sequence of expression* approach. This yields an effective method for symbolic tolerance analysis of large scale analogue systems. Modified Quantile Arithmetic is then applied to circuit examples and comparisons with the results of other tolerance analysis methods are accomplished.

In Chapter 7, the main conclusions of the work in this thesis are presented and directions for future research work in tolerance analysis and parametric fault diagnosis are outlined.

Chapter 2

An Overview of Testing

In this chapter, the reader is introduced to test preparation, generation and evaluation techniques. The aim is to find a method which can be developed for parametric fault diagnosis of analogue circuits to support the IC characterisation process.

The test consists of mounting the fabricated chip on the Automatic Test Engine (ATE), applying stimuli to the input pins and comparing responses at the output pins with those expected. The target of testing is either pure *detection* (production test, go/no-go testing) or, a more rigorous demand, the *localization* (diagnosis) of faults.

Test generation techniques should provide tests of high quality at minimal cost. *Test quality* is normally denoted by the term fault coverage. It is defined as the fraction of faults which are detected by the test sequence:

$$\text{Fault Coverage} = \frac{\text{number of detected faults}}{\text{total number of possible faults}}. \tag{2.1}$$

Test cost may be broken down into two categories with the process of test generation and test application respectively. The former is a one-time cost determined by the computational expense to generate the test vector set. The latter is a recurrent cost and refers to the time it takes to apply the vectors to the Circuit Under Test (CUT) on the one hand and to on-chip test circuitry, additional test pins and the ATE equipment on the other

hand.

Basically, there exist two different classifications of test strategies [1, 2, 3]: *functional* testing and *structural* testing. Within the functional approach, the CUT is tested for fulfilment of the design specification. Typically, starting from a characterisation test program the final production test program is generated by empirically omitting tests which seem to be redundant and adding some quality related screens. Unfortunately, this approach often produces test programs with a runtime being orders of magnitude too long. Additionally, specification testing is mostly purely go/no-go testing and a localization of the cause of problems from functional test results is difficult. Moreover, it is not clear whether the omitted tests are really redundant. Since these tests are not focused on the defects that might occur during fabrication and during lifetime of the product, there always remains a risk for the product's reliability.

On the other hand, structural testing is defect-oriented. The IC is supposed to function correctly as long as there is no defect on the chip. Therefore, the tests aim at detecting faults which are caused by physical defects. Structural test generation follows the route shown in Figure 2.1. The starting point is the analysis of the physical defects which might occur during fabrication of the chip. Fault models are abstracted from the underlying physical analysis by mapping the defect to the appropriate electrical faulty behaviour. Based on fault models, automatic test generation algorithms and fault simulations can be applied. The result of fault simulation is an estimate for the fault coverage of the test set under consideration. Usually, the structured approach allows for generation of much cheaper and more effective tests than in the case of functional testing. However, due to the need for the product specification to be guaranteed to the customer, total elimination of functional testing seems unrealistic.

Research into digital fault diagnosis and test generation is far ahead of its analogue

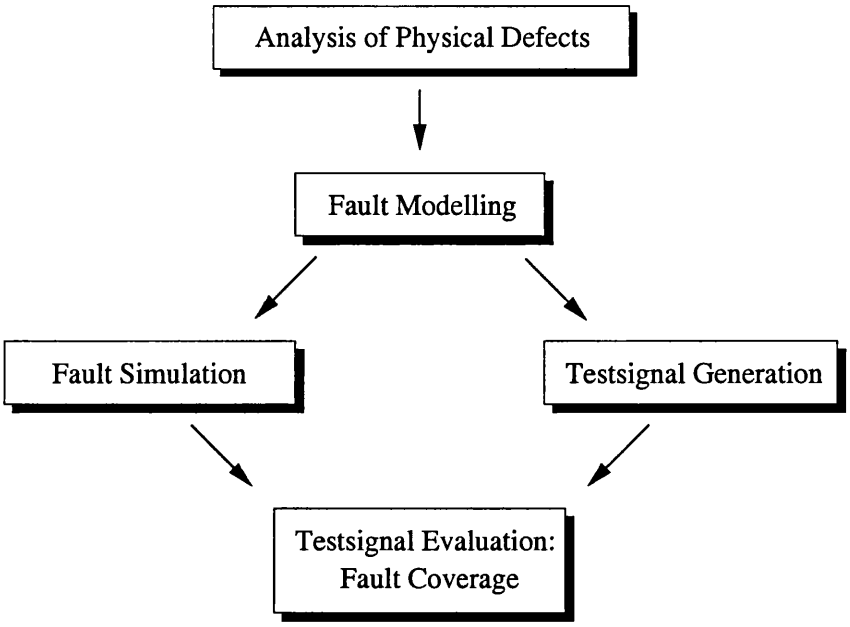


Figure 2.1: Steps of structural test generation

counterpart. The development of a structured test generation methodology in this domain was encouraged by the availability of realistic, easy to analyse fault models. Therefore it is worth having a look at the digital test development for VLSI devices first. This overview continues by reporting the current situation in the analogue domain and classifying the different approaches in analogue fault modelling and test generation. Based on this overview the approach to analogue parametric fault diagnosis used in Chapter 3 is chosen.

2.1 Digital Testing

A detailed review concerning automatic test generation in the digital domain is given in [4]. The requirements on manufacturing tests are very high, ideally these tests should check whether all components and interconnections are fabricated correctly. Manufacturing tests are generated automatically by procedures which are based upon fault models.

2.1.1 Digital Fault Modelling

Failures can have diverse causes [5]. There are intrinsic failure mechanisms which are defects coming directly from the process like ionic contamination, charge trapping at the oxide interface of a metal oxide semiconductor transistor (MOST), or missing/added interconnections. Moreover there are extrinsic failure mechanisms originating for example from packaging or bonding of the chip. Additionally, environmental effects like radiation or stress caused by electrostatic discharge need to be considered.

In order to analyse the faulty behaviour and to develop techniques to detect and locate failures it is mandatory to abstract from the origin of the failure. Fault models are generated as a representation of the failure at the level of description at which the analysis should take place. Fault models allow cost effective development of test stimuli that identify failed chips and, if necessary, diagnose the failure. They also limit the number of necessary tests as opposed to applying all possible tests. Fault models are technology dependent and one has to make compromises concerning the complexity of the models necessary for accuracy against the tractability of analysis. Referring to these criteria the most significant digital fault models are

- **Stuck-at Fault Models:** Stuck-at faults are defined as a faulty property of interconnecting lines [6]. A line with a stuck-at-1 fault will always have a logical value of 1 irrespective of the correct logical output of the gate driving it. So every node of the circuit may have 3 possible states: *stuck-at-1*, *stuck-at-0* or *fault free*. Stuck-at fault models are functional fault models based on the logical description level of digital systems, in which the circuit is represented as an interconnection of logical gates. They are the simplest models to analyse and have proven to be very effective in displaying the faulty behaviour of actual devices.

- **Stuck-open, Stuck-short Fault Models:** Up to 1978 it was believed that the

stuck-at fault models were sufficient to describe defects at the logical level. Unfortunately, circuits implemented in CMOS (Complementary Metal Oxide Semiconductor) technology could display fault modes other than stuck-at faults, called *nonclassical* faults [7]: e.g. *stuck-open* or *stuck-short.* The main point is that tests generated for stuck-at faults are no longer valid in detecting all the nonclassical faults. These faults are modelled at the switch level description as conducting or not conducting paths irrespective of the transistor gate input. Because of the low level of description these models are more complicated to analyse.

- **Bridging Fault Models:** Bridging faults, shorts between adjacent signal lines, need extra modelling and analysis [8, 9].

- **Physical Fault Models:** Besides functional faults, parametric properties must also be considered, e.g. leakage current and timing.

- **Hierarchical, Functional Fault Models:** In order to reduce computational expense of test generation procedures, fault models at a hierarchical functional level are proposed. An example, concerning microprocessors, can be found in [10].

In spite of all of their shortcomings, the stuck-at fault models are a standard most commonly used in industry. Test algorithms and CAD tools normally rely on this standard. But with increasing clock speed and scaling down IC dimensions other failure modes become more and more relevant.

2.1.2 Digital Test Generation

The goal of test generation is to obtain stimuli test vectors for the detection of the modelled faults. The stimuli vectors must cause output vectors in the faulty case which are different to the fault-free case. Automatic Test Generation (ATG) techniques may be divided into

the

- **Exhaustive method:** If the number of primary inputs is small, application of all possible input vectors to a combinational circuit will ensure 100% fault coverage. The *exhaustive* method has very low computational cost and may be applied quickly. It can be extended to more complex circuits as long as the logical partition of the circuit into smaller subcircuits is possible.

- **Random method:** The *random* method is another inexpensive way to generate tests. In this technique test vectors are generated successively by a random number generator. Using a *fault simulator* and the fault models of the CUT it can be found out whether an additional fault can be detected by the vector under consideration. For combinational circuits it has been proven to be a good method as long as the number of levels of logic and gate fan-ins is not too large [11]. Random test generation programs for sequential circuits require special considerations concerning clock and control signals.

- **Algorithmic method:** This method relies mostly on stuck-at fault models and implements test generation algorithmically based on the principles of *controllability* and *observability*. One of the oldest procedures for combinational circuits is the D-algorithm [12]. The Boolean Difference method captures similar ideas in algebraic terms [13].

Because of growing circuit complexity, test generation at the hierarchical/functional level has attracted some attention also. In addition, the consideration of nonclassical faults led to completely new test methods [14]. One example concerning CMOS technology is the IDDq-test, in which the supply current is observed after all switching transients have decayed.

Test generation for sequential circuits remains a challenge. This provides the motivation for Design for Testability (DFT) methods, e.g. *scan design.* Sequential circuits, which are implemented by using scan design, can be tested like combinational circuits. The IEEE Standard 1149.1 covers a test access port and a boundary scan architecture for the board level, thereby enabling a unified test procedure for boards with ICs from different manufacturers. Additionally there are techniques for on-line testing of circuits such as *Built-In-Self-Test* (BIST).

2.1.3 Conclusion

In the digital domain, there are solutions concerning fault modelling and automatic test generation allowing for a structured test approach. Algorithms are implemented as CAD tools and actually used by industry. The main reason why automatic digital test preparation is very successful is the availability of easy to analyse fault models which describe a large number of possible physical defects. Fault models form the basis for test generation procedures and can be used in fault simulations leading to an assessment of test quality of the test set under consideration. However, as operating frequencies are increasing and operating voltage is reducing, digital circuits are beginning to show analogue behaviour especially during signal transitions [15] which will complicate testing in the future.

2.2 Analogue Testing

Currently, no clear concepts for a structural testing approach and no tools for ATG are available in the analogue domain. As a consequence, test preparation is up to now based on a functional approach. This process depends on the experience of the test and design engineer and the resulting test sequences merely check the specification and critical functions. This often results in expensive and sometimes even in improper test sequences.

The main reasons why analogue testing is less advanced than the digital counterpart can be attributed to the following:

- **accessibility to circuit nodes of ICs:** besides adding some pin overhead, the insertion of test nodes is very critical because sensitive analogue behaviour may be compromised.

- **continuous signals:** analogue signals are continuous in nature. For this reason, there are many more modes of failure in analogue circuits than in digital circuits.

- **tolerance problem:** one has to consider tolerances of the circuit components. Since these tolerances may be relatively large (e.g. 50% in the case of integrated resistors) measurement results are not easy to evaluate in a deterministic manner, and the definition of pass/fail limits often remains fuzzy.

- **diversity of functionality:** analogue functionality is quite diverse. In the case of an amplifier, voltages, currents, gain, bandwidth, offsets, input and output impedances are important parameters and measurements may be performed with respect to time, frequency and different temperatures.

- **lack of analogue fault model:** because of the three previous items there exists no generally accepted analogue fault model to date.

- **simulation time and accuracy:** increasing circuit complexity and the continuous nature of analogue signals make circuit simulation slow and sometimes inaccurate. This causes fault simulation to be extremely time-consuming, mostly to an extent that its application is prohibitive in the test preparation stage.

- **critical designs:** critical and marginal designs complicate testing and often require additional test steps under various parametric conditions (e.g. temperatures).

Since fault models are required as the basis for a structural test approach researchers have paid close attention to analogue fault modelling during the last decade.

2.2.1 Analogue Fault Modelling

As in the digital case, fault models must represent the effect of process shortcomings at the electrical level. On the one hand, they should be as precise as necessary to describe the electrical failure correctly. On the other hand, the models should be easy to analyse to make their application in fault simulation and test generation procedures tractable even for larger circuits. Obviously, to fulfil these requirements, some compromise is needed. According to [15], analogue fault models can be classified as

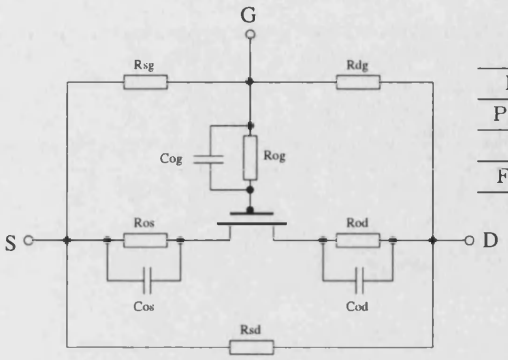
- **hard/structural:** These models describe faults which are caused by random spot defects of the process. The structure, i.e. the topology, of the circuit is changed by an open or short circuit situation often resulting in a complete electrical malfunction.
- **soft/parametric:** These models describe faults which are caused by a component parameter exceeding the tolerance band. Typically, there is no complete failure of the circuit, but an out-of-specification behaviour.
- **hierarchical/behavioural:** These are descriptions of either of the two faults, hard or soft, at a higher level, especially at the level of typical analogue functional blocks. This modelling methodology becomes obligatory in the case of large scale circuits.

The different fault models are now discussed in detail.

Hard Fault Models

One may distinguish between the implementation of the fault models and the derivation of these models from the underlying defects.

Implementation of Hard Fault Models: Hard faults are mostly modelled at the device level by introducing additional equivalent circuitry, e.g. in the transistor model. In Figure 2.2 an example of a parametrised hard fault model for a MOST is presented. An



Fault #	1	2	3	4	5	6
Parameter	Rsd	Rsg	Rdg	Rog	Ros	Rod
Faulty	10Ω	1Ω	1Ω	1GΩ	10MΩ	10MΩ
Fault free	1GΩ	1MΩ	1GΩ	1mΩ	1mΩ	1mΩ

Figure 2.2: MOST hard fault model used in [16] for Fault Simulation

open conducting line is modelled by a large resistance whereas a shorted line is modelled by a tiny resistance. The capacitors represent the remaining coupling of conducting lines in the open circuit situation. Considering gate oxide shorts, more refined models which take into account the actual position of the conducting path in respect to the transistor channel can be found in [17]. A similar fault model for a bipolar junction transistor (BJT) is described for example in [18].

Naturally there are not only faults within a device but also between devices. Principally, modelling of hard faults affecting the overall topology of a circuit is the same as for single devices. Additional conducting lines, parasitics and resistors are introduced and several publications on test generation, e.g. [19], are based on this approach.

Obviously the consequences of faults in the analogue domain are more diverse than in the digital. In the latter, it is sufficient to model faults by setting signal lines to the exactly defined value 0 or 1, or by considering the respective signal lines completely opened or closed. The results gained with fault models in the analogue case are much more sensitive to the exact properties of the model. Consider, for example, an open gate in a MOST. In-

serting a high-value resistor to model this fault is likely to generate incorrect results since the circuit's behaviour in the open fault case might be unpredictable, whereas the resistor model produces deterministic results. Additionally, since some faults are completely unrealistic in respect of process properties and the IC layout, simply introducing shorts and opens into a circuit's topology might be misleading. Therefore it is necessary to analyse the defects at the process level, to get more information about the fault characteristics and ideally to provide a derivation of fault models.

Derivation of Hard Fault Models: For generating realistic fault models for ICs a knowledge of the failure mechanisms, i.e. the physics, and the related effects at the electrical level is important. In [5] a review of various defect mechanisms is given. Since technologies are diverse, the defect mechanisms are quite different. There are defect mechanisms in the substrate, in dielectric layers, defects caused by metalization, at interconnections, package bonding, overstress, both electrical and mechanical.

There exist well-proven techniques for deriving faults at the electrical level given a circuit and its layout, i.e. relating process shortcomings with their electrical effects. One of the oldest systematic methods is Inductive Fault Analysis (IFA) [20]. Although often applied to generate fault models for digital circuits, the procedure is also valid for hard fault model generation of analogue circuits. IFA involves three major steps:

- Identification of key physical defect mechanisms that occur in the IC process. IFA is concentrating mainly on local disturbances. These so-called "spot defects" are modelled as a flat disk of extra or missing material that may occur in any conducting, semiconducting or insulating layer of the IC. The model of the spot defect correctly takes into account the density of spots and the size probability distribution function.

- Distributing the spot defects over the IC surface in a Monte Carlo fashion.

- Mapping of the defects on circuit-level faults: For each defect a series of fault analysis procedures is called, to examine the layout geometry in the neighbourhood of the defect in order to determine if any circuit faults, i.e. shorts or opens, have occurred.

The result of IFA is a ranked list of circuit-level hard faults (ranking according to the probability of occurrence of the faults). IFA was implemented in a simulator called VLASIC [21] and successfully applied in test generation, failure analysis and defect parameter extraction.

IFA has several drawbacks. Very limited data is available from fabrication lines describing spot defects and their characteristics. Therefore IFA fault modelling tends to assume defect statistics. Moreover, every defect is assumed to deform only one layer and the three dimensional structure of contaminations is not modelled. To overcome these shortcomings and to simulate the interaction of different layers with contamination deposited on the wafer, an Inductive Contamination Analysis (ICA) was developed [22] and implemented as a simulator called CODEF [23]. One of the major disadvantages of both techniques, IFA and ICA, is the inherent computational complexity which is aggravated by the required Monte Carlo loop. One alternative to avoid the Monte Carlo loop is predicting the fault probability based on the critical area concept [24].

A way of obtaining realistic hard fault models at the circuit level at reduced computational cost is proposed in [25]. It is based on the observation that analogue ICs are composed of only a few frequently occurring building blocks, such as current mirrors, differential stages, etc. These blocks can be identified from the layout. By investigating such typical structures with respect to the impact of characteristic defects, a dictionary of realistic circuit faults can be generated. A method to obtain a fault list in an earlier design stage merely based on the schematic (netlist) is proposed in [26]. The advantages of these approaches are reduced computational cost on the one hand and early availability of

a realistic fault list on the other. However, these methods lack generality. An alternative technique [27] combines the generality of IFA with regression analysis to derive explicite models which describe the probability of occurence of faults in different transistor structures. These models are used to predict realistic faults before going to the final layout of a circuit.

Soft Fault Models

The statistics of an IC fabrication process manifests itself in a variation of layer thickness, doping concentrations, length and width of structures, mask alignments, etc. This has to be taken into account by soft fault models. Up to now, process statistics examinations have rather aimed at design evaluation, considering manufacturing yield prediction and design centering. Since component structures in ICs are scaled down, these parametric deviations also gain importance in the characterisation and testing stage.

Implementation of Soft Fault Models: Soft fault models are implemented as a parameter value outside of an acceptable tolerance box. Modelling is carried out at the transistor level in most cases and a number of publications on test generation and fault diagnosis are based on this fault model, e.g. [28]. Due to the continuous character of parameter variations there is an infinitely large number of different fault states of a circuit. As a consequence, the generation of a fault list for the circuit is less promising.

Typically, the tolerance boxes are postulated and not deduced. However, to be significant for any real application, the tolerances must correlate with the actual test strategy. Concerning a structural testing approach, it should be determined whether all devices are fabricated correctly. In such a case, the tolerances of the components are defined by the statistical properties of the process. Applying functional testing, the acceptable tolerances of the circuit components are dependent on the actual design and the performance

specification of the IC.

In the test optimization strategy of [29] parametric faults are based on tolerances in process parameters, and are modelled by statistical distributions. This approach supports the structural test approach and relies on the user to identify critical parameters and supply a parameter model of process fluctuations. A fault modelling approach supporting functional testing is described in [30]. Adopting the designer's point of view, a fault is defined as a deviation from the performance specification. The faulty performance is mapped to the measurement space of the actual test. The outcome of this procedure is a minimal set of measurements necessary for characterising the state of the CUT. However this was only accomplished for a small circuit and with simplifications.

Derivation of Soft Fault Models: Two different approaches are found in literature for deriving IC device parameter statistics (e.g. the electrical parameters of a MOST) caused from process characteristics. The first one (left side of Figure 2.3) is based on process simulation, e.g. [31]. The starting point is the information about the process parameters — such as times and temperatures of diffusion steps — and the respective disturbances.

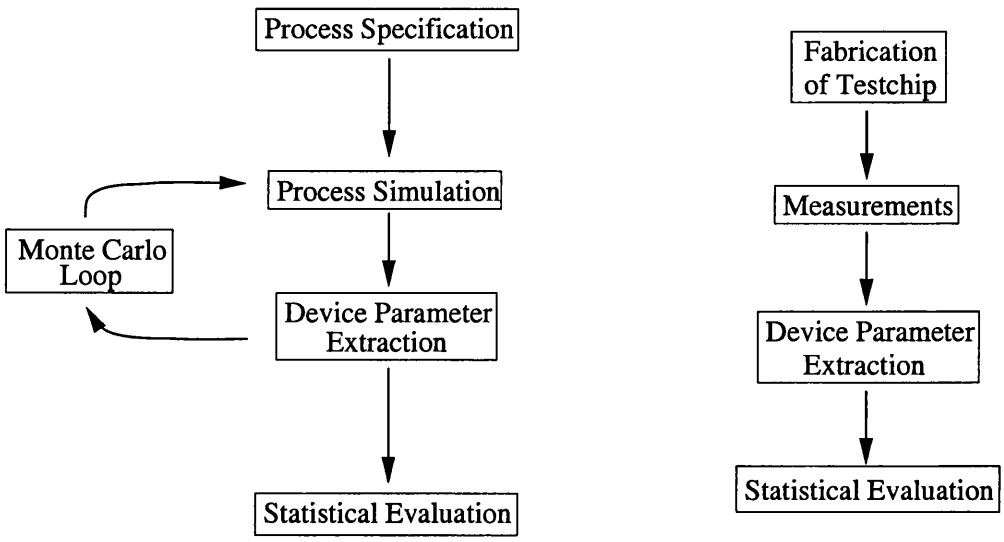


Figure 2.3: Device parameter statistics by: Left: process simulation, Right: testchip measurements

Then the process steps are simulated by the use of an IC-process simulator. The results of this step are for example doping profiles or dielectric layer thickness. After that, a device simulator extracts the device parameters. The last two steps are carried out in a Monte Carlo loop leading to a statistical sample of device parameter sets. The success of process simulation depends on the modelling accuracy of the physics of each single process step. Given today's complex processes with more than a hundred distinct steps, process simulation is not promising for reliable results.

An alternative way to obtain the device statistics is based on measurements (right side of Figure 2.3). A testchip including the device structure of interest is fabricated. The device characteristics are measured and the model parameters are extracted. The number of dice manufactured should be as large as possible to obtain statistical confidence in the results. Typical uses of this method for CMOS technology are found in [32, 33]. The test chip approach circumvents the problem of process modelling and its accuracy. The major difficulty is the availability of enough test chip data and the strong correlations between devices fabricated on chip (e.g. matching properties).

Hierarchical Fault Models

Fault models have been mostly formulated at the transistor level in the analogue case. This causes their application in fault simulation and test generation procedures to be extremely time expensive. The situation is aggravated by the large number of possible faults and the increasing complexity of today's large scale analogue and mixed-signal circuits. To overcome this problem, approaches in hierarchical fault modelling, both for hard and soft faults, have been proposed. For typical circuit blocks, e.g. operational amplifiers (opamps), behavioural models are developed which describe merely the terminal behaviour of the block. By this means, the computational expense for fault simulation of the overall circuit can be reduced significantly.

Starting from transistor level faults, the hierarchical circuit block is simulated and the faulty behaviour is extracted. The faulty behaviour may be modelled by using an fault-free hierarchical model and placing extra passive elements around the terminals in an appropriate manner [34]. For an opamp this is illustrated in Figure 2.4. Herein the

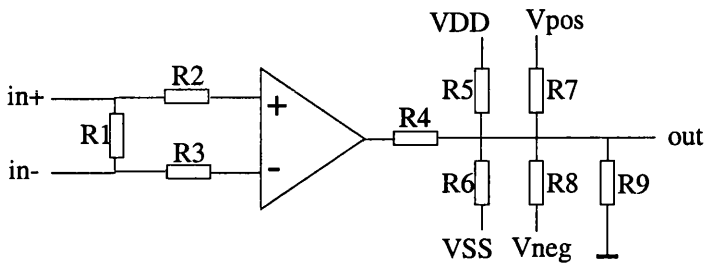


Figure 2.4: Hierarchical fault model with external resistors [34]

resistors at the opamp output are connected to diverse supplies depending on what value the output is stuck-at, hence modelling the hard faults. Modelling parametric faults hierarchically can be done by suitably varying the behavioural model parameters, e.g. gain and bandwidth of an opamp.

Generally, a set of transistor level faults may cause the same or almost the same faulty behaviour at the hierarchical block level. Then it is possible to collapse the set of transistor level faults to just one fault of the hierarchical block. This further reduces the computing time of fault simulation.

In [35] an opamp fault macromodel describing dc and ac faults is presented. The model is derived based on short/bridging faults at the transistor level. A more general concept to derive macromodels of faulty circuit blocks is described in [36]. Using neural network techniques, a fast mapping between the faulty block behaviour and the respective behavioural model parameters is achieved. In [37] neural networks are used for characterisation of analogue macromodels under fault conditions. Additionally, the transistor

level faults which result in similar behaviours of the circuit block are grouped into one fault at the hierarchical level. This further reduces fault simulation time by reducing the number of required simulations [38]. In [39], each circuit block is replaced by its behavioural model, except the subcircuit in which faults are injected, which is described by its layout extracted netlist. This approach avoids the complicated hierarchical modelling of faulty blocks and needs only behavioural models of fault-free subcircuits. Behavioural fault modelling of digital-to-analogue converters can be found in [40].

One way to generate behavioural soft fault models is a Monte Carlo analysis with the device parameter statistics as input. As a result the deviations and correlations of behavioural parameters of a circuit block may be obtained [41]. An approach to parametric testing based on behavioural modelling is presented in [42]. Using measurement results the behavioural model parameters are estimated and a good-bad decision with respect to the hierarchical component can be achieved.

The main criticism against the hierarchical approach is the lack of accuracy. The question is whether the behavioural models are capable of representing the transistor level faults with sufficient precision. Additionally, it is not clear whether an out of specification output signal of a previous faulty stage is propagated correctly by the behavioural models of succeeding stages. However, hierarchical modelling is mandatory for handling today's complex systems, otherwise computing times become prohibitively long and fault simulations cannot be performed. Moreover, the aim in behavioural fault modelling is mainly to differentiate a good circuit from a faulty one rather than to simulate the circuit responses with high accuracy. Since test preparation based on behavioural models is conceptually close to specification testing, this approach may help in future to link the structural test approach with the industrially applied functional testing [15].

In Chapter 3, a hierarchical approach will be adopted to develop a method which

makes parametric fault diagnosis of large scale analogue circuits possible.

2.2.2 Analogue Test Generation

Reviews of analogue test generation can be found in [43, 44, 45]. Basically, test generation and fault diagnosis techniques for analogue circuits may be divided into Simulation Before Test (SBT) and Simulation After Test (SAT) methods.

SBT-techniques

As the name implies, with these techniques the simulations are carried out before the actual test is accomplished. Firstly, a set of measurement nodes and test signals is chosen. Starting from a fault list containing the fault models for the faults under consideration, fault simulation is applied and the signatures for the normal and each faulty condition of the circuit are extracted. This information is stored in a *fault dictionary.* During testing, the measured signatures can be compared with the stored ones. A fault is diagnosed when its corresponding signature matches the measured one within a defined tolerance. Practical implementations of SBT-methods differ mainly in the techniques used for establishing and handling the fault dictionary, [46, 47]. An algorithm which selects a minimal set of measurement nodes for the fault dictionary approach can be found in [48]. The use of a fault dictionary within an oscillation based test strategy is described in [49]. Currently, increasing attention is directed to novel mathematical methods like artificial neural networks, e.g. [50], or discrete event systems [51].

The main advantages of the SBT-techniques are their suitability for several levels of description (transistor level, hierarchical), independence with respect to technology and no assumption on the type of system. Their main drawback lies in the large volume of data to be processed, fuzziness in the definition of the tolerances for the fault signatures and the risk of overlooking faults. Concerning soft and multiple faults these drawbacks

are extremely aggravated. Therefore these methods normally address the single hard fault situation only.

SAT-techniques

In the SAT techniques, the responses of the analogue network to the test stimuli are analysed to determine the faulty elements of the circuit. Consequently, the major part of the calculation is accomplished when the test results are available. The SAT techniques are mostly used for soft fault diagnosis. Early publications concentrate on *parameter identification* by solving a non-linear equation set for all network parameters [52]. Its main drawback is on the one hand the numerical complexity. On the other hand, to solve for all circuit parameters one needs sufficient independent measurement results and consequently a large set of test nodes. This requirement can usually not be fulfilled, especially for larger circuits. A modern alternative into this direction has been presented in [28, 53] where computation time is reduced by applying sensitivity computations.

In most cases, however, the number of independent measurements is less than the number of circuit parameters. Then, two main approaches can be found in literature:

1. **Estimation methods:** the faulty element is identified based on an estimation criterion using either deterministic methods [54] or probabilistic techniques [55]. These methods have low demands concerning accessibility of internal circuit nodes, however, they suffer from high computing time and are only adequate for single soft fault diagnosis.

2. **Fault Verification methods:** an upper bound is assumed on the number of simultaneous faults, usually less than the number of measurements performed. The most promising approaches represent the topology of the circuit as a linear graph. In [56] a testability condition has been established which depends on topology only.

However, it is not very practical as its focus is on node-faults, where a faulty node is defined as node to which faulty components are connected. This has been improved in [57] where the faults are defined as usual in terms of faulty components and the testability condition is also based merely on topology information. Topology based testability analysis can be used for time efficient test point selection in an early design stage before the chip has been laid out [58, 59, 60]. A symbolic fault verification technique has been presented in [61]. In [62] the CUT is hierarchically decomposed into subcircuits using measurement nodes. Fault diagnosis is achieved by checking the consistency of Kirchhoff's current law between decomposed subcircuits. This method is adopted for testing chips at the board level where all chip pins are accessible. The approach can be applied only to testing bipolar ICs as the current in the case of CMOS ICs is too small to check current consistency.

To summarize, the SAT techniques are more appropriate for testing discrete analogue circuits than integrated ones [44] due to the number of test points required by most methods. Usually, the time for performing the SAT on-line computations becomes unacceptable for larger circuits. However, within the fault verification techniques, the topological approach is in general the most efficient for large scale CUTs.

Based on this overview of analogue test generation techniques, the decision has been made to use a topological SAT method with fault verification for the development of a parametric fault diagnosis algorithm. The reasons for this decision are:

- a SAT technique is more appropriate than a SBT method for diagnosing parametric deviations.

- a parameter identification technique cannot be applied to integrated circuits due to the high number of test nodes required.

- topological fault verification methods are computationally more efficient than estimation methods and can also be applied to diagnosing multiple simultaneous parametric deviation faults.

To overcome the computing time limitations of the SAT technique and to reduce the number of required test points a hierarchical approach is adopted in Chapter 3. The goal is to develop a parametric fault diagnosis method applicable to large scale analogue integrated circuits within characterisation test.

2.2.3 Design for Testability

During the last decade, mixed-signal chips which integrate both analogue and digital functionality on a common substrate have become very popular. Currently, mixed-signal circuits are widely used in automotive and consumer electronic applications to increase reliability and reduce cost. The incorporation of analogue and digital circuitry in a single mixed-signal ASIC, as illustrated in Figure 2.5, makes the task of testing very difficult. On the one hand, this is caused by the difficulties associated with analogue testing as discussed in the previous sections. On the other hand, there is a lack of *controllability* and *observability* of the embedded circuit modules. The testing task is exacerbated further

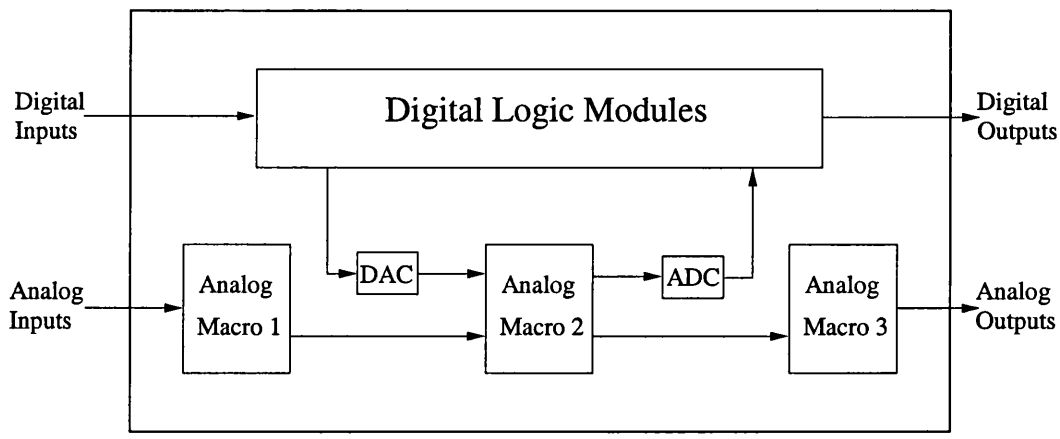


Figure 2.5: Architecture of a mixed-signal IC

by the presence of interface circuit blocks, such as analogue-to-digital (ADC) and digital-to-analogue converters (DAC) and other circuit modules (e.g. switched capacitor (SC) circuits) that exhibit both analogue and digital characteristics.

The accepted test practice leading to good fault isolation for mixed-signal circuits is the divide and conquer approach [63]. This approach partitions a mixed-signal circuit into analogue, digital (memory and logic) blocks so that each block can be tested with its own specific methods. Central to this approach is that the CUT has a test mode to allow direct access to the inputs (controllability) and outputs (observability) of each block via boundary scan and additional analogue test buses.

To isolate the modules in an analogue or mixed-signal IC, and provide access to some of the circuit internal nodes to enable the applications of mode specific tests, a number of analogue DFT techniques have been proposed. Reviews and classifications of these techniques may be found in [2, 64].

Firstly, there is the *analogue access or test point insertion* methodology. A conceptually simple and often used approach [65] is illustrated in Figure 2.6. An analogue multiplexer

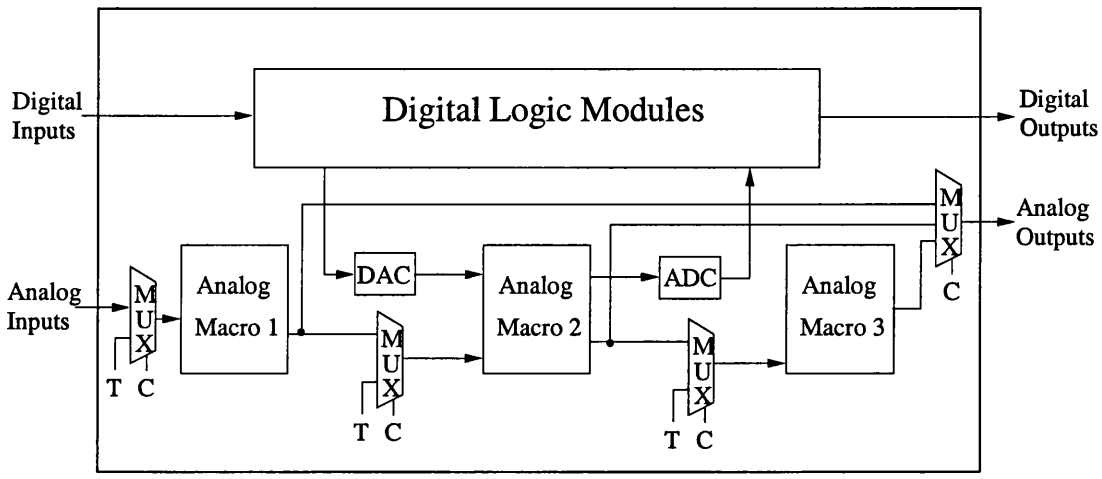


Figure 2.6: Multiplexer-based analogue DFT technique

(MUX) is placed at the input of each analogue macro to give controllability. The output of the macro is observed by using another MUX which is common to the output of all analogue macros. All the analogue test inputs (T) are routed through a demultiplexer which is not shown in the diagram in the interest of clarity. The MUXs are controlled by a control signal (C).

Another approach in the analogue access methodology is the use of an analogue shift register block [66] similar to the scan path in digital circuits. Alternatively to the analogue shift register, a switchable opamp (sw-opamp) concept [67, 68] can be applied. Within this concept, the sw-opamp has two modes that are controlled by a digital signal. Within the normal mode the sw-opamp works as a normal opamp whereas in the test mode the sw-opamp becomes a buffer, where its input passes directly to its output. For circuits composed of blocks of opamps, controllability and observability of an arbitrary block is then achieved by controlling the opamp modes in a suitable way.

Opposite to the analogue access based DFT is the *reconfiguration* methodology which achieves a testability improvement by reconfiguring the CUT with CMOS switches. Approaches into this direction can be found for active filters in [69] and SC filter circuits in [70].

The third methodology is directly parallel to the digital BIST idea. The goal of BIST is to incorporate circuitry to an IC to enable it to carry out some form of self-testing. By this means, the effort of test signal generation, performing measurements and data postprocessing is alleviated and circuit testability enhanced.

The BIST methods can be divided into functional and fault based BIST. Functional BIST techniques are based on the functional test approach where the circuit blocks are tested with functional stimuli signals. Examples of functional BIST can be found in [71, 72]. In [71] the combined performance of the on-chip ADC and DAC are tested by a

FFT (Fast Fourier Transform) analysis on the ADC output signal, with the ADC input derived from the output of the DAC which in turn takes its input stimulus generated from the on-chip DSP (Digital Signal Processor) cores. A functional BIST approach for an ADC using a ramp stimuli derived by reconfiguration of existing functions is described in [73].

Fault based BIST techniques adopt a structured test approach and aim at the fault detection typically using stimuli not related to the specification. One of the earliest schemes is the hybrid built-in-selft-test (HBIST) [74] which is applicable to ICs which combine large digital kernel systems with peripheral analogue subcircuits. It uses BIST of the digital section to scan in the test data for the analogue section, and the DAC to generate a multi-level piece-wise constant signal, from the scanned in data, to be applied to the analogue section. The response of the analogue section is then converted to digital formats by the ADC and scanned out by the digital BIST. The oscillation test method of [49, 75] removes the need for test signal generation by turning the CUT, e.g. opamp or ADC, into an oscillator. A fault is detected based on a shift in the oscillation frequency. In the BIST technique of [76] a DC signal is applied to the CUT and an additional error detection circuit derives an error voltage. A non-zero value of this error voltage is used as an indication for both soft or hard faults. Test signature analysis to accommodate with tolerances in analogue signals is described in [77].

To address the problem of testing mixed-signal ICs at the board level, work on a new mixed signal test bus standard IEEE 1149.4 [78], an extension of the 1149.1 boundary scan standard, has been going on since 1991. However, the interest of the European industry in this standard is rather limited [2].

Partitioning an analogue or mixed-signal IC by applying a DFT strategy has several disadvantages. Firstly, the performance of analogue blocks might be compromised by the

additional circuitry and test nodes, especially in the case of high performance designs and SC circuits. DFT usually increases product cost by the silicon area needed for the test circuitry and the additional test pins. For example, the analogue shift register approach requires two opamps, two switches and a capacitor for each test point. Such an increase in area and power consumption is usually prohibitive. Therefore, to make DFT considerations in the analogue domain for the designer acceptable, it is strongly required to find a way to keep the DFT overhead as low as possible. Despite the strong need for a structured DFT, there is currently no DFT standard and CAD support available which helps the designer in finding an optimal set of test points.

2.2.4 Conclusion

Analogue testing and test preparation is still in its infancy. As opposed to the digital domain, there is still a lack of software tools supporting DFT and test generation. This is mainly due to the diversity of analogue behaviour and the increasing complexity of large scale analogue systems which makes circuit simulation slow. As IC component structures are scaled down parametric deviations gain importance during characterisation and testing. However, due to the low accessibility to circuit nodes of analogue ICs it is difficult and sometimes even impossible to diagnose these parametric deviations. Parametric problems which are not diagnosed during characterisation and removed before the chip is passed on to production, usually complicate production test thereby increasing cost and decreasing quality.

In the next chapter, a method for diagnosis of parametric deviations in large scale analogue integrated circuits is investigated. The diagnosis method is based on a topological SAT technique with fault verification according to the approach of Wey [57]. The merits of Wey's contribution is structured DFT which integrates circuit diagnosibility into

the early design stage before the chip has been laid out. This has been achieved based on a testability condition which merely depends on circuit topology and guarantees diagnosibility, also for multiple deviation faults, with a minimal set of test points and no additional on-chip circuitry. When the first silicon is available, the test points can be used to characterise the device and, in the event of problems, allow for a diagnosis of the components with parametric deviations. This strongly alleviates the elimination of design problems thereby reducing time to market and improving quality of the product. As the approach is algorithmic based, an automatic DFT tool support becomes possible which complements the other hardware-based DFT techniques.

Despite all the advantages of Wey's approach, its application to large scale circuits is impractical due to the inherent computational complexity. Additionally, fault diagnosis of SC circuits using Wey's method is not possible because the underlying circuit description doesn't fit the time discrete character of this type of circuit. To overcome these handicaps and make the algorithm usable for today's large scale analogue networks, a hierarchical approach is presented in the next chapter. As a by-product of the higher level of abstraction, the number of test points required is reduced which makes the method applicable to integrated circuits.

Chapter 3

Modelling Aspects of Analogue Parametric Fault Diagnosis

The purpose of this chapter is to present an approach to diagnosis of parametric deviations in analogue circuits. Based on the Fault Diagnosis Algorithm (FDA) introduced in [57, 58] a hierarchical approach is adopted which results in the development of a Hierarchical Fault Diagnosis Algorithm (HFDA) capable of performing parametric fault diagnosis on large scale ICs.

The development of the HFDA splits into two tasks: Firstly, a modification of the previous FDA is required such that diagnosis of faults in a circuit built out of hierarchical components (such as opamps) becomes possible. Secondly, a hierarchical circuit modelling strategy needs to be developed for the description of the parametric behaviour of hierarchical components. Herein, the hierarchical circuit modelling must fit the circuit description used within the FDA.

In relation to this task splitting, the HFDA development was organized as a collaboration between the University of Bath and the Robert Bosch company. The first task has been treated at the University of Bath and the results of this work have been reported

in [79]. The second task rested with the author of this thesis as a member of Bosch and the work on this topic is reported in this chapter. Publications of the results of the collaboration can be found in [80, 81].

The layout of the chapter is as follows. Firstly, the FDA of [57, 58] and the underlying circuit description are presented. The limitations of this approach concerning its applicability to large scale ICs and SC circuits is explained. Then, the hierarchical modelling strategy used within the HFDA is introduced. Emphasis is drawn to the implications of the hierarchical modelling on the algorithmic aspects of the HFDA to guarantee maximal performance in the case of large scale ICs. The HFDA is extended such that the fault diagnosis of SC circuits becomes possible. Experimental results are presented to benchmark the HFDA with respect to its efficiency, applicability and limitations.

3.1 Parametric Fault Diagnosis

In this section, the approach of [57, 58] to diagnosis of parametric deviation faults is presented and the component connection model as the underlying circuit description is introduced.

3.1.1 Component Connection Model (CCM)

The topology of an electrical CUT may be expressed as a directed linear graph consisting of edges representing the circuit components and connecting the circuit nodes [82]. Each edge represents a voltage and current quantity. Arrows are associated with the edges to define the direction of current flow through and voltage drop across them. A tree of a graph is a subset of edges connecting all nodes without completing any closed loop. The respective cotree is then defined as the complement of this tree in respect of the edges. Once a tree of the circuit graph is specified, the Component Connection Model (CCM)

[83] separates the CUT model into component behaviour and topology description. The component behaviour is modelled by a matrix *component equation*

$$\mathbf{b} = \mathbf{Z}\mathbf{a}, \tag{3.1}$$

$$\text{where} \quad \mathbf{a} = \begin{pmatrix} \mathbf{i}_{trec} \\ \mathbf{v}_{cotree} \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} \mathbf{v}_{tree} \\ \mathbf{i}_{cotree} \end{pmatrix} \tag{3.2}$$

are the input and output vectors respectively. The elements of the vectors $\mathbf{i_{tree}}$ ($\mathbf{i_{cotree}}$) and $\mathbf{v_{tree}}$ ($\mathbf{v_{cotree}}$) are the currents through and voltages across the tree (cotree) edges. The component transfer matrix $\mathbf{Z}$ describes the linear voltage-current relation of the CUT components with help of the component admittances or impedances. Extensions of the CCM for non-linear components are considered in [84, 85]. The topology of the circuit is represented by the *connection equation*

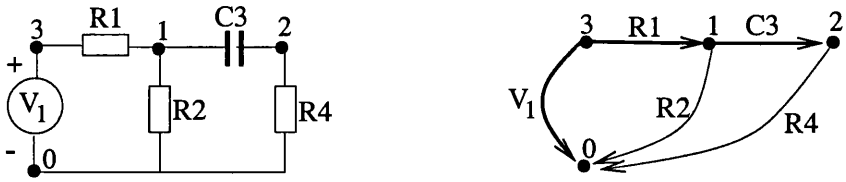
$$\mathbf{a} = \mathbf{L_{11}}\mathbf{b} + \mathbf{L_{12}}\mathbf{u}, \qquad \mathbf{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_{n_u} \end{pmatrix} \tag{3.3}$$

which links the input and output vector with the stimulus vector $\mathbf{u}$ containing the n_u stimuli quantities. From the circuit theory point of view, the connection equation comprises the Kirchhoff's Current Law and Kirchhoff's Voltage Law. The measurement results obtained by testing the CUT are described by the *measurement equation*

$$\mathbf{y} = \mathbf{L_{21}}\mathbf{b} + \mathbf{L_{22}}\mathbf{u} \qquad \text{where} \qquad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_{n_y} \end{pmatrix} \tag{3.4}$$

is the test point vector containing the n_y measurement results. The connection matrices $\mathbf{L_{ij}}$ are derived from the fundamental matrix $\mathbf{D} = \mathbf{A}_T^{-1}\mathbf{A}_{CT}$, where $\mathbf{A}_T$ and $\mathbf{A}_{CT}$ are the node incidence matrices referring to the tree and cotree edges respectively [79, 83].

The CUT shown in Figure 3.1a is considered to illustrate the circuit description within the CCM. In Figure 3.1b the corresponding circuit graph is illustrated. Choosing the



a) circuit under test b) corresponding graph representation

Figure 3.1: CUT and corresponding graph representation [79]

edges V_1, R_1 and C_3 as the tree (bold edges in Figure 3.1b) yields

$$\mathbf{a} = \begin{pmatrix} i_{R_1} \\ i_{C_3} \\ v_{R_2} \\ v_{R_4} \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} v_{R_1} \\ v_{C_3} \\ i_{R_2} \\ i_{R_4} \end{pmatrix} \quad \text{and} \quad \mathbf{u} = (v_{V_1}) \tag{3.5}$$

where i_e and v_e denote the current through and voltage across the edge e. The current i_{V_1} and voltage v_{V_1} are omitted in the **a** and **b** vector because V_1 is considered as stimulus rather than as CUT component. The component equation (3.1) becomes

$$\begin{pmatrix} v_{R_1} \\ v_{C_3} \\ i_{R_2} \\ i_{R_4} \end{pmatrix} = \begin{pmatrix} R_1 & 0 & 0 & 0 \\ 0 & \frac{1}{j\omega C_3} & 0 & 0 \\ 0 & 0 & \frac{1}{R_2} & 0 \\ 0 & 0 & 0 & \frac{1}{R_4} \end{pmatrix} \begin{pmatrix} i_{R_1} \\ i_{C_3} \\ v_{R_2} \\ v_{R_4} \end{pmatrix}. \tag{3.6}$$

The connection equation (3.3) is

$$\begin{pmatrix} i_{R_1} \\ i_{C_3} \\ v_{R_2} \\ v_{R_4} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_{R_1} \\ v_{C_3} \\ i_{R_2} \\ i_{R_4} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} v_{V_1}. \tag{3.7}$$

Now it is supposed that during testing the current i_{R1} and the voltage v_{R_2} are measured. This results in

$$\mathbf{y} = \begin{pmatrix} i_{R1} \\ v_{R_2} \end{pmatrix} \tag{3.8}$$

and the measurement equation (3.4) becomes

$$\begin{pmatrix} i_{R1} \\ v_{R_2} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ -1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_{R_1} \\ v_{C_3} \\ i_{R_2} \\ i_{R_4} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} v_{V_1}. \tag{3.9}$$

3.1.2 Fault Diagnosis Algorithm

The Fault Diagnosis Algorithm (FDA) of [58] splits into two phases: the **CCM set-up** phase and the **fault diagnosis** phase.

CCM Set-Up

In the CCM set-up phase the CCM equations (3.1) – (3.4) are derived. Since the connection matrices $\mathbf{L_{ij}}$ depend on the actual circuit tree and since the number of different trees increases rapidly with circuit size, there are many possibilities for the actual structure of the CCM equations. In [58] an optimal tree generation procedure is presented, which heuristically derives a circuit tree which guarantees maximal sparse connection matrices $\mathbf{L_{ij}}$. This reduces the computational expense of the CCM analysis in the fault diagnosis phase of the FDA. Once the optimal tree has been generated, suitable *test points* (elements of the $\mathbf{y}$-vector) are selected to ensure testability [58] with respect to parametric deviation faults. Based on the knowledge of the circuit tree and the test point vector $\mathbf{y}$, the CCM equations (3.1) – (3.4) can easily be generated.

Fault Diagnosis

In the second phase of the FDA, fault diagnosis is achieved based on the *Self Test* (ST) algorithm [57, 86]. The ST algorithm divides circuit components into tester and testee groups. Consequently, the input and output vectors are split into tester (superscript 1) and testee (superscript 2) elements respectively:

$$\mathbf{a} = \begin{pmatrix} \mathbf{a^1} \\ \mathbf{a^2} \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} \mathbf{b^1} \\ \mathbf{b^2} \end{pmatrix}. \tag{3.10}$$

In the first place, tester elements are assumed to be fault free. Good testees are identified within a test cycle as illustrated in Figure 3.2. The first step of the test cycle is based on the Pseudo Circuit description. To derive this description, the CCM equations (3.1), (3.3)

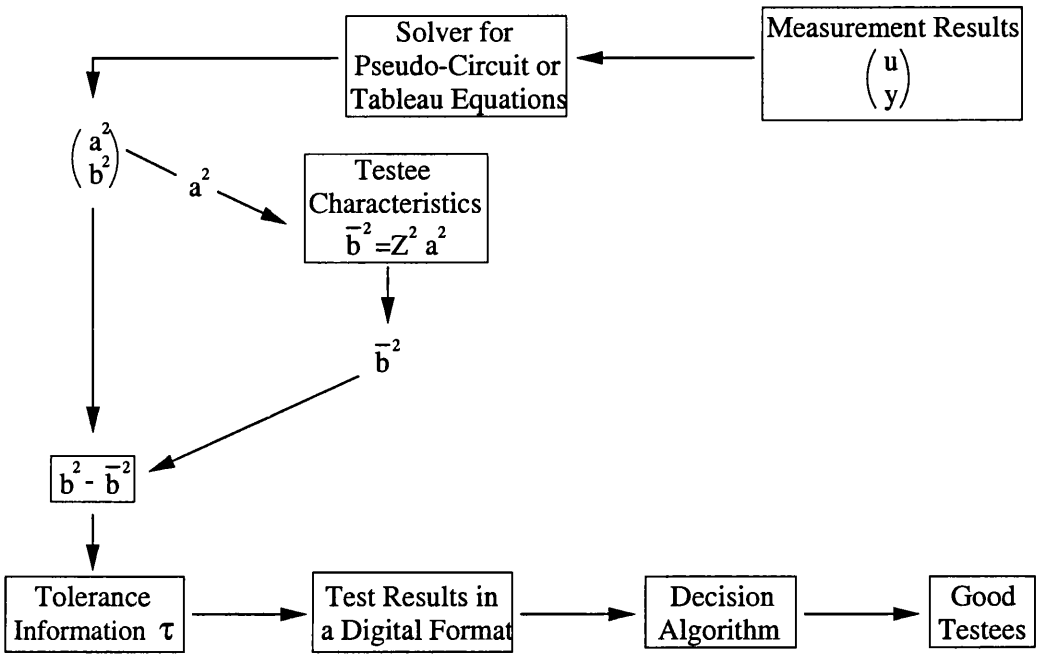


Figure 3.2: Test cycle of the Self Test Algorithm

and (3.4) are written in terms of tester and testee quantities

$$\begin{pmatrix} \mathbf{b}^1 \\ \mathbf{b}^2 \end{pmatrix} = \begin{pmatrix} \mathbf{Z}^1 & 0 \\ 0 & \mathbf{Z}^2 \end{pmatrix} \begin{pmatrix} \mathbf{a}^1 \\ \mathbf{a}^2 \end{pmatrix}, \tag{3.11}$$

$$\begin{pmatrix} \mathbf{a}^1 \\ \mathbf{a}^2 \end{pmatrix} = \begin{pmatrix} \mathbf{L}_{11}^{11} & \mathbf{L}_{11}^{12} \\ \mathbf{L}_{11}^{21} & \mathbf{L}_{11}^{22} \end{pmatrix} \begin{pmatrix} \mathbf{b}^1 \\ \mathbf{b}^2 \end{pmatrix} + \begin{pmatrix} \mathbf{L}_{12}^{1} \\ \mathbf{L}_{12}^{2} \end{pmatrix} \mathbf{u}, \tag{3.12}$$

$$\mathbf{y} = \begin{pmatrix} \mathbf{L}_{21}^{1} & \mathbf{L}_{21}^{2} \end{pmatrix} \begin{pmatrix} \mathbf{b}^1 \\ \mathbf{b}^2 \end{pmatrix} + \mathbf{L}_{22}\mathbf{u}, \tag{3.13}$$

where the matrices $\mathbf{L}_{ij}^{kl}$ and $\mathbf{L}_{ij}^{k}$ are obtained by appropriately picking up rows and columns of the connection matrices $\mathbf{L}_{ij}$. Solving the above equations for the testee quantities yields the *Pseudo Circuit* equation [57]:

$$\begin{pmatrix} \mathbf{a}^1 \\ \mathbf{y}^p \end{pmatrix} = \begin{pmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{b}^1 \\ \mathbf{u}^p \end{pmatrix}, \tag{3.14}$$

$$\text{where} \quad \mathbf{y}^p = \begin{pmatrix} \mathbf{a}^2 \\ \mathbf{b}^2 \end{pmatrix}, \qquad \mathbf{u}^p = \begin{pmatrix} \mathbf{u} \\ \mathbf{y} \end{pmatrix}, \tag{3.15}$$

$$\mathbf{K}_{11} = \mathbf{L}_{11}^{11} - \mathbf{L}_{11}^{12}(\mathbf{L}_{21}^{2})^{-1}\mathbf{L}_{21}^{1}, \tag{3.16}$$

$$\mathbf{K}_{12} = \begin{pmatrix} \mathbf{L}_{12}^{1} - \mathbf{L}_{11}^{12}(\mathbf{L}_{21}^{2})^{-1}\mathbf{L}_{22} & \mathbf{L}_{11}^{12}(\mathbf{L}_{21}^{2})^{-1} \end{pmatrix}, \tag{3.17}$$

$$\mathbf{K}_{21} = \begin{pmatrix} \mathbf{L}_{11}^{21} - \mathbf{L}_{11}^{22}(\mathbf{L}_{21}^{2})^{-1}\mathbf{L}_{21}^{1} \\ (\mathbf{L}_{21}^{2})^{-1}\mathbf{L}_{21}^{1} \end{pmatrix}, \tag{3.18}$$

$$\mathbf{K}_{22} = \begin{pmatrix} \mathbf{L}_{12}^{2} - \mathbf{L}_{11}^{22}(\mathbf{L}_{21}^{2})^{-1}\mathbf{L}_{22} & \mathbf{L}_{11}^{22}(\mathbf{L}_{21}^{2})^{-1} \\ (\mathbf{L}_{21}^{2})^{-1}\mathbf{L}_{22} & (\mathbf{L}_{21}^{2})^{-1} \end{pmatrix}. \tag{3.19}$$

The Pseudo Circuit equation[1] is used to solve for the testee quantities $\mathbf{a}^2$ and $\mathbf{b}^2$ based on the knowledge of the test stimuli values $\mathbf{u}$ and the measurement results $\mathbf{y}$. Whether a testee is fault-free or not depends on the question whether the testee quantities $\mathbf{a}^2$ and $\mathbf{b}^2$ obtained from measurement, i.e. obtained from the Pseudo Circuit equation, are in accordance with the expected testee component behaviour described by $\mathbf{Z}^2$. To answer this question, the testee component equation is used to calculate $\overline{\mathbf{b}}^2 = \mathbf{Z}^2\mathbf{a}^2$ (see Figure 3.2). For ideal fault-free testee characteristic, the relation $\overline{\mathbf{b}}^2 = \mathbf{b}^2$ should be fulfilled. However, due to the continuous character of analogue signals and the tolerances associated with the parameters of all manufactured components the difference $b^2 - \bar{b}^2$ needs not vanish completely for the testee to be actually fault-free. A tolerance τ_i is associated with the parameter of each testee element $\#i$. This tolerance defines the area of acceptability around the nominal value, in which the component is considered fault-free:

$$\begin{aligned} |b_i^2 - \bar{b}_i^2| \leq \tau_i \quad &\Rightarrow \quad \text{testee \#i fault-free,} \\ |b_i^2 - \bar{b}_i^2| > \tau_i \quad &\Rightarrow \quad \text{testee \#i faulty.} \end{aligned} \tag{3.20}$$

The tolerances need to be defined in respect of design requirements on the one hand and real component variations originating from process statistics on the other hand. The fault-free/faulty decision (3.20) provides test results in a digital format for each testee and allows for an identification of fault-free testees based on a decision algorithm [57, 86]. Any component which is determined to be fault-free is moved into the tester group resulting in a re-partitioned circuit graph for the next test cycle according to Figure 3.2. This

[1] The test points have been selected in [58] such that the Pseudo Circuit equation exists in most cases, i.e. $\mathbf{L}_{21}^2$ is invertible. However, when $\mathbf{L}_{21}^2$ is not invertible, it still may be possible to solve for $\mathbf{a}^2$ and $\mathbf{b}^2$ with help of the so-called Tableau Equations [57].

process is repeated until all the testers are established to be fault-free, at which point the test results from the actual test cycle are completely reliable and the diagnosis of faulty components is achieved.

3.1.3 Conclusions

Implementations of the FDA as described above were published in [58, 87, 88]. The FDA can detect and locate parametric deviation faults (soft faults). Since the testability conditions are based on the invertability of the connection matrix $\mathbf{L}_{21}^{2}$ [57], automatic test point selection can be implemented merely using circuit topology information. The FDA approach allows therefore for DFT considerations in an early design stage where the component parameters have not been designed yet. Since the FDA is programmable, a DFT tool support can be implemented. Hard faults cannot be diagnosed because this fault classification alters the circuit topology and the FDA relies on constant topology for all possible fault situations (connection equation remains unchanged during the ST algorithm). Besides this situation, two necessary conditions for the industrial application of the FDA are not met:

- ability to diagnose deviation faults in large scale analogue ICs,
- ability to diagnose deviation faults in SC circuits.

Within the FDA described so far, analysis is done at the transistor level of circuit description. The analysis is based on a matrix approach and, as the size of the circuit increases, so does the size of the matrices. Additionally, the number of test cycles applied by the ST algorithm before reliable diagnosis is achieved increases significantly with the number of circuit components. Altogether, computing time and storage requirements depend severely on network size making transistor level testing with the FDA impractical for larger analogue circuits.

A further consequence of the transistor level analysis is that the FDA needs a large number of test points. As a result, the FDA in the presented version can not be used for diagnosing integrated circuits.

The second limitation is related to the fact that the CCM is based on pairs of voltage/current quantities described in the s-domain. This sort of circuit description is not suited for SC circuits.

The next two sections present techniques which overcome these limitations and allow for fault diagnosis of large scale ICs and SC circuits. The approaches adopted are

- hierarchical fault diagnosis
- voltage/charge based z-domain circuit description

Emphasis will be drawn to the modelling aspects and the inclusion of the CUT models within the CCM. Consequences of the hierarchical approach for the algorithmic aspects of the FDA, e.g. test point selection, have been treated in [79].

3.2 Hierarchical Parametric Fault Diagnosis

Network size limitations of circuit analysis procedures can be tackled by adopting a hierarchical approach. Examples for hierarchical methods are the symbolic analysis technique in [89] or the behavioural modelling concepts of [90, 91, 92]. To make the FDA applicable to large scale circuits, the CCM needs to be extended in a way that hierarchical circuit analysis becomes possible. This results in the development of a Hierarchical Fault Diagnosis Algorithm (HFDA).

A hierarchical description organizes blocks of the circuit into multi-terminal components. Modelling of the hierarchical components is achieved by suppressing the voltage and current quantities of nodes and edges internal of the component and merely describing the

electrical behaviour seen at the terminals from the outside world. Within a hierarchical CCM, the component equation of a hierarchical component has the structure

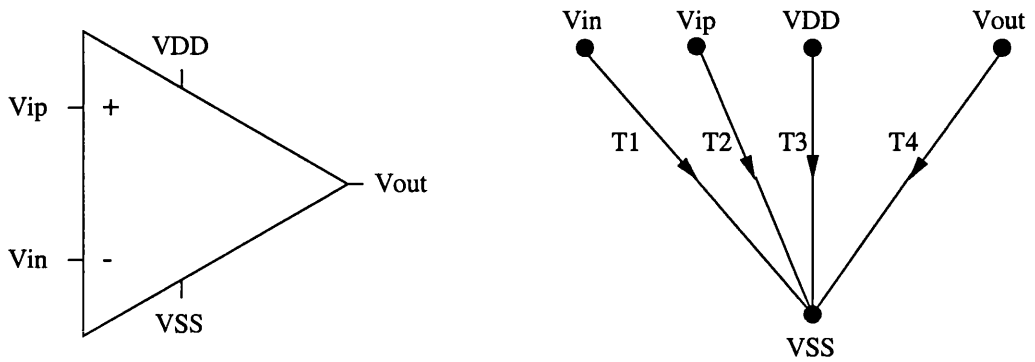
$$\mathbf{b}_{\mathbf{hier}} = \mathbf{Z}_{\mathbf{hier}}\mathbf{a}_{\mathbf{hier}}, \tag{3.21}$$

where the input- and output vectors $\mathbf{a}_{\mathbf{hier}}$ and $\mathbf{b}_{\mathbf{hier}}$ contain only the terminal voltages and currents of the hierarchical component. By using a hierarchical CCM description, the sizes of the component transfer matrix $\mathbf{Z}$ and the connection matrices $\mathbf{L}_{\mathbf{ij}}$ are reduced allowing the fault diagnosis of large scale analogue circuits.

Hierarchical components can be considered at different complexity levels. For example, inverters and current mirrors may be considered at low level, differential and output stages at medium level, and opamps and comparators at a higher level. Typical analogue functions like filters or integrators may be regarded for system level analysis. Generally, the higher the hierarchical description level, the faster analysis can be performed. The cost for this improvement is diagnosis resolution. HFDA can perform only go/no-go testing on the hierarchical blocks because the internal block behaviour is hidden by the hierarchical description. If a deviation fault has to be located within a circuit block, an individual test of the respective block needs to be performed at a lower hierarchical level.

3.2.1 Hierarchical Model Structure

In relation to the mathematical implementation of the CCM, the description of hierarchical models is based on the graph representation. A hierarchical component has a multi-edge graph representation. An example for an single-ended opamp is shown in Figure 3.3. In the corresponding graph representation (Figure 3.3b), each terminal node is connected by an edge to the common reference node (VSS). The graph edges represent pairs of voltage and current quantities which are specific to the input and output relation of the hierarchical component. According to equation (3.21), these voltages and currents are described by a



a) operational amplifier b) hierarchical graph representation

Figure 3.3: Opamp and its hierarchical graph representation

hierarchical component matrix equation which has the following structure in the case of the opamp

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}_{op} = \begin{pmatrix} Z_{11} & Z_{12} & Z_{13} & Z_{14} \\ Z_{21} & Z_{22} & Z_{23} & Z_{24} \\ Z_{31} & Z_{32} & Z_{33} & Z_{34} \\ Z_{41} & Z_{42} & Z_{43} & Z_{44} \end{pmatrix}_{op} \cdot \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix}_{op} . \tag{3.22}$$

By skipping internal nodes and edges a significant reduction of the number of edges in the hierarchical graph representation compared to a transistor level description has been achieved, resulting in a considerably reduced 4×4 matrix. Picking one of the possible assignments of the terminal voltages and currents to the **a**- and **b**-vector elements, equation (3.22) becomes

$$\begin{pmatrix} i_{in} \\ i_{ip} \\ i_{DD} \\ v_{out} \end{pmatrix}_{op} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ g_{V_{in},V_{DD}} & g_{V_{ip},V_{DD}} & g_{V_{DD}} & h_{V_{out},V_{DD}} \\ -A + \frac{A}{2CMRR} & A + \frac{A}{2CMRR} & \frac{A}{PSRR} & R_{\text{out}} \end{pmatrix}_{op} \cdot \begin{pmatrix} v_{in} \\ v_{ip} \\ v_{DD} \\ i_{out} \end{pmatrix}_{op} , \tag{3.23}$$

where $g_{V_{in},V_{DD}}$, $g_{V_{ip},V_{DD}}$ and $g_{V_{DD}}$ are (trans)conductances between the subscripted pins, $h_{V_{out},V_{DD}}$ is a current transfer function between the subscripted pins, A is the amplification, $CMRR$ is the common mode rejection ratio, $PSRR$ is the power supply rejection ratio and R_{out} is the output resistance. $PSRR$ is defined only with respect to VDD, since VSS is the reference node. For simplicity the input admittances of the opamp are set to zero (upper two zero rows in the matrix of equation 3.23) which is valid in CMOS technology

in the low frequency region. The representation in (3.22) should not be confused with a impedance matrix, as currents and voltages may be mixed on both sides of the equation.

3.2.2 Hierarchical Model Characterisation

In the last section, the behaviour of a hierarchical component was described with the help of a hierarchical component transfer matrix $\mathbf{Z_{hier}}$. The structure (number of columns and rows) of $\mathbf{Z_{hier}}$ is purely dictated by the number of terminals of the hierarchical component. The entries of $\mathbf{Z_{hier}}$ are functions in respect of process characteristics, the actual schematic realization and the frequency of operation. For example, the opamp amplification A depends on the dominant and second pole which determine the gain and phase margins.

The actual values of the entries of $\mathbf{Z_{hier}}$ need to be determined by characterisation before the models can be used within the FDA. Characterisation is accomplished by simulating each hierarchical component separately and afterwards, extracting the respective electrical behaviour, i.e. transfer function, conductances, transconductances, $PSRR$, $CMRR$, etc. Characterisation can be done manually, or with help of a tool which supports automatic model characterisation, e.g. [93]. In the case where only linear circuit behaviour is requested, symbolic simulation can be used for automatic model generation and characterisation (see Chapter 4).

Besides the nominal values of the $\mathbf{Z_{hier}}$ entries, the respective tolerances τ_i are required by HFDA to decide whether the testees of a test cycle are fault-free or faulty (Figure 3.2, equation 3.20). The HFDA can be applied to deviation fault diagnosis with respect to two different test concepts: *specification testing* or testing whether all circuit elements have been fabricated correctly (*structural testing*). In the former case, the functional specification of the circuit is tested. The designer has to decide which tolerances of a hierarchical block are in accordance to the circuit specification. These "block specification

tolerances" are then used as τ_i-values in equation 3.20. In the latter case, it should be tested whether devices at the lowest hierarchical level have been fabricated correctly, i.e. whether the parameter values of the devices which constitute the hierarchical components are in tolerance. Then, the tolerances τ_i of hierarchical blocks need to be determined by tolerance analysis (see Chapter 6) or by behavioural tolerance modelling [41].

3.2.3 Consequences of the Hierarchical Approach for the CCM Circuit Description

The CCM is based on the circuit graph representation and requires the generation of a tree within this graph. In [58] an Optimal Tree Generation (OTG) procedure has been presented which leads to the sparsest CCM matrix sets. This yields large savings in computing time when the FDA is applied to practical circuits for fault diagnosis. However, the OTG in [58] is restricted to a flat circuit graph at the lowest abstraction level. The hierarchical approach introduced in the previous sections poses some restrictions on the tree generation procedure. These restrictions and the implications of the hierarchical approach for the CCM are now discussed.

A hierarchical component is characterised by specific pairs of voltage and current which are abstracted as star-connected edges (see Figure 3.3). When a circuit block of a CUT is treated as a hierarchical component, the edge currents and voltages of the hierarchical component must be part of the $\mathbf{a}$ and $\mathbf{b}$ vectors which are partitioned into tree and cotree edge parts. Whether one of the edges is on a tree or on a cotree is obvious from the type of physical quantity (voltage or current) represented by each element in its corresponding $\mathbf{a_{hier}}$ or $\mathbf{b_{hier}}$ vector. The following rule on partitioning the constituent edges of a hierarchical component into tree and cotree edges results from examining the associated vector $\mathbf{b_{hier}}$ according to equation (3.2): if an element of the $\mathbf{b_{hier}}$ vector is a

current or voltage quantity, its corresponding edge will be on the cotree or tree respectively.

Concerning the opamp example of Figure 3.3, the above rule and equations (3.22) and (3.23) suggest the edges $T1$, $T2$ and $T3$ to be cotree edges, $T4$ to be a tree edge. This primarily suggested tree/cotree-partition is called root partition. Now, the following questions arise: are other tree/cotree-partitions also possible, which tree/cotree-partitions are possible and how does the modelling equation $\mathbf{b_{hier}} = \mathbf{Z_{hier}a_{hier}}$ of a hierarchical component transform when the partitions are changed?

Generally, a new tree/cotree-partition is obtained by starting from the root partition and exchanging tree and cotree edges. According to equation (3.2), this is equivalent to an exchange of **a**-vector elements with their respective **b**-vector elements. Supposing that there are n edges and that the edges to be exchanged are $T_1, \cdots, T_k$, $(k \leq n)$, the old modelling equation of an hierarchical component with respect to the root partition splits into:

$$\begin{pmatrix} \mathbf{b_1^{old}} \\ \mathbf{b_2^{old}} \end{pmatrix} = \begin{pmatrix} \mathbf{Z_{11}^{old}} & \mathbf{Z_{12}^{old}} \\ \mathbf{Z_{21}^{old}} & \mathbf{Z_{22}^{old}} \end{pmatrix} \begin{pmatrix} \mathbf{a_1^{old}} \\ \mathbf{a_2^{old}} \end{pmatrix}, \tag{3.24}$$

where

$$\mathbf{a_1^{old}} = \begin{pmatrix} a_1 \\ \vdots \\ a_k \end{pmatrix}, \quad \mathbf{a_2^{old}} = \begin{pmatrix} a_{k+1} \\ \vdots \\ a_n \end{pmatrix},$$

$$\mathbf{b_1^{old}} = \begin{pmatrix} b_1 \\ \vdots \\ b_k \end{pmatrix}, \quad \mathbf{b_2^{old}} = \begin{pmatrix} b_{k+1} \\ \vdots \\ b_n \end{pmatrix}. \tag{3.25}$$

After the exchange of $T_1, \cdots, T_k$, the new **a** and **b** vectors are

$$\begin{pmatrix} \mathbf{a_1^{new}} \\ \mathbf{a_2^{new}} \end{pmatrix} = \begin{pmatrix} \mathbf{b_1^{old}} \\ \mathbf{a_2^{old}} \end{pmatrix}, \quad \begin{pmatrix} \mathbf{b_1^{new}} \\ \mathbf{b_2^{new}} \end{pmatrix} = \begin{pmatrix} \mathbf{a_1^{old}} \\ \mathbf{b_2^{old}} \end{pmatrix}, \tag{3.26}$$

Applying some algebraic manipulation yields the new component transfer equation

$$\begin{pmatrix} \mathbf{b_1^{new}} \\ \mathbf{b_2^{new}} \end{pmatrix} = \begin{pmatrix} \mathbf{Z_{11}^{new}} & \mathbf{Z_{12}^{new}} \\ \mathbf{Z_{21}^{new}} & \mathbf{Z_{22}^{new}} \end{pmatrix} \begin{pmatrix} \mathbf{a_1^{new}} \\ \mathbf{a_2^{new}} \end{pmatrix}, \tag{3.27}$$

where

$$\mathbf{Z_{11}^{new}} = (\mathbf{Z_{11}^{old}})^{-1},$$

$$\begin{aligned}
\mathbf{Z}_{12}^{\text{new}} &= (\mathbf{Z}_{11}^{\text{old}})^{-1}\mathbf{Z}_{12}^{\text{old}}, \\
\mathbf{Z}_{21}^{\text{new}} &= \mathbf{Z}_{21}^{\text{old}}(\mathbf{Z}_{11}^{\text{old}})^{-1}, \\
\mathbf{Z}_{22}^{\text{new}} &= \mathbf{Z}_{22}^{\text{old}} - \mathbf{Z}_{21}^{\text{old}}(\mathbf{Z}_{11}^{\text{old}})^{-1}\mathbf{Z}_{12}^{\text{old}}.
\end{aligned} \tag{3.28}$$

Consequently, the condition for the existence of the new tree/cotree-partition is that the block matrix $\mathbf{Z}_{11}^{\text{old}}$ is invertible [80]. For example, the edges $T1$ and $T2$ of the opamp in Figure 3.3 must not be tree-edges of the circuit graph. Otherwise a zero row would occur in the respective $\mathbf{Z}_{11}^{\text{old}}$ (see equation 3.23) and the transformation in (3.28) would not be valid. With this condition in mind and the knowledge of the component transfer matrix $\mathbf{Z}_{\text{hier}}$, a table with all possible tree/cotree-partitions of a hierarchical component can be established. For the opamp model of (3.23) this partition table is shown in Table 3.1. The partition tables of all hierarchical components need to be taken into account when

Elements to be exchanged	Tree	Co-tree	condition
-	T_1, T_2, T_3	T_4	none (root partition)
$i_{DD} \leftrightarrow v_{DD}$	T_1, T_2	T_3, T_4	$g_{V_{DD}} \neq 0$
$i_{out} \leftrightarrow v_{out}$	T_1, T_2, T_3, T_4	-	$R_{out} \neq 0$
$i_{DD} \leftrightarrow v_{DD}, i_{out} \leftrightarrow v_{out}$	T_1, T_2, T_4	T_3	$\det \begin{pmatrix} g_{V_{DD}} & h_{V_{out},V_{DD}} \\ \frac{A}{PSRR} & R_{out} \end{pmatrix} \neq 0$

Table 3.1: Possible tree/cotree partitions for the opamp

generating a tree for the whole circuit. The development of a *Hierarchical* Optimal Tree Generation (HOTG) procedure which relies on the partition rules derived in this section has been presented in [80]. The HOTG selects an optimal tree for a circuit with interconnected hierarchical components leading to effective fault diagnosis of large scale analogue circuits.

The hierarchical description of circuit components introduced above has also consequences for the test point selection strategy and the tester/testee-repartitioning after each

test cycle (see Figure 3.2). These consequences are discussed in [79].

3.3 Parametric Fault Diagnosis of Switched Capacitor (SC) Circuits

SC circuits are very popular in signal processing applications due to their high precision performance. At Bosch Microelectronics, SC circuits are widely in use within automotive ASICs. Unfortunately, the HFDA presented so far is not able to handle this type of circuit. The underlying reason is that the CCM is based on a current flow description whereas SC circuits are characterised by charge exchange between capacitors. Additionally, SC circuits work in a time discrete manner whereas the CCM describes time continuous systems. This section presents CCM extensions which allow time discrete system analysis and make the HFDA applicable to fault diagnosis of SC circuits.

3.3.1 CCM Graph Representation for SC Circuits

There exist many concepts for analysis of SC circuits, e.g. [94, 95, 96]. These methods are based on a charge exchange description in the discrete-time- or z-domain rather than on time continuous current flow. The goal of this section is to define pairs of voltage and charge quantities (instead of voltage current pairs) and a respective graph description in such a way that the CCM and the HFDA developed in the previous sections can also be applied to SC circuits.

In the CCM of [83] (see Section 3.1.1), the edge currents obey Kirchhoff's Current Law (KCL):

$$\text{time-continuous system:} \qquad \sum_{\text{edges } e \text{ incident}}^{\text{at node } n} i(e) = 0, \quad \forall \text{ nodes } n, \tag{3.29}$$

where $i(e)$ denotes the time continuous current through the circuit edge e.

For a SC circuit, the equivalence to this relation is that the charge flow into a circuit node n which occurs during the switching event from a previous equilibrium state "before" to the actual equilibrium state "now" is zero. This situation is illustrated in Figure 3.4. When the switch S is closed, charge flows from one capacitor to the other one. The amount

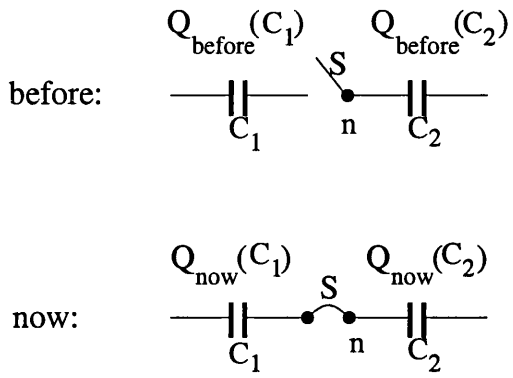


Figure 3.4: Charge conservation at a SC circuit node

of charge flowing out of each capacitor during the transition from "before" to "now" is:

$$\begin{aligned} \Delta Q(C_1) &= Q_{now}(C_1) - Q_{before}(C_1), \\ \Delta Q(C_2) &= Q_{now}(C_2) - Q_{before}(C_2), \end{aligned} \tag{3.30}$$

and charge conservation forces

$$\text{SC circuit:} \quad \sum_{i=1}^{2} \Delta Q(C_i) = 0. \tag{3.31}$$

The switches of a SC circuit are controlled by periodic clock signals. The period time refers to one clock cycle, and the clock cycles divide into clock phases. In the following it is supposed that there are K non-overlapping clock phases within one clock cycle. As a result, there are K different equilibrium states of the circuit within a specific clock cycle. The clock phases are denoted by the subscript k, $k = 1 \ldots K$, and the clock cycles by superscript m. During each of the K transitions from one clock phase equilibrium state k to the next state $k+1$, there occurs a charge flow out of each circuit element and its

representing graph edge e. According to equation set (3.30), these edge charge flows are defined as

$$\begin{aligned}
\Delta Q_1^m(e) &:= Q_1^m(e) - Q_K^{m-1}(e), \\
\Delta Q_2^m(e) &:= Q_2^m(e) - Q_1^m(e), \\
\Delta Q_3^m(e) &:= Q_3^m(e) - Q_2^m(e), \\
&\vdots \\
\Delta Q_K^m(e) &:= Q_K^m(e) - Q_{K-1}^m(e).
\end{aligned} \tag{3.32}$$

Using this definition, the charge conservation equation (3.31) can be generalized to

$$\text{SC circuit:} \quad \sum_{\substack{\text{at node } n \\ \text{edges } e \text{ incident}}} \Delta Q_k^m(e) = 0, \tag{3.33}$$

$$\forall \text{nodes } n, \text{ clock phases } k, \text{ clock cycles } m.$$

Obviously, this charge flow relationship has the same structure as the KCL (3.29) for the currents $i(e)$ in the time continuous case. Consequently, the quantities $\Delta Q_k^m(e)$ replace the currents $i(e)$ in the **a** and **b** vectors of the CCM for SC circuits.

For a time continuous circuit, there is a one to one correspondence between nodes of the electrical circuit and nodes in the representing CCM circuit graph (see Section 3.1.1). For each circuit graph node the respective KCL is formulated as part of the connection equation (3.3). In a SC circuit, there exist K different charge flows and the charge flow relationship (3.33) is valid for each clock phase $k, k = 1 \ldots K$. Consequently, there are K different charge flow relationships for each circuit node in the connection equation. For this reason, each node n of a SC circuit is represented by K graph nodes $n_k, k = 1 \ldots K$, in the respective SC circuit graph. Correspondingly, all primitive circuit elements (e.g. capacitors) have a K edge representation in the SC circuit graph. The SC graph representation of typical SC circuit elements is shown in Table 3.2. For simplicity, a two phase clock is assumed ($K = 2$). The in_1 and out_1 terminals refer to the connectivity

Component	Graph Representation
Capacitor in —‖— out C	in_1 —C→ out_1 in_2 —C→ out_2
Switch in —S/Φk— out	in_1 —S_1→ out_1 in_2 —S_2→ out_2
opamp in, out, Vref	in_1, op_in, op_out, out_1, in_2, out_2, op_in, op_out, Vref

Table 3.2: Graph representation of SC components for a 2 phase clock

of a component in phase 1, the in_2 and out_2 terminals to the connectivity in phase 2. The description in case of a clock with K phases would result in K-fold *in*- and *out* nodes. The symbols Φ_k at the switch of Table 3.2 indicates that the switch is closed in phase k and open in all other phases. The edges representing the switch S are denoted differently (with S_1 and S_2) for the two phases because of the different behaviour of the switch in the two different phases (closed in phase k / open otherwise). Since the electrical behaviour of the opamp and the capacitor C is independent of the phase, the respective edge notation is identical for both phases.

Based on this graph representation, the CCM connection equation (3.3) of SC circuits can be derived with the same procedures[2] as in the case of the time continuous circuits.

The CCM component equations of SC elements can be formulated in the time- or in the z-domain. Assuming again a two phase clock, these equations are shown in Table 3.3 and 3.4 respectively. Herein, v_k^m denotes the voltage across an edge in clock phase k and clock cycle m, $v_k(z)$ is the respective z-transform and $\Delta Q_k(z)$ denotes the z-transform of

[2]The procedures make use of node incidence matrices (see Section 3.1.1 and the thesis [79]).

ΔQ_k^m. For each edge in the graph representation (Table 3.2), there exists one row in the component matrix equation of the respective element. The opamp is described with zero output resistance.

component	component equation
capacitor	$\begin{pmatrix} \Delta Q_1^m \\ \Delta Q_2^m \end{pmatrix} = C \cdot \begin{pmatrix} v_1^m - v_2^{m-1} \\ v_2^m - v_1^m \end{pmatrix}$
switch closed in phase 1	$\begin{pmatrix} v_1^m \\ \Delta Q_2^m \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
switch closed in phase 2	$\begin{pmatrix} \Delta Q_1^m \\ v_2^m \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
opamp	$\begin{pmatrix} v_{op\_out1}^m \\ \Delta Q_{op\_in1}^m \\ v_{op\_out2}^m \\ \Delta Q_{op\_in2}^m \end{pmatrix} = \begin{pmatrix} A \cdot v_{op\_in1}^m \\ 0 \\ A \cdot v_{op\_in2}^m \\ 0 \end{pmatrix}$

Table 3.3: Time-domain component equations of SC elements

component	component equation
capacitor	$\begin{pmatrix} \Delta Q_1(z) \\ \Delta Q_2(z) \end{pmatrix} = C \cdot \begin{pmatrix} 1 & -z^{-1} \\ -1 & 1 \end{pmatrix} \begin{pmatrix} v_1(z) \\ v_2(z) \end{pmatrix}$
switch closed in phase 1	$\begin{pmatrix} v_1(z) \\ \Delta Q_2(z) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
switch closed in phase 2	$\begin{pmatrix} \Delta Q_1(z) \\ v_2(z) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
opamp	$\begin{pmatrix} v_{op\_out1}(z) \\ \Delta Q_{op\_in1}(z) \\ v_{op\_out2}(z) \\ \Delta Q_{op\_in2}(z) \end{pmatrix} = \begin{pmatrix} A \cdot v_{op\_in1}(z) \\ 0 \\ A \cdot v_{op\_in2}(z) \\ 0 \end{pmatrix}$

Table 3.4: z-domain component equations of SC elements

3.3.2 Example: SC Integrator Circuit

The introduced CCM graph representation of SC circuits is illustrated with an example. In Figure 3.5 a SC integrator is shown, node 1 being the input node and node 5 being the output node. The corresponding SC graph is shown in Figure 3.6. Since there are two clock phases Φ_1 and Φ_2, each circuit node n is represented by two SC graph nodes n_1 and n_2 which results in two similar graph structures. The two structures referring to the two phases differ merely in the characteristic of the switches $S1$, $S2$ and $S3$.

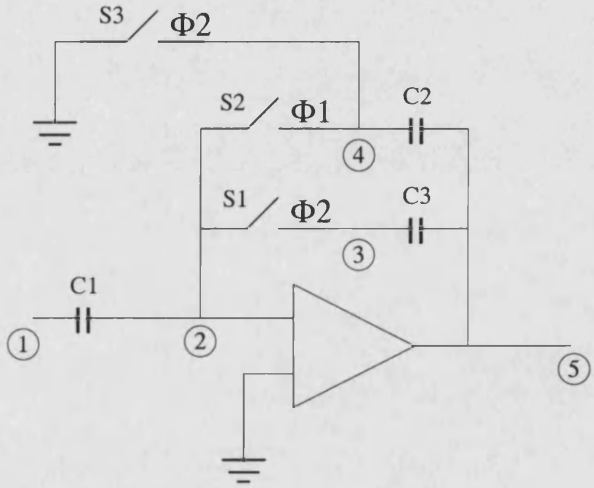


Figure 3.5: SC integrator circuit

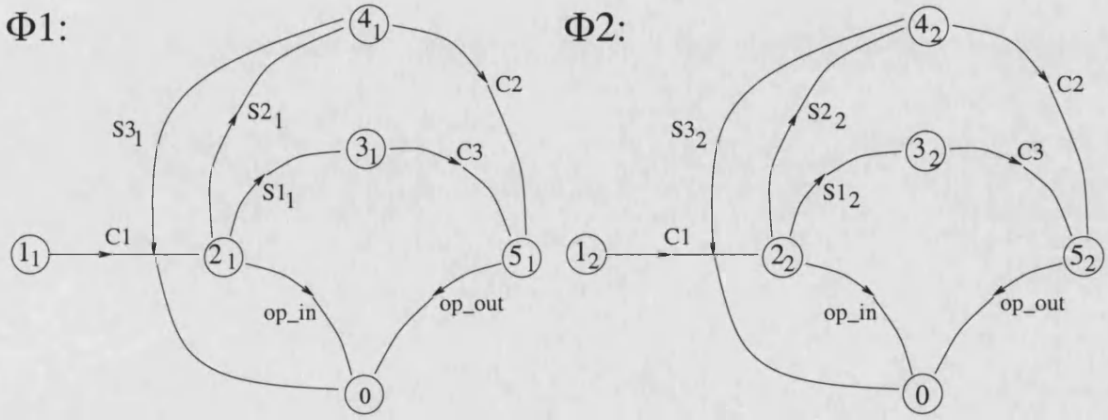
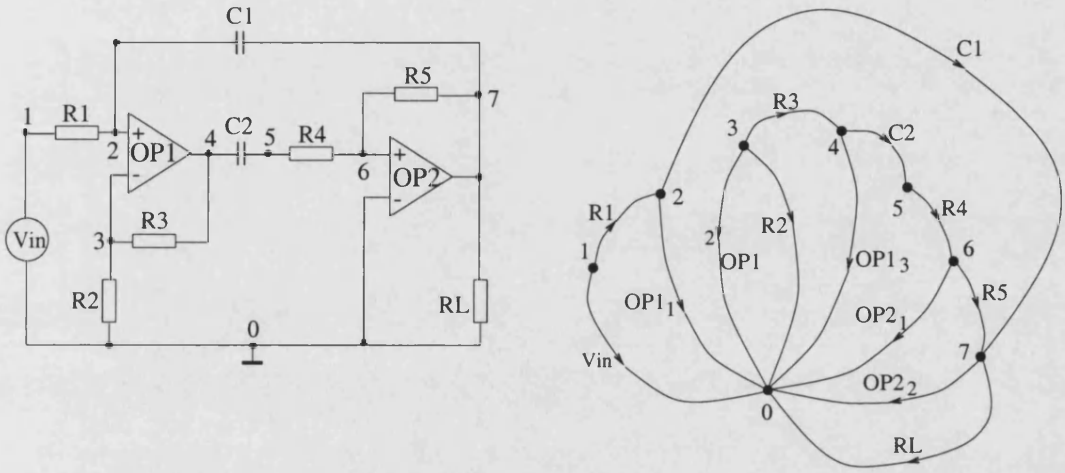


Figure 3.6: Graph representation of SC integrator circuit

3.4 Experimental Results

In this section, the ANSI C implementation of the developed HFDA [97] is applied to a practical circuit example to examine the effectiveness of the hierarchical approach. Deviation fault diagnosis of the bandpass filter [98] shown in Figure 3.7a has been investigated [81]. In the circuit graph illustrated in Figure 3.7b, the opamps $OP1$ and $OP2$ are rep-



a) filter circuit b) circuit graph

Figure 3.7: Bandpass filter circuit and corresponding graph

resented as 3 edge ($OP1_1$, $OP1_2$ and $OP1_3$) and 2 edge ($OP2_1$ and $OP2_2$) hierarchical components respectively. For $OP1$, the three edges are from the two inputs and one output to the reference node (ground 0), hence a total of three edges. For $OP2$, as the positive input is connected to ground, this reduces to a two edge graph. The hierarchical component equations of the opamps are given as

$$\begin{pmatrix} i_{OP1_1} \\ i_{OP1_2} \\ v_{OP1_3} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ A1 & -A1 & 0 \end{pmatrix} \begin{pmatrix} v_{OP1_1} \\ v_{OP1_2} \\ i_{OP1_3} \end{pmatrix} \tag{3.34}$$

and

$$\begin{pmatrix} i_{OP2_1} \\ v_{OP2_2} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ -A2 & 0 \end{pmatrix} \begin{pmatrix} v_{OP2_1} \\ i_{OP2_2} \end{pmatrix}, \tag{3.35}$$

where $A1$ and $A2$ are the open loop gain of opamp $OP1$ and $OP2$ respectively. These opamp equations are the same as in Section 3.2.1, except that the power supply terminals have been removed and the output resistance R_{out} is set to zero for simplicity. The nominal values of the circuit component parameters are summarized in Table 3.5. Parametric fault

$A1$	$A2$	R_1	R_2	R_3	R_4	R_5	R_L	C_1	C_2
1000	1000	$50k\Omega$	$50k\Omega$	$50k\Omega$	$50k\Omega$	$50k\Omega$	$1M\Omega$	$50pF$	$50pF$

Table 3.5: Parameter values for bandpass filter

diagnosis has been examined with the following global setting:

- tolerance decision threshold $\tau_i = 1\%$ for all parameters
- test stimulus: $V_{in} = 1V$ at 50 kHz
- test points selected (measured quantities): i_{OP2_2}, i_{R_2}, v_{R_1}, v_{R_3} and v_{R_4}

After the injection of the different deviation faults into the circuit description, the bandpass circuit is simulated and the values of the test points i_{OP2_2}, i_{R_2}, v_{R_1}, v_{R_3} and v_{R_4} are extracted. These test point values are required by the HFDA for solving the Pseudo Circuit equations.

3.4.1 Parametric Fault Diagnosis under Ideal Conditions

Firstly, the HFDA is applied to the bandpass filter under ideal test conditions. Test conditions are defined in respect of the measurement precision and the tolerance behaviour of good (fault-free) components. The ideal test conditions are:

- high precision in measurement results. This has been realized by using a simulator resolution of 10 digits.
- good components having zero deviation from nominal value.

Under these conditions, the results of the HFDA for single fault diagnosis are summarized in Table 3.6. In the left column, the injected single parametric faults are listed. The

injected deviation fault	HFDA results		
	ambiguous components	faulty component (or edge)	diagnosis performance
no fault	–	–	correct
$\Delta R_1 = 10\%$	–	R_1	correct
$\Delta R_2 = 10\%$	–	R_2	correct
$\Delta R_3 = 10\%$	–	R_3	correct
$\Delta R_4 = 10\%$	–	R_4	correct
$\Delta R_5 = 10\%$	–	R_5	correct
$\Delta R_L = 10\%$	–	R_L	correct
$\Delta C_1 = 10\%$	–	C_1	correct
$\Delta C_2 = 10\%$	–	C_2	correct
$A1 = 10\%$	–	$OP1_3$	correct
$A2 = 10\%$	–	$OP2_2$	correct
leakage current at + input node of $OP1$	R_1, $OP1$	–	incorrect
leakage current at − input node of $OP1$	R_3, $OP1$	–	incorrect
leakage current at − input node of $OP2$	–	$OP2_1$	correct

Table 3.6: Results of single deviation fault diagnosis

opamp leakage current faults have been modelled by adding a $1M\Omega$ resistor from the respective opamp input nodes to ground. In the three right columns of Table 3.6, the HFDA results are shown. Most of the single parametric faults are diagnosed correctly by the HFDA. Moreover, because of the hierarchical approach, diagnosis is achieved within relatively short computing time[3]. In the case where a hierarchical component is diagnosed faulty, the HFDA also provides some information about the location of the fault within the component. For example, with the leakage current fault at the negative input node of opamp $OP2$ (last row of Table 3.6), HFDA outputs the edge $OP2_1$ to be faulty. From

[3]Analysis of the computational efficiency of the HFDA in comparison with the previous non-hierarchical FDA can be found in [79].

this information it can be concluded that the input behaviour of the opamp deviates from the correct one.

The leakage current faults at the inputs of opamp $OP1$ are not diagnosed correctly. The reason is that the testee partitions including all the ambiguous components (R_1 and $OP1$ / R_3 and $OP1$) are untestable, i.e. the Pseudo Circuit and the Tableau Equations cannot be solved.

3.4.2 Parametric Fault Diagnosis under Real Conditions

So far, the performance of the HFDA under ideal conditions has been investigated. It is clear that in reality the measurements can only be accomplished with limited precision. Moreover, all the circuit components, both the faulty and the fault-free ones, reveal more or less parametric deviations from the ideal nominal value. The performance of the HFDA under those conditions is now analysed.

Finite Measurement Precision

Firstly, the influence of inaccuracies of the test measurements is considered. For this purpose, the resolution of the simulated bandpass test point values has been reduced from 10 to 5 digits. A 10% parametric deviation fault has been injected in R_4, R_5 and C_1 in turn. The diagnosis results are shown in Table 3.7. The diagnosis results are all incorrect except

injected fault	HFDA results		
	ambiguous components	faulty component (or edge)	diagnosis performance
$\Delta R_4 = 10\%$	–	R_4	correct
$\Delta R_5 = 10\%$	–	R_5, R_L	incorrect
$\Delta C_2 = 10\%$	–	R_L	incorrect

Table 3.7: Results of single fault diagnosis at reduced measurement precision

for the fault on R_4. This shows that the diagnosis procedure is very sensitive to precision in the test point values. The requirement on precision depends on the particular CUT and the respective parameter values. In general, large value resistors are more sensitive to a loss of precision of the test points, as even a small variation in current causes a large change in the voltage across them. For this reason, the resistor R_L is misdiagnosed as faulty.

Tolerance Effects

Now, the influence of parametric tolerances of fault-free components on the performance of the HFDA is investigated. For this purpose, parametric deviations on fault-free components are injected into the bandpass circuit of Figure 3.7. According to equation (3.20), the deviations of the fault-free components are smaller than the tolerance decision threshold τ. The experiments performed are defined in Table 3.8. The results of these experiments

experiment number	injected fault	deviations of good components	tolerance threshold τ
1	$\Delta R_4 = 10\%$	$\Delta C_2 = 1\%$	5%
2	no fault	$\Delta R_5 = -0.1\%$	1%
3	no fault	$\Delta A1 = -0.9\%$	1%

Table 3.8: HFDA experiments concerning good component deviations

experiment number	HFDA results		
	ambiguous component	faulty component (or edge)	diagnosis performance
1	–	R_1, R_5 and C_1	incorrect
2	–	–	correct
3	–	R_5	incorrect

Table 3.9: HFDA results for the experiments of Table 3.8

are summarized in Table 3.9. Two diagnosis results are incorrect. The reason behind the misdiagnosis lies in the Self Test algorithm [57, 86] (see Figure 3.2) and the implemented decision algorithm which is based on the assumption: *"in the case that all testers components are good, i.e. in tolerance, the test result of the respective test cycle are reliable."* This statement is correct in the case when good components in the tester partition have zero parametric deviation, as in such a case the results obtained from solving the Pseudo Circuit equation (3.14) are reliable. However, when the good tester components reveal a parametric deviation (deviations in $\mathbf{a_1}$ and $\mathbf{b_1}$), it usually happens that the testee values $\mathbf{a_2}$ and $\mathbf{b_2}$ obtained by the Pseudo Circuit equation are corrupted in the sense that the tolerances of the good tester components *mask* the true testee results. Consequently, in the presence of parametric tolerances of good components, the applied decision algorithm is not valid.

3.5 Summary & Conclusions

In this chapter, CUT modelling aspects for parametric fault diagnosis of analogue circuits have been presented. The FDA of [57, 86] has been adopted. The advantage of this method is that it enables automatic test point selection merely based on circuit topology information [57]. This has been used to develop a technique supporting a structured DFT approach to parametric characterisation test.

To overcome the circuit size limitations of the FDA of [57, 86], a hierarchical modelling strategy has been presented which resulted in the developments of a hierarchical fault diagnosis algorithm (HFDA). The HFDA reduces significantly the computing time of fault diagnosis by reducing the size of the matrices in the CCM. As a by-product of the higher level of abstraction within the hierarchical approach, the number of required test points is diminished, which is essential with respect to the applicability of the HFDA to integrated

circuits. Additionally, the CCM circuit graph representation has been modified such that a description of SC circuits is possible.

The HFDA has been used to diagnose parametric deviation faults in an active bandpass circuit example. The diagnosis results are positive in the sense that the HFDA does diagnose faults under ideal conditions with reduced computing time compared to the previous FDA and therefore allows DFT considerations for large analogue ICs. Work has to be done on the decision algorithm within the Self Test approach to compensate for tolerance masking effects and to reduce the sensitivity of the HFDA to measurement inaccuracies. The ideas presented in [99] may be helpful in this direction.

Further experimental results (e.g. concerning diagnosis of multiple deviation faults) and a more detailed analysis of the HFDA properties which are not directly related to CUT modelling (e.g. test point selection) can be found in [79].

Chapter 4

A Review of Symbolic Analysis

In the previous chapters, parametric deviation fault diagnosis during characterisation has been addressed. The following chapters are dedicated to the analysis of parametric tolerance effects in the design process.

In this chapter, symbolic analysis methods and applications for analogue circuits are presented. The capabilities and limitations of different symbolic approaches are reviewed to evaluate the potential of symbolic techniques for tolerance and sensitivity analysis. Emphasis will be drawn to the applicability of the procedures to large scale analogue circuits. Herein, the use of hierarchical decomposition is a promising approach for efficiently reducing the inherent computational complexity.

4.1 State of the Art

4.1.1 Definition of Symbolic Analysis

Symbolic analysis is a formal technique to calculate the characteristics of a system where the independent variables, frequency (or time) and some (or all) of the system parameters, are represented by symbols. The technique is complementary to numerical analysis where the independent variables are represented by numbers. Symbolic methods can be applied

to a large variety of physical systems, such as mechanical, thermal or electronic systems. Symbolic analysis of electronic circuits is divided into techniques for analogue and digital networks respectively.

Reviews of symbolic analysis for analogue circuits can be found in [100, 101, 102, 103, 104]. Almost all of the procedures concern the description of linear networks in the frequency domain. For lumped, linear, time-invariant circuits, the symbolic network functions obtained are rational functions in the complex frequency variable s (z for discrete-time circuits) and the circuit parameters x_j

$$H(s, x_1, \ldots, x_{N_X}) = \frac{N(s, x_1, \ldots, x_{N_X})}{D(s, x_1, \ldots, x_{N_X})} = \frac{\sum_i s^i \cdot a_i(x_1, \ldots, x_{N_X})}{\sum_i s^i \cdot b_i(x_1, \ldots, x_{N_X})}, \tag{4.1}$$

where the coefficients a_i and b_i of each power of s are polynomial functions in the circuit parameters x_j. These polynomials in their turn can be in nested format or expanded into the sum-of-product form. The network function H may describe the transfer function of a circuit, its input or output resistance, or more generally, any relation between the input/output voltages and currents V_{in}, I_{in}, V_{out} and I_{out}. For illustration, consider the 2-stage ladder circuit shown in Figure 4.1. Applying symbolic analysis yields the following

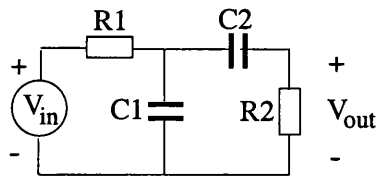


Figure 4.1: 2-stage ladder circuit

transfer function in the sum-of-product form

$$H(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{s\, C_2 R_2}{1 + s(R_1C_1 + R_1C_2 + R_2C_2) + s^2(R_1R_2C_1C_2)}. \tag{4.2}$$

In this equation all circuit parameters are represented by symbols, which refers to a fully symbolic analysis. A mixed symbolic-numerical analysis is also possible where only some

of the parameters are represented by symbols and the others by their numerical values. In the extreme case, only the frequency variable s remains as a symbol.

The generation of symbolic network functions by hand is tedious and error prone, especially for large circuits. For this reason, a lot of automatic symbolic analysers such as ISAAC [105, 106], ASAP [107, 108], SYNAP [109, 110], SAPEC [111], SSPICE [112], SCYMBAL [94], SCAPP [89], GASCAP [113], SANTAFE [114] and SAGA2 [115] have been developed. The basic flow of symbolic analysis programs is illustrated in Figure 4.2. The input is a circuit description, typically a netlist in SPICE format [116]. Since

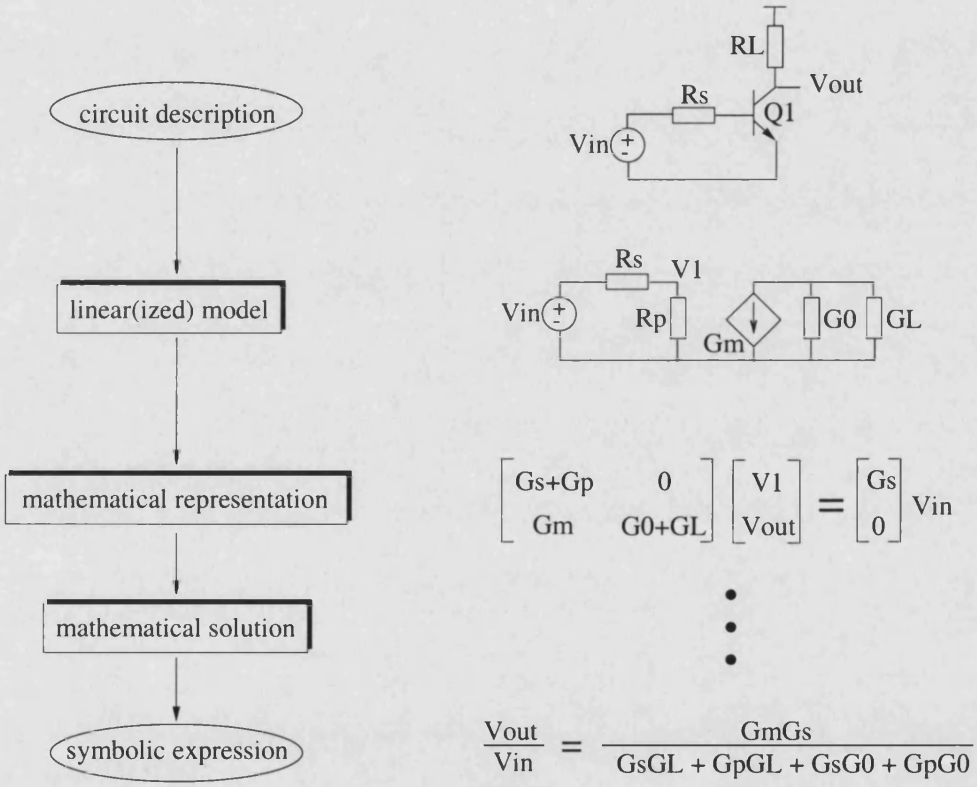


Figure 4.2: Flow chart of symbolic analysers [102]

symbolic analysis is mainly restricted to linear circuits, the first step for nonlinear circuits is to generate a linearized small-signal model of the circuit. After the user has specified the network function that he wants, a mathematical representation of the circuit is created.

Besides the matrix representation indicated in Figure 4.2 there exist also a variety of graph-based descriptions. Based on the mathematical representation, an internal algorithm solves for the requested network function. The following section will describe the applications of symbolic analysis.

4.1.2 Applications of Symbolic Analysis

Symbolic analysis is completely different from numerical circuit simulation. By providing analytical information it complements the results from numerical simulation and offers some new solutions when classical techniques fail. The major applications of symbolic techniques can be summarized as follows [102]:

Insight into Circuit Behaviour

Numerical simulation generates a set of numbers tabulated or plotted. Although these numbers describe the circuit behaviour correctly, they are specific for a particular set of parameters values. No indication is given which circuit elements determine the observed performance and no solutions are suggested when the circuit does not meet the specifications.

A symbolic simulator returns closed-form symbolic expressions for the characteristics of a circuit. These expressions remain valid even when the numerical parameter values change. By inspecting the different terms in the network function, the influence of the circuit parameters on performance can be derived. As such, symbolic analysis gives a different perspective on a circuit than provided by numerical simulators, which is most appropriate for students and practising designers in order to obtain insight into the behaviour of a circuit [101, 113, 117, 118]. Even experienced designers obtain valuable information from expressions describing second order characteristics such as power-supply rejection ratio and harmonic distortion.

Model Generation for Circuit Optimization

A symbolic simulator automatically generates analytic expressions for the ac characteristics of a circuit. The expressions can be used as a model within a circuit optimization program. During circuit optimization, the network behaviour needs to be determined several times with modified parameter values. By using the analytic model, optimization time is strongly reduced compared to a full numerical simulation at each iteration. This approach is adopted in the OPTIMAN [119] and OPASYN [120] tools. In [121] and [122] symbolic equations are picked out of a library to synthesize specific classes of analogue-to-digital converters in terms of their subblocks. The use of a symbolic simulator largely reduces the effort required to develop the analytic model for a new circuit schematic. In this way, an open, non-fixed topology analogue circuit library can be created where the designer himself can easily include new circuit topologies [123, 124]. The optimization approach of [125] additionally takes into account the DC operating point conditions whereas [126] concentrates on filter applications.

Circuit Exploration

Symbolic simulation can be used to *interactively* or *automatically* explore and improve new circuit topologies [94, 127]. The symbolic expressions are used to get insight into the behaviour of new circuit topologies (*interactive* approach) or to speed up numerical simulations (*automatic* approach).

Repetitive Formula Evaluation

Many tasks within the design, such as sensitivity and tolerance analysis, yield estimation, parameter optimization and fault diagnosis, are very time intensive. Since the circuit topology is the same for all parameter and frequency values it is a good idea to exploit

this situation maximally to speed up simulation times. Numerical results for the circuit behaviour can be obtained by evaluating the results of a symbolic analysis at a specific numerical point for each symbol (parameter or frequency). So ideally, only one simulation run is needed to analyse the circuit, and successive evaluations of the symbolic results replace the need for any extra numerical iterations through the simulator. This method is most useful when the symbolic expressions are compiled to increase evaluation speed [128].

The efficiency of this technique has been shown in [94] for the analysis of switched capacitor circuits. Using symbolic expressions, orders of magnitude of acceleration compared to numerical simulation can be achieved in frequency analysis applications [129], tolerance analysis [130] or fault diagnosis [128].

Further Applications

The above enumeration of applications is by far not complete. Modern applications include automatic behavioural model generation [131] and speeding up of numerical techniques, such as improving convergence of relaxation methods in electrothermal analysis [132]. It can be concluded that symbolic analysis is very helpful in analogue circuit analysis and design.

4.1.3 Capabilities and Limitations of Symbolic Analysis

The different circuit types which can be examined and the different analysis types which are presently feasible within symbolic approaches are shown in Figure 4.3. The capabilities of symbolic analysis have been extended both in functionality and computational efficiency during the last fifteen years. Concerning functionality, symbolic distortion analysis of weakly nonlinear circuits based on Volterra series has been proposed in [133, 134]. Symbolic pole/zero analysis has become feasible in [107, 135] which is of significant help

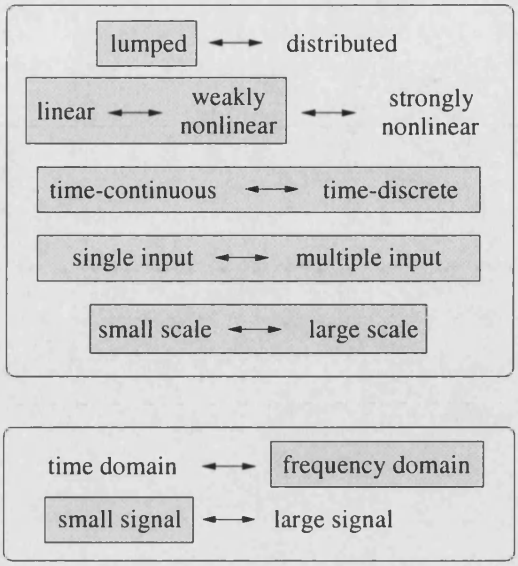


Figure 4.3: Capabilities and limitations of symbolic analysis [102]
above: circuit types
below: analysis types
shaded (unshaded) items currently can (cannot) be tackled with symbolic analysis

for interactive circuit improvement. As closed-form solutions for poles and zeros can only be found for lower order systems, approximate expressions are obtained using the pole-splitting hypothesis. Simulators dedicated to analogue integrated circuits were presented in [106, 108, 109]. The properties of integrated circuits are taken into account by using a built-in small-signal linearization and by exploiting matching characteristics of neighboured devices on the die. Symbolic analysis of large scale circuits has become feasible using hierarchical decomposition approaches [89, 136].

On the downside, open research topics are still large-signal behaviour, time-domain simulation and symbolic analysis of strongly nonlinear circuits.

4.1.4 Algorithmic Aspects of Symbolic Analysis

In the literature, many symbolic analysis techniques have been reported. The methods can be classified as follows:

- matrix methods
- determinant methods
- signal-flow-graph methods
- tree-enumeration methods
- parameter reduction methods
- interpolation methods

Within matrix approaches, a matrix representing Kirchhoff's equations is used for analysis. Applying Gaussian elimination the symbolic network functions are derived, e.g. [89]. Determinant methods are based on Cramer's rule and symbolic calculation of determinants, for example with help of the sparse recursive Laplace expansion algorithm used in [106] or by determinant decision diagrams applied in [137, 138]. Signal-flow-graph methods, e.g. [136, 139], use a signal-flow-graph representation of Kirchhoff's laws and derive the network functions based on finding loops and paths of all orders in the graph according to Mason's rule. Tree-enumeration methods represent the nodal admittance matrix by a directed graph. Symbolic simulation is then accomplished by enumeration of all directed trees. This method however, has difficulties in handling all types of controlled sources and suffers from the term cancellation problem (the generation of equal terms with opposite sign) which consumes extra CPU time. Parameter extraction methods are based on a recursive extraction of the symbolic parameters one by one out of matrix determinant, each time splitting up the determinant into two other determinants not containing the extracted symbol. Parameter reduction is best suited for generating partially symbolic network functions where only a small fraction of the circuit parameters are represented by symbols. Interpolation methods numerically simulate the circuit at different frequency

points and then fit the network function coefficients to the obtained results [140, 141, 142]. By this means, expressions with the frequency variable as the only symbol are generated.

In recent years, two techniques emerged as most flexible and efficient for a fully symbolic circuit analysis: the signal-flow-graph method (implemented for instance in ASAP [107]) and the matrix/determinant methods (used in ISAAC[106]). Experimental data of the simulation times of the different simulators show that both techniques can have comparable performance and that none of the methods is a priori superior over the other [102].

4.1.5 Symbolic Analysis of Large Circuits

Symbolic network functions in the fully expanded sum of product form tend to be lengthy, especially in the case of large circuits. To illustrate this, the ladder circuit shown in Figure 4.4 is considered. The goal is to obtain the output admittance function $Y_n = \frac{I_{out}}{V_{out}}$

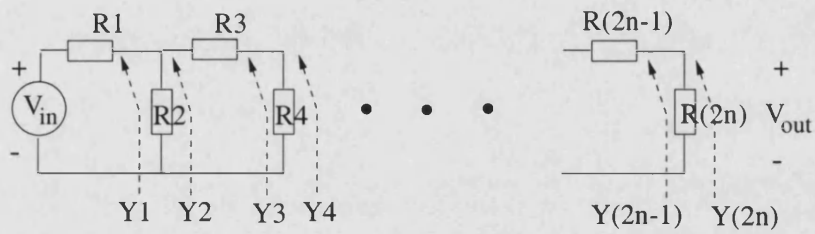


Figure 4.4: Resistive ladder circuit

of the network consisting of the resistors $R_1, \ldots, R_n$. For $n = 4$ the impedance function in the expanded sum of product form is

$$Y_4 = \frac{G_4G_1 + G_4G_2 + G_4G_3 + G_3G_1 + G_3G_2}{G_1 + G_2 + G_3}, \tag{4.3}$$

where $G_i = 1/R_i$. The number of terms in the numerator and denumerator fulfil the Fibonacci difference equation [89]

$$N(n+2) = N(n+1) + N(n), \qquad n = 0, 1, 2, \ldots$$
$$N(0) = 0,\ N(1) = 1. \tag{4.4}$$

An explicit solution to this equation is

$$\begin{aligned} N(n) &= \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right] \\ &\approx 0.447 \cdot 1.618^n \qquad \text{for large } n. \end{aligned} \tag{4.5}$$

This example shows that even in the case of sparse circuits, the number of terms in Y_n grows exponentially with circuit size. So, in the case of a 100 resistor ladder network, the expanded expression would contain more than 10^{20} terms, which requires unrealistic huge computer storage. For large circuits, expanded symbolic expressions can therefore neither be used to speed up iterative numerical simulations nor for easy interpretation of circuit behaviour. Basically, there exist two different approaches to improve the situation:

- symbolic expression simplification
- hierarchical decomposition

The two techniques and their capabilities are now presented.

Symbolic Expression Simplification

Symbolic expression simplification reduces the number of terms in the network function by discarding smaller terms against larger terms. Consider, for example, an arbitrary expression

$$g_{m1} + g_{o1} + g_{o2} + s(C_L + C_{db1}) \tag{4.6}$$

where g_{m1} refers to a transistor transconductance, g_{o1} and g_{o2} to transistor output conductances, C_L to a load capacitance and C_{db1} to a transistor parasitic capacitance. Now, if the

ratio of the values of the transistor small-signal parameters are such that $g_{m1} \gg g_{o1}, g_{o2}$ (typically, the transistor transconductances are larger than the output conductances) and $C_L \gg C_{db1}$, then the following expression, which contains only two terms instead of five, is a good approximation

$$g_{m1} + sC_L. \tag{4.7}$$

This simplified expression shows the dominant contributions in a much clearer way at the penalty of some error. Generally, symbolic expression simplification has to find an approximating expression $h(s, x_1, \ldots, x_{N_X})$ for the exact network function $H(s, x_1, \ldots, x_{N_X})$ such that the relative error is bounded

$$\left| \frac{H(s, x_1, \ldots, x_{N_X}) - h(s, x_1, \ldots, x_{N_X})}{H(s, x_1, \ldots, x_{N_X})} \right| \leq \epsilon_{max}, \tag{4.8}$$

where the circuit parameters $x_1, \ldots, x_{N_X}$ can be varied over a certain range around the nominal design point and ϵ_{max} is the maximal error allowed by the user. Symbolic simplification methods are implemented in simulators like ISAAC [106], SYNAP [110], ASAP [107], SSPICE [112] in [143] and [144]. Since the approximation should be valid in the whole frequency range, all coefficients a_i and b_i in equation (4.1) need to be simplified separately up to an error ϵ_{max} [118]. Additionally, the actual errors on the different coefficients may have opposite sign which requires the accuracy of poles and zeros to be observed [107]. The requirement to obtain a correct approximation under all conditions results in the necessity to (partially) expand each individual coefficient polynomial a_i and b_i, especially because the generated exact expressions are not always cancellation-free, in which case an (at first sight) unimportant term can become dominant after the cancellations have been carried out. As a result, the time complexity of these algorithms grows with circuit size and their application is limited in the case of filter circuits to networks in the range of 40 nodes and in the case of semiconductor circuits to networks with not more than 15 transistors [102]. From these facts it can be concluded that simplification

techniques are helpful to improve the insight into circuit behaviour of moderately sized networks only. For model generation and repetitive formula evaluation, e.g. tolerance analysis, however, there exist more efficient approaches.

Hierarchical Decomposition

The hierarchical decomposition approach relies on the observation that the number of terms in a symbolic expression can be reduced by using a *sequence of expressions* format. To illustrate this, the ladder circuit of Figure 4.4 is considered again. Instead of using a single expanded expressions, the output impedance is now described by a sequence of small nested expressions:

$$\begin{aligned} Y_1 &= G_1, \\ Y_2 &= Y_1 + G_2, \\ Y_3 &= \frac{Y_2 G_3}{Y_2 + G_3}, \\ Y_4 &= Y_3 + G_4. \end{aligned} \tag{4.9}$$

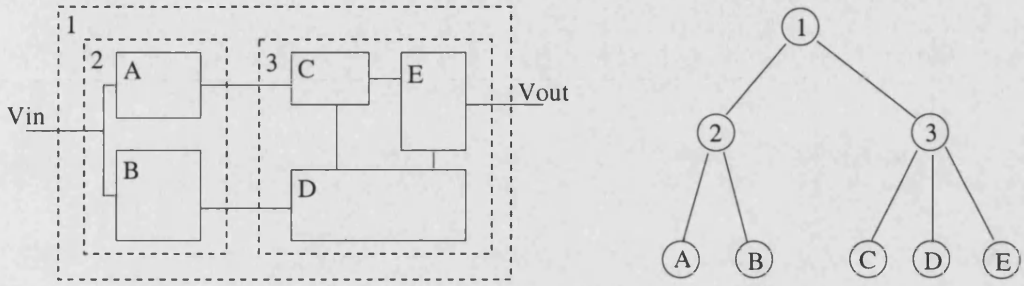
The calculation of this sequence requires only 3 additions, 1 multiplication and 1 division whereas the calculation of the expanded expression in equation (4.3) requires 6 additions, 5 multiplications and 1 division. In case of larger ladder circuits with many resistors, the number of terms in a sequence of expressions is given as [89]

$$N(n) = \begin{cases} 2.5n - 2 & \text{for } n \text{ even} \\ 2.5n - 1.5 & \text{for } n \text{ odd} \end{cases} \tag{4.10}$$

which exhibits a linear growth with respect to the number of resistors n contrary to the exponential growth of terms in the expanded expression (eq. 4.5). As a result, a 100 resistor ladder circuit can be described using a sequence of expression with 248 terms only (as opposed to 10^{20} terms using the expanded expression format). Consequently, if one wants to extend the capabilities of symbolic analysis to large scale circuits, the only way is

to avoid the single expanded expression description but to use a sequence of expressions.

In this respect, an interesting method which can exploit the topology of the circuit is the use of hierarchical decomposition [89, 136]. The circuit is recursively decomposed into more or less loosely connected subcircuits as illustrated in Figure 4.5. The hierarchical



a) hierarchical circuit partitioning b) partition tree

Figure 4.5: Hierarchical circuit partitioning and corresponding partition tree

partitioning is modelled by an partition tree. After partitioning, the lowest level subcircuits (leaves of the partition tree) are analysed separately by the symbolic simulator resulting in the following sets of symbolic network functions:

$$H_A = f_A(s, X_A), \tag{4.11}$$

$$H_B = f_B(s, X_B), \tag{4.12}$$

$$H_C = f_C(s, X_C), \tag{4.13}$$

$$H_D = f_D(s, X_D), \tag{4.14}$$

$$H_E = f_E(s, X_E), \tag{4.15}$$

where X_Y is the set of circuit parameters of the subcircuit Y. By proceeding the partition tree bottom up, the network functions of the nonleaf subcircuits are obtained without expansion in terms of the composing subcircuits. For subcircuits 2 and 3, this is in terms

of the above leaf network functions

$$H_2 = f_2(s, H_A, H_B), \tag{4.16}$$

$$H_3 = f_3(s, H_C, H_D, H_E). \tag{4.17}$$

Finally, the top-level network function of the complete circuit is given as

$$H_1 = f_1(s, H_2, H_3). \tag{4.18}$$

The result of the hierarchical procedure is a sequence of small expressions having a hierarchical dependence on each other. The technique of hierarchical decomposition allows symbolic analysis of large-scale analogue circuits. Both the CPU time for symbolic analysis and the number of operations needed to numerically evaluate the obtained expressions increases typically linearly (quadratically in the worst case) with circuit size [103].

Implementations of the hierarchical approach are FLOWUP [145] using signal-flow-graph analysis and SCAPP [89] using matrix based analysis. Since hierarchical decomposition yields very compact symbolic expressions, this approach is useful for repetitive formula evaluation within iterative applications as circuit optimization and tolerance analysis. The hierarchical symbolic approach can here provide a much more efficient solution compared to numerical simulation in terms of computational effort [117]. Non of these techniques, however, provide any approximation, which is important to obtain insight into the behaviour of semiconductor circuits.

4.1.6 Conclusion

In the above sections, an overview of the different symbolic techniques available today and their properties has been presented. The demands for the usability of a method in tolerance and sensitivity analysis within the design process are as follows

- possibility to analyse the influence of all circuit elements with arbitrary value range

of the element parameters,

- good accuracy over the whole frequency range,
- applicability to large scale circuits,
- efficiency in terms of numerical evaluation time.

Since statistical analysis is a highly iterative task which typically requires many circuit simulations, computational efficiency is the essential point. Symbolic analysis requires only one simulation run to analyse the circuit, and successive evaluations of the results replace the need for any time intensive numerical iterations through the simulator. As a result, symbolic analysis has been proven to be more efficient than numerical simulation in iterative applications, e.g. [94, 129, 146].

A comparison of the above requirements with the presented symbolic approaches shows that the hierarchical decomposition technique is the best solution. From the available hierarchical approaches, SCAPP (Symbolic Circuit Analysis Program with Partitioning) [89] has been selected as one of the modern simulators with high performance. It provides an efficient fully symbolic analysis capable of handling large scale circuits. The statistical methods presented in the next chapters are based on the obtained sequence of expressions generated by SCAPP. Novel sensitivity and tolerance analysis procedures are developed which are significantly faster than previously suggested approaches. As a result, statistical analysis even for large scale circuits becomes feasible in reasonable simulation times during the design process, thereby enhancing the quality of the product.

The next section describes the hierarchical symbolic approach of SCAPP and its relevant properties concerning the sensitivity and tolerance analysis procedures presented in this thesis.

4.2 Hierarchical Symbolic Analysis

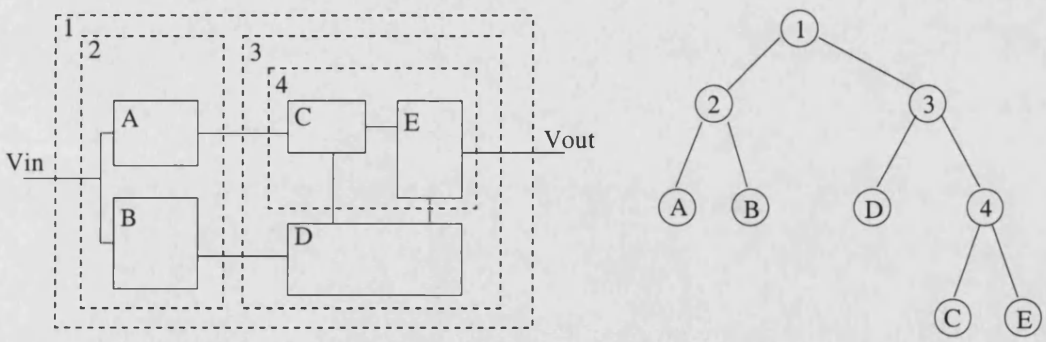
Hierarchical symbolic analysis in SCAPP follows the route:

1. binary circuit partitioning
2. subcircuit analysis
3. upward hierarchical analysis

These three steps are explained in the following sections.

4.2.1 Binary Circuit Partitioning

Any hierarchical network approach requires circuit partitioning. SCAPP is based on a recursive binary partitioning process. The circuit is decomposed into two subcircuits, ideally of similar size. The subcircuits, in their turn, are partitioned into two subcircuits, and so on. This process is then modelled by a Binary Partition Tree (BPT). For the circuit of Figure 4.5, the binary partitioning and the corresponding BPT are illustrated in Figure 4.6. By introducing the new intermediate subcircuit 4 (consisting of C and E)



a) binary circuit partitioning b) binary partition tree

Figure 4.6: Binary partitioning and corresponding binary partition tree

the partitioning shown in Figure 4.5 is transformed into the binary partitioning of Figure

4.6. The leaves, A, B, C, D and E, of the BPT refer to the smallest subcircuits (terminal blocks). The other nodes of the BPT, 1, 2, 3 and 4, refer to the complete circuit at highest hierarchical level and to the intermediate subcircuits (middle blocks) respectively.

The partitioning aims at decomposing the circuit into loosely connected subcircuits. In this respect "loosely connected" means that the number of tearing nodes between subcircuits is minimal which guaranties most efficient symbolic analysis. For this purpose, automatic network partitioning as suggested in [147, 148, 149] is most adequate. In addition, SCAPP allows for user defined subcircuits, which is appropriate for the typical design process where a circuit is developed using blocks out of a library.

4.2.2 Subcircuit Analysis

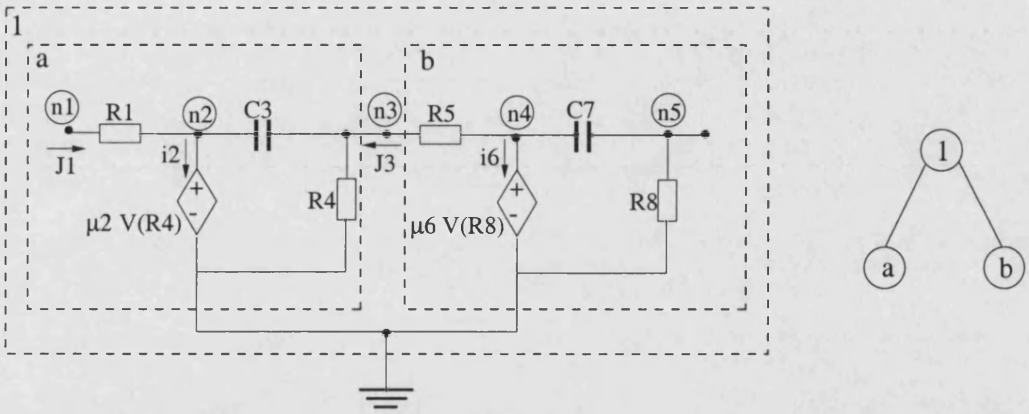
Subcircuit analysis refers to the characterisation of the terminal blocks (the leaves in the BPT) in terms of their electrical behaviour. SCAPP is based on Modified Nodal Analysis (MNA) [150]. The advantage of MNA is that it is able to describe all types of controlled sources. A Laplace Transform representation of the admittance values of the circuit elements is used in the formulation of the network equations. The MNA equation set has the following structure:

$$\begin{pmatrix} Y_n & B \\ C & D \end{pmatrix} \begin{pmatrix} V \\ I \end{pmatrix} = \begin{pmatrix} J \\ E \end{pmatrix} \tag{4.19}$$

where V is vector containing the node voltages, I contains the branch current variables, Y_n is the modified nodal admittance matrix, B, C and D are the contributions of the branch current relationship equations, J and E represent the independent current and voltage sources respectively.

Once the MNA equation set has been generated, the next step is suppressing all the internal variables of the subcircuit by applying Gaussian elimination. The result is a set of Reduced Modified Nodal Analysis (RMNA) equations which describes the electrical be-

haviour in terms of the external node voltages and branch currents. The benefit of variable suppression is a reduction in the size of the matrices and an elimination of information not needed for the analysis at higher hierarchical level. For illustration, the circuit of



a) circuit and its partitioning b) BPT

Figure 4.7: A simple circuit, its partitioning and its BPT [89]

Figure 4.7 is considered. The MNA matrix equation of the terminal circuit a is

$$\begin{pmatrix} G_1 & -G_1 & 0 & 0 \\ -G_1 & G_1 + sC_3 & -sC_3 & 1 \\ 0 & -sC_3 & sC_3 + G_4 & 0 \\ 0 & 1 & -\mu_2 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ i_2 \end{pmatrix} = \begin{pmatrix} J_1 \\ J_2 \\ J_3 \\ 0 \end{pmatrix}. \tag{4.20}$$

Gaussian elimination is applied to remove the internal variables v_2 and i_2. Herein J_2 is set to zero because no current is entering subcircuit a at terminal 2 form the other subcircuit. The RMNA description of terminal circuit a becomes

$$\begin{pmatrix} G_1 & -G_1\mu_2 \\ 0 & sC_3 + G_4 - sC_3\mu_2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_3 \end{pmatrix} = \begin{pmatrix} J_1 \\ J_3 \end{pmatrix} \tag{4.21}$$

which only contains dependencies in terms of the external voltages v_1 and v_3 and the currents J_1 and J_3 entering the subcircuit at the terminals $n1$ and $n3$ from the outside world. Similarly, the RMNA description of circuit b becomes

$$\begin{pmatrix} G_5 & -G_5\mu_6 \\ 0 & sC_7 + G_8 - sC_7\mu_6 \end{pmatrix} \begin{pmatrix} v_3 \\ v_5 \end{pmatrix} = \begin{pmatrix} J_3 \\ J_5 \end{pmatrix}. \tag{4.22}$$

4.2.3 Upward Hierarchical Analysis

Once an electrical characterisation of each terminal block has been generated, the behaviour of higher level subcircuits is calculated. The RMNA matrices of two lower level subcircuits are used for the description of the subcircuit on the next hierarchical level. The first step is to combine the two RMNA equations of the subcircuits according to their connectivity. Terminal block a and b of Figure 4.7, share the node $n3$ and ground. The matrix equation which describes circuit 1 as the interconnection of a and b is

$$\begin{pmatrix} G_1 & -G_1\mu_2 & 0 \\ 0 & sC_3 + G_4 - sC_3\mu_2 + G_5 & -G_5\mu_6 \\ 0 & 0 & sC_7 + G_8 - sC_7\mu_6 \end{pmatrix} \begin{pmatrix} v_1 \\ v_3 \\ v_5 \end{pmatrix} = \begin{pmatrix} J_1 \\ J_3 \\ J_5 \end{pmatrix}. \tag{4.23}$$

The algorithmic details of the hierarchical combination of two RMNA matrices are described in [89].

After combination of the two RMNA matrices, the second step is to suppress all the new internal variables of the higher level subcircuit. Applying again Gaussian elimination, a RMNA equation system is generated. For the circuit 1 of Figure 4.7, the elimination of the variable v_3 (set $J_3 = 0$) results in

$$\begin{pmatrix} G_1 & \frac{-G_1\mu_2 G_5\mu_6}{sC_3+G_4-sC_3\mu_2+G_5} \\ 0 & sC_7 + G_8 - sC_7\mu_6 \end{pmatrix} \begin{pmatrix} v_1 \\ v_5 \end{pmatrix} = \begin{pmatrix} J_1 \\ J_5 \end{pmatrix} \tag{4.24}$$

The combination of RMNA matrices is recursively applied while proceeding the BPT bottom up. When the root of the BPT is reached, a characterisation of the circuit at highest hierarchical level in terms of the requested external input and output variables has been generated.

The final RMNA equation system can easily be used to derive network functions. Defining for example the input terminal to be $n5$ and the output terminal to be $n1$ the transfer function of circuit 1 of Figure 4.7 is

$$H(s) = \left.\frac{v_1(s)}{v_5(s)}\right|_{J_1=0} = -\frac{1}{G_1}\frac{-G_1\mu_2 G_5\mu_6}{sC_3 + G_4 - sC_3\mu_2 + G_5}. \tag{4.25}$$

4.2.4 Sequence of Expressions(SOE)

The output of SCAPP is a *sequence of expressions* (SOE) which describes the network function requested by the user. The SOE is generated by storing the arithmetic operations accomplished during hierarchical analysis. Terminal block and upward hierarchical analysis produce for each subcircuit a matrix equation

$$\mathbf{M}\,\mathbf{L} = \mathbf{R} \tag{4.26}$$

where $\mathbf{M}$ is a RMNA matrix and $\mathbf{L}$ and $\mathbf{R}$ are the respective external voltage and current variables. The arithmetic operations for the generation of $\mathbf{M}$ are stored in a sequence of nested expressions as illustrated in Table 4.1. The last expression H_{16} refers to the transfer function of the circuit defined as in equation (4.25).

(sub)circuit	RMNA matrix $\mathbf{M}$	SOE expressions
a	$\begin{pmatrix} G_1 & -G_1\mu_2 \\ 0 & sC_3 + G_4 - sC_3\mu_2 \end{pmatrix} = \begin{pmatrix} H_1 & H_6 \\ 0 & H_7 \end{pmatrix}$	$H_1 = G_1$ $H_2 = \mu_2$ $H_3 = C_3$ $H_4 = G_4$ $H_5 = s \cdot H_3$ $H_6 = -H_1 \cdot H_2$ $H_7 = H_5 + H_4 - H_5 \cdot H_2$
b	$\begin{pmatrix} G_5 & -G_5\mu_6 \\ 0 & sC_7 + G_8 - sC_7\mu_6 \end{pmatrix} = \begin{pmatrix} H_8 & H_{13} \\ 0 & H_{14} \end{pmatrix}$	$H_1 = G_5$ $H_9 = \mu_6$ $H_{10} = C_7$ $H_{11} = G_8$ $H_{12} = s \cdot H_{10}$ $H_{13} = -H_8 \cdot H_9$ $H_{14} = H_{12} + H_{11} - H_{12} \cdot H_9$
1	$\begin{pmatrix} G_1 & \frac{-G_1\mu_2 G_5\mu_6}{sC_3+G_4-sC_3\mu_2+G_5} \\ 0 & sC_7 + G_8 - sC_7\mu_6 \end{pmatrix} = \begin{pmatrix} H_1 & H_{15} \\ 0 & H_{14} \end{pmatrix}$	$H_{15} = -\frac{H_6 \cdot H_{13}}{H_7 + H_8}$ $H_{16} = -\frac{H_{15}}{H_1}$

Table 4.1: Storing of arithmetic operations in a SOE

An important observation is that in electronic circuits the number of elements connected to a node (excluding ground and power supply nodes) is limited by a constant K which is in the order of 6 in the worse case. Then the MNA matrices are sparse and it is shown in [89] that the number of terms and arithmetic operations in the SOE grows only linearly with circuit size. The SOE approach is therefore superior to numerical simulation in terms of simulation time and can be used for fast numerical evaluation of the circuit's behaviour. For this reason, the SOE approach is best suited for tolerance and sensitivity analysis.

Finally, it should be noted that there exist also non-hierarchical methods to derive very compact SOEs purely using algebraic techniques. Examples for such techniques can be found in [151, 152].

4.3 Summary

In this chapter the state of the art of symbolic analysis of analogue circuits has been reviewed. The usefulness of symbolic simulation for highly iterative tasks as sensitivity and tolerance analysis has been concluded. For the examination of today's large scale circuits, the hierarchical approach of [89] is adopted in this thesis. The outcome of the hierarchical approach is a *sequence of expressions* describing the requested circuit behaviour(s). This SOE is used in the next chapters to develop novel methods for efficient symbolic sensitivity and tolerance analysis of large scale analogue circuits.

Chapter 5

Hierarchical Symbolic Sensitivity Analysis

In this chapter, hierarchical symbolic sensitivity analysis procedures for large scale analogue circuits are presented. An introduction to numerical and symbolic sensitivity methods is given. Previous hierarchical SOE approaches for fast sensitivity computation are reviewed and their limitations concerning multi-parameter sensitivity analysis are explained. The limitations mainly refer to the large number of arithmetic operations required when the sensitivities with respect to many parameters need to be determined. Two novel methods are described which overcome the limitations:

- the *balanced* symbolic sensitivity analysis,
- the *parallel* symbolic sensitivity analysis.

The first method reduces computational complexity by applying a hierarchical balanced partitioning strategy. The second one uses the SOE to calculate the sensitivities with respect to all parameters in parallel. Experimental results are presented to illustrated the applicability and effectiveness of both approaches.

5.1 Introduction

A classical approach to assess the tolerance of a circuit behaviour H is to simulate the circuit under different parametric conditions X around the nominal design point X_0. A popular representative of these approaches is the Monte Carlo analysis. Unfortunately, in most practical applications, hundreds or thousands of simulations need to be performed to get reliable results which makes the Monte Carlo analysis very time consuming, especially for large scale circuits. Consequently, this technique normally cannot be used in the design process.

Sensitivity analysis is here an effective alternative to investigate the influence of parametric deviations. In many cases the strays of the parameters $\Delta x_i = x_i - x_{i0}$ are not too large and the deviation of H can be calculated using a linear approximation:

$$\Delta H = \sum_{i=1}^{N_X} \left.\frac{\partial H}{\partial x_i}\right|_{X_0} \cdot \Delta x_i. \tag{5.1}$$

where N_X is the number of circuit parameters. The partial derivatives in this equation are called sensitivity functions:

$$sen(H, x_i)(X_0) = \left.\frac{\partial H}{\partial x_i}\right|_{X_0} \tag{5.2}$$

where the differentiation is normally performed at the nominal design point X_0. Typically, the calculation of equation (5.1) can be done much faster than performing a Monte Carlo analysis, as long as there are efficient sensitivity analysis methods available. More rigerous estimations for the circuit tolerances than that one of equation (5.1), can be obtained by using the sensitivity functions within the root sum square technique [153], or with help of worst case analysis [154] (see Chapter 6).

Applications of the sensitivity functions are by far not restricited to tolerance analysis. Examples for further applications are:

- **tolerance design:** the aim of tolerance design is not to estimate the circuit tolerances but to increase yield. Algorithmic methods can help the designer to solve this task. The algorithms choose the nominal design point X_0 ("design centering") or the component tolerances ("tolerance assignment") such that yield is increased. Herein sensitivity information is required to direct the search for the optimal solution in the parameter space (gradient method) [155]. Additionally, since the sensitivities are a measure for the influence of parametric deviations, the tolerance behaviour of a circuit can be optimized by minimizing the sensitivity functions [156].

- **fault diagnosis:** sensitivity functions can also be applied to analogue circuit fault diagnosis [28, 53, 157]. Using measurement results and sensitivity equations, component parameter values are calculated. As a result, parametric deviations faults can be diagnosed and located.

- **circuit optimization:** the goal of circuit optimization is to choose the component parameter values such that the simulated behaviour fits the specified one. Mathematically, such tasks can be formulated as non-linear optimization problems. Methods for electronic circuit optimization are typically based on gradient methods (e.g. Fletcher-Powell) and therefore require sensitivity information [158]. Further applications of sensitivity information in design optimization can be found in [159].

This summary shows that sensitivity analysis plays an essential role in analogue circuit design.

Principally, there exist two different methods for the calculation of sensitivity functions: numerical and symbolic methods. Classical numerical techniques are the sensitivity network approach [160] and the adjoint network approach [161]. The first method is advantageous in investigating the sensitivities of many circuit outputs with respect to one circuit parameter, whereas the second one is best suited for the determination of the sen-

sitivities of one output to many parameters. In [162] the sensitivities of poles and zeros in linear systems are considered numerically. However, sensitivity analysis typically needs to be applied several times within iterative tasks (see summary given above) and numerical methods require a complete new solution of the system matrix at each new frequency and parameter point.

Symbolic analysis expresses the network function in terms of the frequency variable s and the circuit parameters X. Consequently, it is an appropriate technique for sensitivity analysis, especially when the behaviour at many frequency points and parameter values should be investigated [101]. In direct symbolic approaches, e.g. [163, 164, 165, 166], the sensitivities can be extracted as the coefficients of the terms in the network function containing the respective parameter. However, the drawback of these procedures is that the number of terms grows exponentially with circuit complexity. As a consequence, the direct symbolic approach is inadequate for sensitivity analysis of today's large scale systems [164].

In the last chapter, the hierarchical symbolic approach of [89] has been presented which describes the network function by a SOE. The benefit of the SOE approach is that the computational expense depends only linearly up to quadratically on circuit complexity. This makes the SOE method best suited for sensitivity analysis of large-scale electronic circuits.

5.2 Sensitivity Analysis based on Sequence of Expressions

The SOE approach has originally been used for sensitivity analysis in [167, 168, 169]. The principle of these techniques is illustrated with the following example:

$$
\begin{aligned}
H_1 &= G_1 & \frac{\partial H_1}{\partial G_1} &= 1 \\
H_2 &= G_2 & \frac{\partial H_2}{\partial G_1} &= 0 \\
H_3 &= H_1 + H_2 & \frac{\partial H_3}{\partial G_1} &= \sum_{j<3} \frac{\partial H_3}{\partial H_j}\frac{\partial H_j}{\partial G_1} = \frac{\partial H_3}{\partial H_1}\frac{\partial H_1}{\partial G_1} = 1 \cdot \frac{\partial H_1}{\partial G_1} \\
H_4 &= G_3 & \frac{\partial H_4}{\partial G_1} &= 0 \\
H_5 &= G_4 & \frac{\partial H_5}{\partial G_1} &= 0 \\
H_6 &= \frac{H_3 H_4}{H_3 + H_4} & \frac{\partial H_6}{\partial G_1} &= \sum_{j<6} \frac{\partial H_6}{\partial H_j}\frac{\partial H_j}{\partial G_1} = \frac{\partial H_6}{\partial H_3}\frac{\partial H_3}{\partial G_1} = \frac{H_4^2}{(H_3 + H_4)^2} \cdot \frac{\partial H_3}{\partial G_1} \\
H = H_7 &= H_6 + H_5 & \frac{\partial H_7}{\partial G_1} &= \sum_{j<7} \frac{\partial H_7}{\partial H_j}\frac{\partial H_j}{\partial G_1} = \frac{\partial H_7}{\partial H_6}\frac{\partial H_6}{\partial G_1} = 1 \cdot \frac{\partial H_6}{\partial G_1}
\end{aligned}
\tag{5.3}
$$

On the left hand side of this equation set the SOE for the output admittance $H = Y_4$ of the resistive ladder circuit of Figure 4.4 (Page 70) is shown. On the right hand side the sensitivity of H with respect to the admittance G_1 is calculated. Sensitivity analysis is performed sequentially by computing for each expression H_i of the nominal SOE the partial derivative $\frac{\partial H_i}{\partial G_1}$, thereby generating a sensitivity SOE with $\frac{\partial H_i}{\partial G_1}$ as the new expressions. Generally, within the SOE for the nominal behaviour there exist two different types of expressions:

- **type 1 expressions** directly depend on a circuit parameter x.

  Examples: $H_1 = G_1$, $H_2 = G_2$, $H_4 = G_3$ and $H_5 = G_4$ in (5.3).

- **type 2 expressions** depend on predecessing expressions of the SOE.

  Examples: $H_3 = H_1 + H_2$, $H_6 = \frac{H_3 H_4}{H_3 + H_4}$ and $H_7 = H_6 + H_5$ in (5.3).

To each type of nominal SOE expression there corresponds a respective type of sensitivity expression in the sensitivity SOE:

sensitivity of type 1 expressions:
$$\frac{\partial H_i}{\partial x} = \begin{cases} 1 & : \quad H_i = x \\ 0 & : \quad H_i \neq x \end{cases} \tag{5.4}$$

sensitivity of type 2 expressions:
$$\frac{\partial H_i}{\partial x} = \sum_{j<i} \frac{\partial H_i}{\partial H_j}\frac{\partial H_j}{\partial x} \tag{5.5}$$

where the summing condition $j < i$ in (5.5) is a consequence of the fact that within the nominal SOE there exist only backward dependencies

$$\frac{\partial H_i}{\partial H_j} = 0 \qquad \text{for} \quad j \geq i. \tag{5.6}$$

Using equations (5.4) and (5.5) a separate sensitivity SOE is generated for each circuit parameter x.

5.2.1 Implementational Aspects

The implementation of SOE sensitivity analysis is guided by the following two observations [168]:

- Typically, only a few equations in a SOE depend on a distinct parameter.

  Example: in equation set (5.3) only H_1, H_3, H_6 and H_7 depend on G_1.

- the summing in equation (5.5) usually generates only a few product terms $\frac{\partial H_i}{\partial H_j}\frac{\partial H_j}{\partial x}$.

  Example: in (5.3) the calculation of $\frac{\partial H_7}{\partial G_1}$ requires only one product term because H_7 depends merely on H_6 and H_5 where $\frac{\partial H_5}{\partial G_1}$ is zero.

The above observations have been used in [168, 169] for an effective implementation of SOE sensitivity analysis by performing calculations only on those equations which are influenced by a distinct circuit parameter. An expressions graph has been defined in [168] to find a minimal set of SOE sensitivity equations.

Definition 5.1 (Expression Graph) *The expression graph of a SOE is a directed graph where*

- *each SOE expression H_i is represented by a* **node** *N_i and*

- *the dependencies between the SOE expressions are represented by* **edges***: there exists a directed* **edge** *(j, i) from node N_j to node N_i if and only if expression H_i depends explicitly on H_j.*

The expression graph of the nominal SOE of equation (5.3) is illustrated in Figure 5.1. Since there exists only backward dependencies within the SOE (see equation 5.6), the

$$
\begin{aligned}
H = H_7 &= H_6 + H_5 \\
H_6 &= \frac{H_3 H_4}{H_3 + H_4} \\
H_5 &= G_4 \\
H_4 &= G_3 \\
H_3 &= H_1 + H_2 \\
H_2 &= G_2 \\
H_1 &= G_1
\end{aligned}
$$

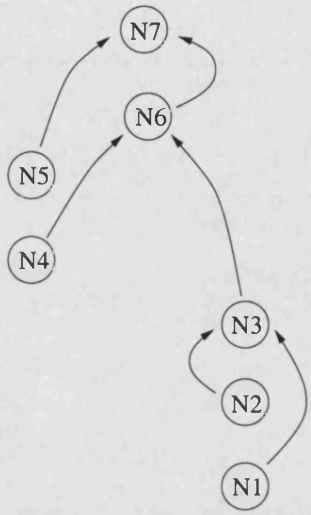


Figure 5.1: SOE and its corresponding expression graph

expression graph contains no loops. The expression graph can therefore be classified as a Directed Acyclic Graph (DAG). Before using the expression graph for sensitivity analysis it is worth noting the following correspondences between the DAG and the SOE:

- to each **leaf node** of the DAG there corresponds a **type 1 expression** in the SOE. This expression directly refers to a circuit parameter.
- to each **non-leaf node** of the DAG there corresponds a **type 2 expression** in the SOE.
- the **root** of the DAG corresponds to the **network function** H calculated by the SOE.

Sensitivity calculation with respect to a parameter x follows the DAG paths bottom-up, starting at the leaf representing the circuit parameter x and ending at the root representing

the network function H [168]. In terms of the DAG, the sensitivity equations (5.4) and (5.5) become

$$\text{for leaf node } N_x \text{ which refers to } x \quad : \quad \frac{\partial H_x}{x} = 1 \tag{5.7}$$

$$\begin{array}{l}\text{for all non-leaf nodes } N_i \text{ which lie} \\ \text{on a path from leaf } N_x \text{ to the root}\end{array} \quad : \quad \frac{\partial H_i}{\partial x} = \sum_{(j,i)\in DAG} \frac{\partial H_i}{\partial H_j}\frac{\partial H_j}{\partial x}. \tag{5.8}$$

Thereby the summing condition is a consequence of the fact that the DAG edges represent the explicit dependencies between SOE expressions. The next step is to define weights for the DAG nodes and edges as follows

$$\text{node weight for } N_i \quad : \quad n_i := \frac{\partial H_i}{\partial x}, \tag{5.9}$$

$$\text{edge weight for edge } (j,i) \quad : \quad w(j,i) := \frac{\partial H_i}{\partial H_j}. \tag{5.10}$$

These weights can be used to formulate the sensitivity equations (5.7) and (5.8) completely in terms of the DAG:

$$\text{for leaf node } N_x \text{ which refers to } x \quad : \quad n_x = 1 \tag{5.11}$$

$$\begin{array}{l}\text{for all non-leaf nodes } N_i \text{ which lie} \\ \text{on a path from leaf } N_x \text{ to the root}\end{array} \quad : \quad n_i = \sum_{(j,i)\in DAG} w(j,i)\, n_j. \tag{5.12}$$

This shows that SOE sensitivity analysis is equivalent to a corresponding signal-graph problem [170]: the corresponding signal-graph is the DAG with edge weights according to equation (5.10). The input nodes are the DAG leaves. The weight of the leaf x is set to 1 (equation 5.11) and the weights of all other leaves are set to zero. The sensitivity is obtained as the weight of the root (root=output node), after the node weights have been propagated along the paths from the leaf to the root according to equation (5.12). Using the weighted DAG, only those SOE expressions which depend on x are examined thereby reducing the number of terms during sensitivity analysis. For example, the weighted DAG equations for the calculation of $\frac{\partial H}{\partial G_1}$ are

$$
\begin{aligned}
n_1 &= 1 \\
n_3 &= w(1,3)\, n_1 \\
n_6 &= w(3,6)\, n_3 \\
\frac{\partial H}{\partial G_1} = n_7 &= w(6,7)\, n_6 \\
&\text{where} \quad w(1,3) = w(6,7) = 1, \; w(3,6) = \frac{H_4^2}{(H_3 + H_4)^2}.
\end{aligned}
\tag{5.13}
$$

The sensitivity equations become very compact using the DAG.

5.2.2 Computational Aspects

Using the procedure described above, a term $w(j,i)\, n_j$ is generated for each edge crossed while traversing, bottom-up, the paths which start at the DAG leaf x and end at the DAG root H. Consequently, the expense for sensitivity analysis with respect to a single parameter x is proportional to the lengths of paths[1] from the leaf x to the root. Based on this observation, the expense of sensitivity calculation considering a single parameter x was reduced in [169] as follows: during the SOE generation with SCAPP, MNA-matrix rows containing the parameter x are eliminated only when the highest hierarchical level is reached. As a result, the parameter x occurs only in the last few SOE expressions and the lengths of the DAG paths from x to H are reduced.

However, this procedure causes other MNA-matrix rows to be eliminated more early and produces longer DAG paths for other parameters. As a consequence, this method doesn't help in the multi-parameter case, and the simulation expense grows significantly with circuit size when the sensitivity of many or all parameters is required [169]. This method is therefore not optimal for the use within the design process where typically the influence of all parameters needs to be considered (see for example equation 5.1).

[1]The length of a path is defined as the number of path edges.

5.3 Balanced Symbolic Sensitivity Analysis

The goal of this section is to develop a method which reduces the expense of multi-parameter SOE sensitivity analysis. To achieve this goal, the average lengths of leaf-root DAG paths needs to be minimized. Theoretically, this task can be formulated as an graph optimization problem with the following cost function and degrees of freedom:

- cost function: average length $\overline{l_{DAG}}$ of DAG leaf-root paths.
- degrees of freedom:
  1. Gaussian elimination sequence of the circuit variables,
  2. structure of Binary Partition Tree (BPT) which models the hierarchical analysis (see Section 4.2).

Unfortunately, both the number of different elimination sequences and the number of different BPTs is very large. Given n_v circuit variables and a decomposition of the network into m terminal blocks, the first one grows as $n_v!$ and the second one even faster than $m!$. For this reason, exhaustively trying all elimination sequences and BPT structures would require an unrealistically huge amount of computing time. The solution of this dilemma is to apply a heuristic which has been introduced in [171]. The heuristic is easy to handle and yields a near optimal solution. The basis for the heuristic is the structural similarity of the BPT and the expression graph DAG. This similarity will be explained in the next section.

5.3.1 Binary Partition Tree and Expression Graph

An interesting observation is that the structures of the BPT and the DAG are strongly related. For illustration, the resistive ladder circuit of Figure 5.2 is considered. The terminal blocks for the hierarchical symbolic analysis are given as the encircled partitions. In the case of a 2-stage ladder circuit ($K = 2$) the BPT, the resulting SOE for the output

admittance and the corresponding DAG are illustrated in Figure 5.3.

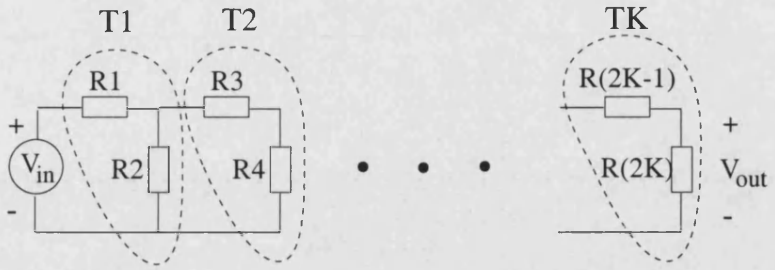


Figure 5.2: Resistive ladder circuit and partitioning

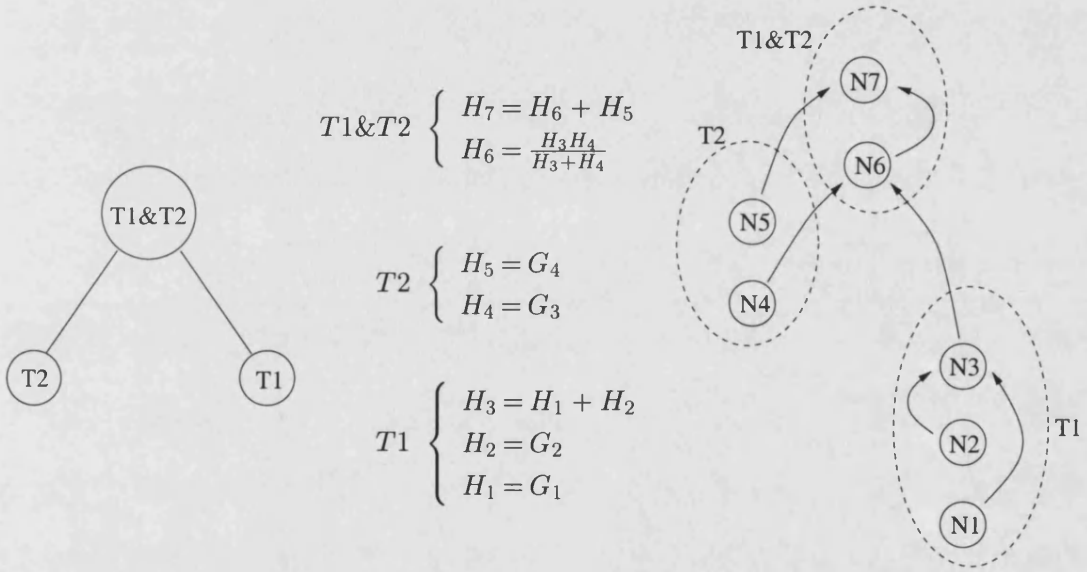


Figure 5.3: BPT, SOE and DAG (2-stage ladder circuit, output admittance)

Node Correspondence

As illustrated in Figure 5.3, each node N^{BPT} of the BPT corresponds to a set of equations in the SOE. These equations were generated while calculating the RMNA matrix of the circuit block which is represented by the node N^{BPT}. As a result, to each node N^{BPT} there corresponds a set of nodes in the DAG. This correspondence is indicated in the DAG

of Figure 5.3 by encircled node sets and is summarized in Table 5.1. In the following, the

BPT nodes	corresponding DAG nodes
$T1$	$\{N1, N2, N3\}$
$T2$	$\{N4, N5\}$
$T1\&T2$	$\{N6, N7\}$

Table 5.1: Node correspondence between BPT and DAG (2-stage ladder circuit)

correspondence between a BPT node N^{BPT} and a DAG node N^{DAG} is formally indicated by the mathematical relation $N^{BPT} \sim N^{DAG}$.

Edge Correspondence

The BPT edges reflect the dependency between different hierarchical blocks. The DAG edges represent the mathematical dependencies between different SOE expressions. Consequently, a correspondence between the BPT and DAG edges can be expected. This correspondence is the content of the following theorem.

Theorem 5.1 *Let N_j^{DAG}, N_i^{DAG} be two DAG nodes and N_j^{BPT}, N_i^{BPT} the corresponding BPT nodes: $N_j^{BPT} \sim N_j^{DAG}$, $N_i^{BPT} \sim N_i^{DAG}$. There exists an edge $(j,i)_{DAG}$ between the DAG nodes N_j^{DAG} and N_i^{DAG}* **only if**

- *either $N_j^{BPT} = N_i^{BPT}$,*
- *or there exists an edge in the BPT between the corresponding BPT nodes N_j^{BPT} and N_i^{BPT}. This edge is called the corresponding BPT edge $(j,i)_{BPT} \sim (j,i)_{DAG}$.*

The reason behind this theorem is that when expression H_i depends on H_j, then

- either the two expressions H_i and H_j are calculated within the RMNA analysis of the same circuit block ($N_j^{BPT} = N_i^{BPT}$),

- or the circuit block whose RMNA analysis has generated expression H_j, is a constituting subcircuit of the circuit block whose RMNA analysis has generated expression H_i. In such a case, N_j^{BPT} refers to a constituting subcircuit of the circuit block which is represented by N_i^{BPT}.

To illustrate this situation, the edge correspondence of the BPT and DAG of Figure 5.3 is summarized in Table 5.2. The DAG edges $(N6, N7)$, $(N2, N3)$ and $(N1, N3)$ have no

BPT edge	corresponding DAG edge(s)
$(T2, T1\&T2)$	$(N5, N7)$ $(N6, N7)$
$(T1, T1\&T2)$	$(N3, N6)$
–	$(N6, N7)$ $(N2, N3)$ $(N1, N3)$

Table 5.2: Edge correspondence between BPT and DAG (2-stage ladder circuit)

corresponding BPT edge because they refer to SOE dependencies between expressions generated during the RMNA analysis of the same circuit block (first case of Theorem 5.1).

Path Correspondence

As a consequence of the node-node and edge-edge correspondence between the BPT and the DAG, there is a close similarity between the leaf-root paths of the BPT and of the DAG. Given any arbitrary leaf-root path $(N_{i_1}^{DAG}, \ldots, N_{i_k}^{DAG})$ in the DAG, the corresponding BPT path is defined as that path of the BPT which is generated by visiting the corresponding BPT nodes $N_{i_1}^{BPT}, \ldots, N_{i_k}^{BPT}$, $N_{i_j}^{BPT} \sim N_{i_j}^{DAG} \quad j = 1 \ldots k$. Thereby Theorem 5.1 guarantees that within the BPT node sequence $N_{i_1}^{BPT}, \ldots, N_{i_k}^{BPT}$ two successive nodes

$N_{i_j}^{BPT}, N_{i_{j+1}}^{BPT}$ are either identical ($N_{i_j}^{BPT} = N_{i_{j+1}}^{BPT}$, case 1 of Theorem 5.1) or connected by an BPT edge $(j, j+1)_{BPT}$ (case 2 of Theorem 5.1). Therefore the existence of the BPT path visiting the nodes $N_{i_1}^{BPT}, \ldots, N_{i_k}^{BPT}$ is guaranteed. In the case of the 2-stage ladder network, the BPT/DAG path correspondence is shown in Table 5.3.

BPT paths	DAG paths
$(T2, T1\&T2)$	$(N5, N7)$ $(N4, N6, N7)$
$(T1, T1\&T2)$	$(N2, N3, N6, N7)$ $(N1, N3, N6, N7)$

Table 5.3: Path correspondence between BPT and DAG (2-stage ladder circuit)

5.3.2 Balanced Binary Partition Tree

Using the results developed in the last section, the DAG path lengths can be approximately minimized by applying a heuristic. Because of the strong relationship between BPT and DAG paths, the following assumption seems to be sensible: *"The longer a path of the BPT, the longer are the corresponding paths in the DAG."* For justification of this assumption, a 4-stage ladder circuit[2] according Figure 5.2 is considered. The SOE describing the output admittance $Y_{out} = H$ of this circuit is

[2]A 2-stage ladder circuit is a too simple example in this situation because the respective BPT (Figure 5.3) contains only paths of length 1.

$$
\begin{aligned}
H_1 &= G_1 \\
H_2 &= G_2 \\
H_3 &= H_1 + H_2 \\
H_4 &= G_3 \\
H_5 &= G_4 \\
H_6 &= \frac{H_3 H_4}{H_3 + H_4} \\
H_7 &= H_6 + H_5 \\
H_8 &= G_5 \\
H_9 &= G_6 \\
H_{10} &= \frac{H_7 H_8}{H_7 + H_8} \\
H_{11} &= H_{10} + H_9 \\
H_{12} &= G_7 \\
H_{13} &= G_8 \\
H_{14} &= \frac{H_{11} H_{12}}{H_{11} + H_{12}} \\
H = H_{15} &= H_{14} + H_{13}
\end{aligned}
\tag{5.14}
$$

The BPT which has been used for hierarchical analysis and the DAG modelling the outcoming SOE are illustrated in Figure 5.4. The length of the corresponding BPT and DAG paths are compared in Table 5.4. These results verify the above assumption. Moreover, by inspecting several BPTs and DAGs it has been found out that in most cases the length

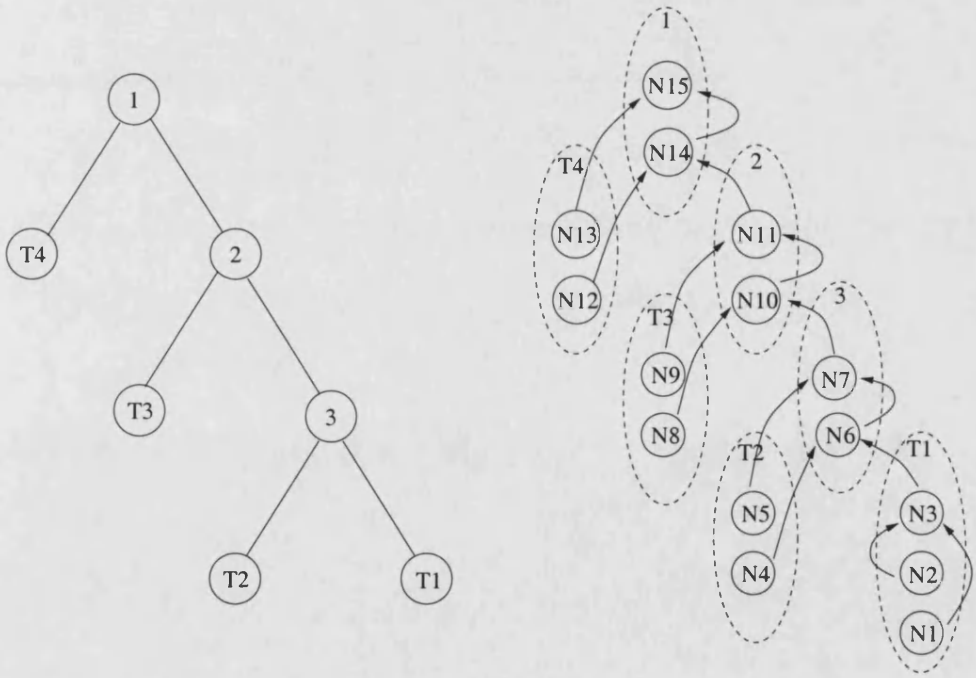


Figure 5.4: BPT and DAG of 4-stage ladder circuit

BPT path	path length	DAG path	path length
$(T4, 1)$	1	(N_{13}, N_{15})	1
		(N_{12}, N_{14}, N_{15})	2
$(T3, 2, 1)$	2	$(N_9, N_{11}, N_{14}, N_{15})$	3
		$(N_8, N_{10}, N_{11}, N_{14}, N_{15})$	4
$(T2, 3, 2, 1)$	3	$(N_5, N_7, N_{10}, N_{11}, N_{14}, N_{15})$	5
		$(N_4, N_6, N_7, N_{10}, N_{11}, N_{14}, N_{15})$	6
$(T1, 3, 2, 1)$	3	$(N_2, N_3, N_6, N_7, N_{10}, N_{11}, N_{14}, N_{15})$	7
		$(N_1, N_3, N_6, N_7, N_{10}, N_{11}, N_{14}, N_{15})$	7

Table 5.4: Comparison of BPT and DAG paths length

of corresponding BPT and DAG paths are approximately proportional to each other

$$l_{BPT} \propto l_{DAG}. \tag{5.15}$$

Because of this situation, the following heuristic will be applied:

BPT heuristic 1: *A minimization of the average leaf-root path length $\overline{l_{DAG}}$ in the DAG is achieved by minimizing the average leaf-root path length $\overline{l_{BPT}}$ in the BPT.*

This heuristic transforms the complicated DAG optimization problem into an easy-to-solve BPT problem. Normally, the circuit is partitioned either by the design process or by automatic decomposition into m small subcircuits. These m subcircuits are chosen as terminal blocks for the hierarchical symbolic analysis. Then, the BPT problem can be formulated as follows: *which hierarchical BPT with m leaves causes the average length $\overline{l_{BPT}}$ of leaf-root paths to be minimal?* As known from graph theory the solution is a maximally balanced BPT. Consequently, the BPT heuristic can be finally formulated as follows [171]:

BPT heuristic 2: *Let a circuit be partitioned into m subcircuits. A minimization of computational expense of SOE sensitivity analysis is achieved by*

choosing a **maximally balanced BPT** *with m leaves for the hierarchical SOE generation with SCAPP.*

5.3.3 Computational Benefit of the Balanced Approach

For the estimation of the computational benefit of the balanced strategy, the two extremes according to Figure 5.5 are considered: a totally unbalanced BPT and a maximally balanced BPT. The respective average lengths of the leaf-root paths are given by

Figure 5.5: Totally unbalanced (left) and maximally balanced (right) BPT

$$\overline{l_{BPT}}(\text{unbalanced}) = \frac{m+1}{2} - \frac{1}{m}, \tag{5.16}$$

$$\overline{l_{BPT}}(\text{balanced}) = \log_2 m. \tag{5.17}$$

In the case of practical circuits (sparse systems, number of circuit elements connected to a distinct node is limited by a constant) the number of expressions required for the calculation of the RMNA-matrix of each block is bounded by a constant K_1 [89]. Consequently, the number of arithmetic operations required for differentiating the expressions within a single RMNA matrix is also limited, say by a constant K_2. Then, using (5.16) and (5.17) the cost for all-parameter sensitivity analysis is given by

$$\begin{aligned} \text{cost}(\text{unbalanced}) &\approx N_X\, K_2\, \overline{l_{BPT}}(\text{unbalanced}) \\ &= N_X\, K_2 \left(\frac{m+1}{2} - \frac{1}{m}\right), \end{aligned} \tag{5.18}$$

$$\begin{aligned} \text{cost(balanced)} &\approx N_X\, K_2\, \overline{l_{BPT}}(\text{balanced}) \\ &= N_X\, K_2\, \log_2 m. \end{aligned} \tag{5.19}$$

A common measure for circuit size is the number of circuit nodes n. In practical circuits, the number of terminal blocks m and the number of circuit parameters N_X are both proportional to the number of circuit nodes:

$$N_X = O(n), \tag{5.20}$$

$$m = O(n). \tag{5.21}$$

Then, (5.18) and (5.19) become

$$\text{cost(unbalanced)} = O(n^2) \tag{5.22}$$

$$\text{cost(balanced)} = O(n \log_2 n). \tag{5.23}$$

This shows that the balanced strategy is expected to reduce considerably the cost for sensitivity analysis. The acceleration factor is $n/\log_2 n$. The balanced strategy therefore reduces computation time especially in the case of large scale circuits.

5.3.4 Experimental Results

To examine the benefit of the balanced approach, practical circuit examples will be analysed. The number of arithmetic operations required for sensitivity analysis will be examined and the estimations of the last section will be verified. Both, the SOE generation according SCAPP [89] and the SOE sensitivity analysis have been implemented in Maple V [172]. Herein, before SOE generation starts, different BPT structures can be entered by the user. SOE sensitivity analysis is a two step procedure:

1. calculation of edge weights according to equation (5.10),

2. successive calculation of sensitivities (for each circuit parameter separately) according to the DAG equations (5.11) and (5.12) by using the edge weights calculated in step 1.

Ladder Networks

The ladder network of Figure 5.2 is an important configuration for analogue filter applications. This circuit is well suited for theoretical examinations because of the simplicity of its topological structure. Firstly, SOE sensitivity analysis according step 1 and 2 are illustrated with help of a 2-stage ladder circuit. Figure 5.6 shows the SOE describing its transfer function $H(s) = V_{out}(s)/V_{in}(s)$ and the corresponding DAG. The sensitivities of the transfer function with respect to all 4 parameters should be calculated. The edge weights obtained by step 1 are shown in Figure 5.7. The sensitivity equations according

$$
\begin{aligned}
H = H_{11} &= -\frac{H_9}{H_{10}} \\
H_{10} &= H_6 + H_8 H_3 \\
H_9 &= H_8 H_1 \\
H_8 &= -\frac{H_3}{H_7} \\
H_7 &= H_3 + H_5 \\
H_6 &= H_3 + H_4 \\
H_5 &= H_1 + H_2 \\
H_4 &= x_4 = G_4 \\
H_3 &= x_3 = G_3 \\
H_2 &= x_2 = G_2 \\
H_1 &= x_1 = G_1
\end{aligned}
$$

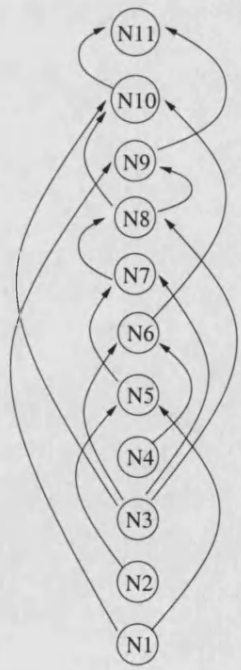


Figure 5.6: SOE and DAG of the transfer function of the 2-stage ladder circuit

to step 2 (one set of equations for each circuit parameter) are shown in Figure 5.8. The

$$
\begin{aligned}
w(1,5) &= \frac{\partial H_5}{\partial H_1} = 1, & w(2,5) &= \tfrac{\partial H_5}{\partial H_2} = 1, \\
w(3,6) &= \frac{\partial H_6}{\partial H_3} = 1, & w(4,6) &= \tfrac{\partial H_6}{\partial H_4} = 1, \\
w(3,7) &= \frac{\partial H_7}{\partial H_3} = 1, & w(5,7) &= \tfrac{\partial H_7}{\partial H_5} = 1, \\
w(3,8) &= \frac{\partial H_8}{\partial H_3} = -\frac{1}{H_7}, & w(7,8) &= \tfrac{\partial H_8}{\partial H_7} = \tfrac{H_3}{{H_7}^2}, \\
w(1,9) &= \frac{\partial H_9}{\partial H_1} = H_8, & w(8,9) &= \tfrac{\partial H_9}{\partial H_8} = H_1, \\
w(3,10) &= \frac{\partial H_{10}}{\partial H_3} = H_8, & w(6,10) &= \tfrac{\partial H_{10}}{\partial H_6} = 1, \qquad w(8,10) = \frac{\partial H_{10}}{\partial H_8} = H_3, \\
w(9,11) &= \frac{\partial H_{11}}{\partial H_9} = -\frac{1}{H_{10}}, & w(10,11) &= \tfrac{\partial H_{11}}{\partial H_{10}} = \tfrac{H_9}{{H_{10}}^2}.
\end{aligned}
$$

Figure 5.7: Edge weights for the DAG of Figure 5.6

quality of SOE sensitivity analysis will be measured based on the required number of arithmetic operations. In the above example, the calculation of the edge weights (Figure 5.7) requires 6 multiplications and 2 additions[3]. Evaluating the node weight equations (Figure 5.8) requires 4, 4, 4, and 0 multiplications[4] and 2, 1, 4, and 0 additions for the parameters G_1, G_2, G_3 and G_4 respectively. Together with the edge weight equations, the overall cost is 18 multiplications and 9 additions.

Now, ladder circuits with different numbers of stages are examined. A totally unbalanced and a maximally balanced BPT are used for SOE analysis. The number of additions and multiplications required for sensitivity analysis are shown in Table 5.5 and 5.6 respectively. An illustration of the computational expense in dependence on the circuit complexity is given in Figure 5.9. The comparison shows that the balanced partitioning

[3]Subtractions and negations are counted as additions, divisions as multiplications.

[4]When a node or edge weight is identical 1, the multiplication $w(j,i)\,n_i$ becomes trivial. In such a case, the respective multiplication is not counted, e.g. $n_5 = w(1,5)\,n_1$ and $n_8 = w(7,8)\,n_7$ in the sensitivity equations with respect to G_1.

sensitivity w.r.t. G_1:

$$
\begin{aligned}
n_1 &= 1 \\
n_5 &= w(1,5)\, n_1 \\
n_7 &= w(5,7)\, n_5 \\
n_8 &= w(7,8)\, n_7 \\
n_9 &= w(8,9)\, n_8 + w(1,9)\, n_1 \\
n_{10} &= w(8,10)\, n_8 \\
n_{11} &= w(9,11)\, n_9 + w(10,11)\, n_{10}
\end{aligned}
$$

sensitivity w.r.t. G_2:

$$
\begin{aligned}
n_2 &= 1 \\
n_5 &= w(2,5)\, n_2 \\
n_7 &= w(5,7)\, n_5 \\
n_8 &= w(7,8)\, n_7 \\
n_9 &= w(8,9)\, n_8 \\
n_{10} &= w(8,10)\, n_8 \\
n_{11} &= w(9,11)\, n_9 + w(10,11)\, n_{10}
\end{aligned}
$$

sensitivity w.r.t. G_3:

$$
\begin{aligned}
n_3 &= 1 \\
n_6 &= w(3,6)\, n_3 \\
n_7 &= w(5,7)\, n_5 \\
n_8 &= w(3,8)\, n_3 + w(7,8)\, n_7 \\
n_9 &= w(8,9)\, n_8 \\
n_{10} &= w(3,10)\, n_3 + w(6,10)\, n_6 + w(8,10)\, n_8 \\
n_{11} &= w(9,11)\, n_9 + w(10,11)\, n_{10}
\end{aligned}
$$

sensitivity w.r.t. G_4:

$$
\begin{aligned}
n_4 &= 1 \\
n_6 &= w(4,6)\, n_4 \\
n_{10} &= w(6,10)\, n_6 \\
n_{11} &= w(10,11)\, n_{10}
\end{aligned}
$$

Figure 5.8: Sensitivity (DAG node weight) equations for the SOE of Figure 5.6

strategy [171] causes a speed-up of sensitivity analysis. As expected, the speed-up increases with circuit size. In the case of a ladder circuit with 8 stages the gain in speed is a factor 2 compared to the unbalanced strategy (referring to the number of multiplications N_{mults}). For a ladder circuit with 128 stages the gain is a factor 11. By using intermediate BPT structures, the cost for sensitivity analysis is in between the results shown above. Generally, the more the BPT is balanced, the lower is the cost for sensitivity analysis.

The quality of the complexity estimations (equation 5.22 and 5.23) can be assessed by

number of stages	number of parameters N_X	unbalanced BPT N_{adds}	balanced BPT N_{adds}
2	4	9	9
4	8	61	44
8	16	261	141
16	32	1045	398
32	64	4149	1039
64	128	16501	2576
128	256	65781	6161

Table 5.5: Number of additions N_{adds} required for all-parameter sensitivity analysis of the ladder circuit

number of stages	number of parameters N_X	unbalanced BPT N_{mults}	balanced BPT N_{mults}
2	4	18	18
4	8	131	92
8	16	619	317
16	32	2651	938
32	64	10939	2527
64	128	44411	6404
128	256	178939	15561

Table 5.6: Number of multiplications N_{mults} required for all-parameter sensitivity analysis of the ladder circuit

calculating K_3 as follows:

$$K_3 = \begin{cases} \frac{N_{mults}}{n(n+1)/2} & : \text{ unbalanced BPT} \\ \frac{N_{mults}}{n \log_2 n} & : \text{ balanced BPT} \end{cases} \tag{5.24}$$

In the case when K_3 turns out to be approximately constant with increasing circuit size n, the estimations can be considered as good. Using the results of Table 5.6, the K_3 values shown in Table 5.7 are obtained. The calculated K_3 values are nearly constant for larger n ($n > 30$) which shows that the formula (5.22) and (5.23) are good estimations for the computational complexity of multi-parameter SOE sensitivity analysis.

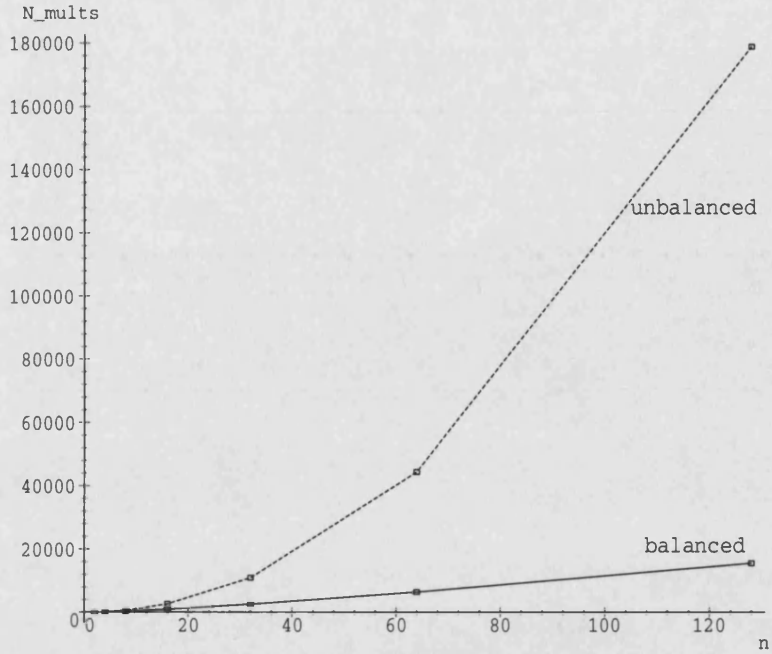


Figure 5.9: Number of multiplications N_{mults} required for all-parameter sensitivity analysis of the ladder circuit
horizontal axis: number of circuit nodes n
vertical axis: number of multiplications N_{mults}

number of circuit nodes n	unbalanced BPT K_3	balanced BPT K_3
3	3	3.8
5	8.7	7.9
9	13.7	11.1
17	17.3	13.5
33	19.5	15.2
65	20.7	16.4
129	21.3	17.2

Table 5.7: K_3 values according to equation (5.24) for different number of circuit nodes n

Large Scale Bandpass Filter

As a second example, the bandpass circuit of Figure 5.10 is analysed. This bandpass filter is a common benchmark circuit for symbolic analysis procedures tackling large scale networks. The encircled partitions $T1, \ldots, T13$ in Figure 5.10 are the terminal blocks for the hierarchical SOE analysis. The two different BPTs, a totally unbalanced and a strongly

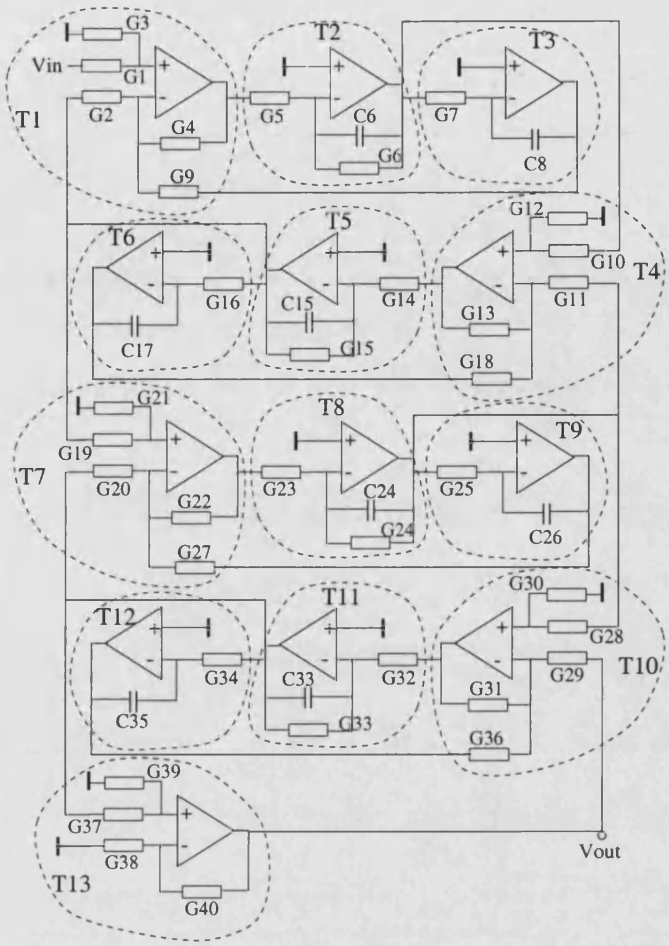


Figure 5.10: Bandpass filter and circuit partitioning

balanced BPT, used for this task are shown in Figure 5.11. The resulting computational expense for multi-parameter sensitivity analysis is shown in Table 5.8. The speed-up when

BPT used	N_{adds}	N_{mults}
a) totally unbalanced	657	1828
b) strongly balanced	258	822

Table 5.8: Expense of multi-parameter sensitivity analysis of the bandpass circuit

using the strongly balanced BPT instead of the unbalanced BPT is 2.2 in terms of number of multiplications. This example shows that it is not required to have a totally balanced BPT. A strongly balanced BPT also helps in accelerating sensitivity analysis.

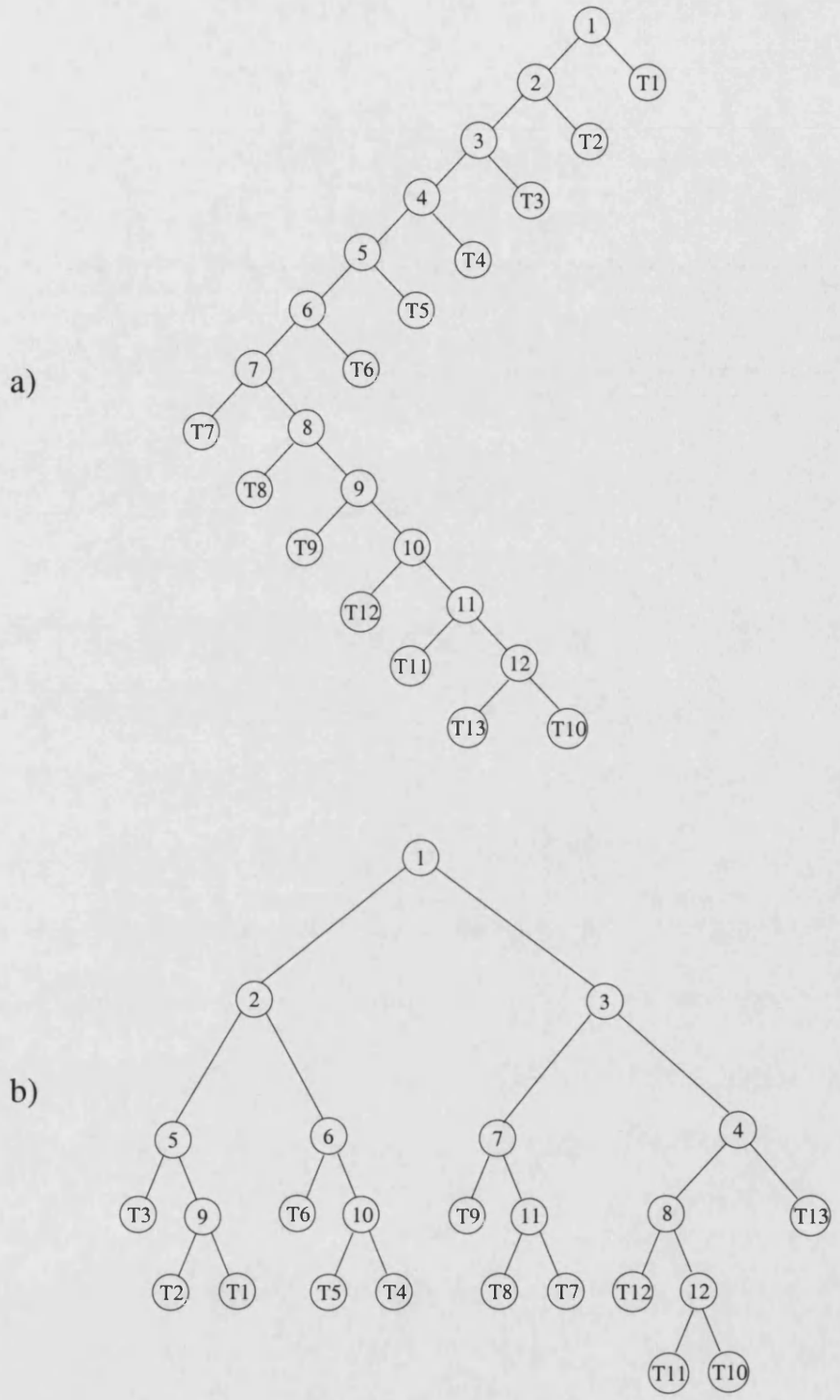


Figure 5.11: Bandpass circuit BPTs used for SOE generation
a) totally unbalanced BPT
b) strongly balanced BPT

5.3.5 Fault Simulation Applications

So far, the balanced partitioning strategy has been applied to accelerate differential sensitivity analysis. In this section it is shown that the balanced strategy can also be applied to analyse the effect of large change deviations of parameters, e.g. in parametric fault simulation.

Symbolic Simulation of Parametric Faults

Symbolic analysis has been applied to simulation of single and double parametric faults of linear analogue circuits [61][5]. Using the SOE approach of [89] the authors achieved an acceleration of factor 15 compared to classical numerical simulation. However, due to the great number of element parameters in large scale circuits, the simulation needs to be repeated several times and the overall computational cost is still very high. The expense of SOE fault simulation can be reduced taking into account the following two items:

1. Only a part of the SOE is influenced by the deviation of a distinct parameter. During fault simulation only this part needs to be re-evaluated using the nominal values for the other expressions in the SOE.

2. The smaller the number of expressions influenced by a parameter, the faster the fault simulation can be performed. Consequently the number of expressions influenced by a parameter needs to be minimized.

Obviously, the techniques developed in the last section can help to solve these tasks. Since the DAG represents dependencies between the SOE expressions it can be used to find the subset of SOE expressions which needs to be re-evaluated in the case of a parametric fault. Concerning the second item, using a balanced BPT during the hierarchical SOE gener-

[5] In this reference, the fault simulations were required to construct a fault dictionary for a fault diagnosis procedure. This diagnosis procedure is an alternative one to that investigated in Chapter 3.

ation will minimize the leaf-root path length in the DAG. Consequently, the number of SOE expressions which are influenced by a single parameter is reduced. Using these ideas, a new SOE parametric fault simulation [173] has been implemented in MAPLE V [172]. The flow-chart of this algorithm is illustrated in Figure 5.12. After the hierarchical circuit

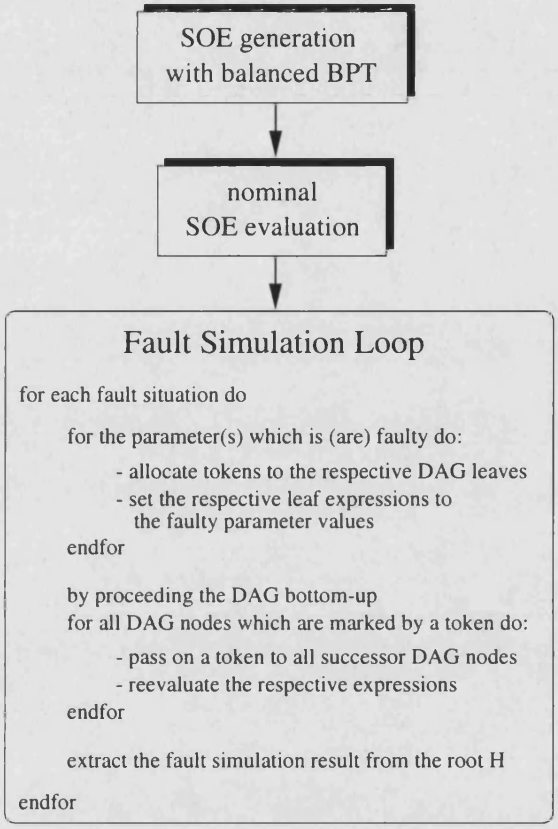


Figure 5.12: SOE fault simulation procedure

partitioning and the construction of a maximally balanced BPT the symbolic analysis procedure of [89] is applied. Firstly, the outcoming SOE is used for the nominal analysis. During fault simulation, the faulty parameter values are assigned to the respective leaf expressions. The part of the SOE which is influenced by the respective parametric deviations is discovered by a token passing procedure and re-evaluated. Those expressions which are not influenced by the fault remain at their nominal value. For each fault, the

result of fault simulation can be extracted as the expression value of the root H.

Results

To examine the benefit of the balanced strategy for fault simulation, the ladder circuit of Figure 5.2 is analysed. Fault simulations of the transfer function are performed by three different methods:

method 1: directly applying SOE analysis without token passing [61]

method 2: SOE analysis using DAG and token passing but totally unbalanced BPT

method 3: SOE analysis using DAG, token passing and maximally balanced BPT [173]

Table 5.9 and Table 5.10 show the computational expense[6] of these methods for single fault and double fault simulation respectively. The results demonstrate that the combination

number of stages	method 1 N_{mults}	method 2 N_{mults}	method 3 N_{mults}
2	16	13	13
4	128	77	56
8	640	349	187
16	2816	1469	542
32	11776	6043	1441
64	48128	24317	3620
128	194560	97789	8743

Table 5.9: Sum of computational expense for simulating all parametric single faults of the ladder circuit

of the balanced partitioning strategy with SOE analysis significantly reduces computation time for parametric fault simulation. Moreover, the acceleration increases with circuit complexity. In the case of single fault simulation of a 4-stage circuit the balanced approach

[6]The computational cost is measured in terms of the number of multiplications performed during fault simulation. The number of additions are similar.

number of stages	method 1 N_{mults}	method 2 N_{mults}	method 3 N_{mults}
2	24	24	24
4	448	370	274
8	4800	3582	2118
16	43648	30838	13390
32	370944	254694	75294
64	3056128	2067910	393022

Table 5.10: Sum of computational expense of double fault simulation of the ladder circuit

is 2 times faster than the direct SOE analysis. For a circuit with 128 stages the acceleration is a factor 20.

As a second example the bandpass filter illustrated in Figure 5.10 is considered. The fault simulations are carried out with respect to deviations of all 44 circuit parameters. The required number of multiplications are shown in Table 5.11. Using the balanced

analysis	method 1 N_{mults}	method 2 N_{mults}	method 3 N_{mults}
1-fault	2816	1080	518
2-fault	60544	31851	17075

Table 5.11: Sum of computational expense of single and double fault simulation of the bandpass filter

approach an acceleration by a factor of 5 is achieved for single fault simulation. In [61] it was shown that the direct SOE approach is about 15 times faster than a numerical analysis for fault simulation. The results presented in this section show that this advantage can be improved by an additional factor of 20 for large circuits using a balanced BPT. This makes the combination of the SOE method with the balanced approach very efficient for parametric fault simulation of large scale linear circuits.

5.3.6 Conclusion

A method for accelerating symbolic sensitivity calculations has been presented. Given any arbitrary circuit decomposition the subcircuits are recombined in a maximally balanced manner during hierarchical symbolic analysis. This results in reduced expense compared to previous symbolic approaches for differential sensitivity analysis and fault simulations. The speed up achieved by the new method increases with circuit size and is in the order of 2 to 20 for the examples presented.

5.3.7 Comparison of Fault Simulation Techniques

An interesting question is how the proposed symbolic fault simulation method compares with numerical techniques. To answer this question, it is useful to recall some of the major criteria for the evaluation of fault simulation procedures:

- **computational expense** is the major criterion because fault simulation usually requires the simulation of circuit behaviour under many different fault conditions. This causes huges computation costs, especially for larger circuits. Two sorts of costs need to be considered: the cost of preparing and the cost of performing the fault simulations. The former cost is related to fault modelling, to generation and characterisation of behavioural models (within hierarchical numerical approaches) or to the generation of symbolic expressions (within symbolic techniques). The latter cost refers to the CPU-time required during fault simulation.

- **precision**.

- **applicability** to linear, non-linear, analogue and/or mixed-signal circuits.

- **target faults:** soft and/or hard faults

- **information obtained:** the result of fault simulation is usually the fault coverage of the test set under consideration. However, some procedures also provide information for fault diagnosis or how the current test set can be improved to increase fault coverage.

In [174], the MiST PROFIT toolset for numerical hierarchical fault simulation and fault diagnosis is described. A faulty module is defined as a circuit block for which one or more specifications are out of their nominal ranges. This requires that for each circuit block, e.g. opamp or filter circuit, specifications are defined by the designer. The hierarchical fault modelling strategy of PROFIT is illustrated in Figure 5.13. The fault simulation

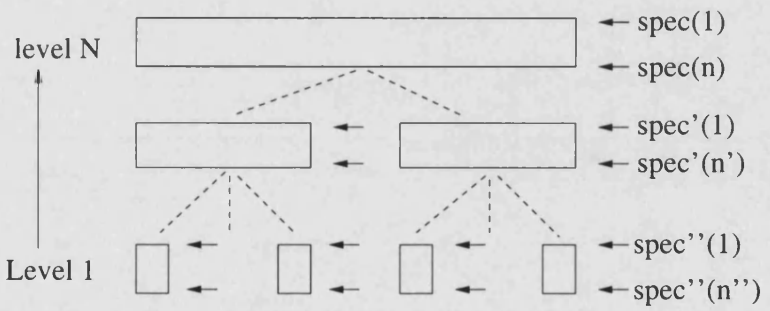


Figure 5.13: Hierarchical Fault Modelling of PROFIT [174]

process begins with level 1 for the leaf cells as the basic building blocks. Consider an opamp as an example for a leaf cell. Shorts and opens can be introduced in the transistor level circuit description (hard faults) or the transistor parameters may be changed (soft faults) to describe the fault conditions of the opamp. Using numerical simulation, the faulty specifications for each fault condition are derived. The specifications of higher level modules are determined by applying hierarchical behavioural simulation. This requires the availability of behavioural models for each circuit block.

Each module, in nominal or faulty condition, is characterised by a point in the n-dimensional specification space. Hence, each hard fault in this module is represented by a point and each parametric fault by a trajectory of points in the above n-dimensional specification space. For each fault, these points and trajectories for all modules on arbitrary hierarchical level are generated using behavioural simulation.

PROFIT associates a tolerance box with each point in the n-dimensional specification space to account for normal process fluctuations. To avoid the cost of the expensive Monte Carlo method for tolerance box derivation, PROFIT calculates the tolerance boxes based on a worst case heuristic. Assuming that the specifications depend in a monotonic manner on the circuit parameters, the extreme values of the specifications are obtained at a corner of the tolerance box in the parameter space based on sensitivity information.

To further reduce fault simulation expense, fault clustering is applied. Faults within an opamp which result in a similar faulty specification need not be distinguished at higher hierarchical levels. This reduces the number of faults to be simulated. The clustering within PROFIT is obtained based on a granularity measure which measures the similarity between two fault 'syndromes'. Two different fault clustering approaches are installed: *clustering for simulation* and *clustering for diagnosis.* Clustering for simulation tries to cluster all faults within a module. Clustering for diagnosis is the same except that no two points belonging to two different sub-modules are clustered together. By this means, fault diagnosis down to lower hierarchical levels becomes possible. Accelerations of approximately 30% and 15% are achieved with clustering for simulation and clustering for diagnosis respectively.

Fault detection and diagnosis in PROFIT is based on a comparison of the entries in the fault dictionary (derived by fault simulation as described above) with the measured specifications. Diagnosis is achieved by calculating a similarity measure with respect to

the measured value and the entries of the dictionary. This similarity measure takes into account the tolerance boxes derived above. The results presented in [174] show good performance in fault detection and mostly good results for fault diagnosis. Fault diagnosis sometimes yields ambigiuous results, especially for parametric faults due to the tolerance bands around the fault trajectories.

Referring to the above criteria for fault simulation procedures, PROFIT can be classified as a general approach applicable to large-scale non-linear mixed-signal circuits. Complex circuits can be analysed due to the effective hierarchical behavioural simulation approach and the applied fault clustering. Fault clustering, however, seems to be of minor relevance because the achieved simulation acceleration lies only between 15% and 30% and the clustering itself also requires some effort. PROFIT considers hard and soft faults. The information obtained is useful for both fault detection and diagnosis. No information is generated, however, on how to increase fault coverage.

One important point to note is that PROFIT relies on the availability of behavioural models for all modules at each hierarchical level. If the models are not provided by the designer, additional manual modelling effort is required. The quality of these behavioural models is crucial for the precision of the obtained results. Care needs to be taken that the specifications and behavioural models reflect all possible modes of operation such that a correct propagation of nominal and faulty behaviour through the hierarchical levels is guaranteed. Also important is that faulty out-of-specification input signals are transmitted correctly by the behavioural models. This may be a critical point, because usually behavioural models are optimized for precision in the case of in-specification input signals. The treatment of tolerance effects in PROFIT is improved due to a new tolerance band algorithm. However, it is still assumed that the parameter-specification dependencies are monotonic and the proposed tolerance band algorithm may fail when applied to strongly

non-linear circuits.

To evaluate the effectivity of the PROFIT approach additional circuit applications would be helpful. Interesting is, how accurate the results are when behavioural models have been provided by the designer and no manual optimization of the models with respect to fault simulation has been made.

A further functional approach to fault simulation is FaultMaxx [53, 175]. Fault simulation is accelerated based on a perturbation model. The effects of parametric and hard faults are determined based on a linear approximation of the relationship between fault parameter and circuit specification. In the case of a parametric fault, the linear approximation is derived based on the sensitivity functions. The effects of hard faults are approximated with help of the gradient method.

The benefit of the FaultMaxx technique is that only two circuit simulations are required to determine the nominal circuit behaviour and the sensitivities (adjoint approach [161]). Successive fault simulations are replaced by the cost effective evaluation of the linear perturbation model. Tolerances due to normal process fluctuation can be determined based on the linear perturbation model also. No hierarchical modelling is required within this approach. Another benefit of FaultMaxx is that information is obtained on how to improve testability. For example, the sensitivity information can be used to find a frequency where the circuit reacts most significantly to a deviation of a distinct parameter. This frequency is the best test frequency for detection of deviations of the respective parameter.

FaultMaxx can be generally applied to linear and non-linear circuits and can handle both parametric and hard faults. The question is whether the applied linear approximation is sufficiently accurate, especially when large parametric deviations and strongly non-linear parameter-performance dependencies are to be considered.

Several techniques have been proposed to improve the precision of modelling the parameter-performance dependencies. In [176] the parameters of the behavioural models at different hierarchical level are related using a piece-wise linear approximation. In [36] cross-correlation techniques and a neural network formulation are used to derive the behavioural model parameters faster and more accurately. In [177] the input-output relationship of each hierarchical block is approximated using 'multiple adaptive regression splines'. All of these procedures significantly alleviate the characterisation of hierarchical models, however the model structure needs to be provided and care needs to be taken that the precision of these models is sufficient.

The major contribution of DRAFT [178] is a unification of analogue and digital fault simulation such that testability analysis of mixed-signal circuits with tightly interconnected analogue and digital circuit blocks becomes feasible. Analogue behaviour is discretized and represented behaviourally in the Z-domain. This provides a similar circuit representation as for digital circuits and allows fast fault simulation. A further advantage of this approach is that it may become unnecessary to isolate analogue and digital parts during the test process, thereby avoiding the overhead of scan. DRAFTS applies only to linear circuit behaviour due to the Z-domain circuit representation. The generation of the Z-domain models need to be accomplished for each faulty condition. Furthermore, a correct choice of the sampling frequency is essential such that the analogue time-continuous behaviour is sampled sufficiently accurate.

As a summary, the numerical behavioural fault simulation procedures become interesting within a top-down design methodology when the designer has already generated behavioural models for the circuit blocks. The crucial point is whether these behavioural models are also capable of representing faulty circuit behaviour. However, because of growing circuit complexity, hierarchical behavioural design methodologies seem of grow-

ing importance in the future.

A totally different approach to accelerate fault simulations is the so-called concurrent fault simulation technique. The idea behind this approach is the observation that the system matrices for the fault-free circuit and the faulty circuit instances are very similar. It may be useful to exploit this similarity by concurrently simulating all circuit instances rather than performing simulation sequentially.

In [179] all fault simulations are performed together with the nominal circuit simulation. In the case of a transient simulation, the same time steps are used for all circuit instances. During the Newton-Raphson linearization, the model equations of a distinct device are only re-evaluated when the terminal voltages of the device significantly differ from the respective voltages in the nominal circuit instance. This saves a lot of computing time.

The CONCERT fault simulator [180] orders the faults concerning their similarity in behaviour. The ordering of circuit instances takes place before every time step of transient simulation, based on the simulation results of previous time steps. While solving the system matrices, the state of a circuit instance is predicted from the preceeding simulated circuit instance. Because of the performed fault ordering, two successive circuit instances usually behave very similarly and a great reduction of the number of Newton-Raphson iterations is achieved. In addition, a 'reduced-order fault matrix' computation exploits the similarity between the nodal admittance matrices of the faulty and fault free circuit instances. The speed-up reported varies between 2 and 200 compared to sequential numerical fault simulation.

The approach presented in [181] exploits the similarity of the LU decomposition of the system matrix of the fault-free and faulty circuit. By organizing the MNA contributions of the inserted fault in an approriate manner, the LU factorization of the major block of

the system matrix is required only once for the fault-free circuit. Simulating the faults requires only calculations on row and column vectors. The speed up reported varies from 9 to 160.

Concurrent fault simulation is an automatic procedure which does not require manual effort like behavioural modelling. Furthermore, since no approximations are involved in the approaches [180, 181], the results should be no less precise than those of any transient simulation. A drawback of the concurrent procedures is their large storage requirement. This is caused by the fact that the system matrices of all circuit instances need to be available concurrently. Additionally, for all circuit instances the time steps of transient simulation need to be the same. This either increases simulation time for some circuit instances (time step smaller than necessary) or deteriorates the accuracy for some other circuit instances (time step too large).

The symbolic SOE fault simulation method described in this thesis is comparable to the numerical concurrent approach. Within the symbolic fault simulation the similarity between the fault-free and faulty system is exploited by re-evaluating only those SOE expressions which depend on the faulty parameter. The SOE approach is primarily for parametric fault simulation but can be easily extended to open faults by setting the respective conductance (G or C) to zero. Short circuit situations can be simulated in the case that the short is located between two nodes which are already connected by a passive element in the fault-free circuit. Then the short is simulated by setting the impedance of this element to zero and evaluating the SOE appropriately. The SOE appoach needs no manual effort (e.g. behavioural modelling) and can be fully automated. The storage requirements of the SOE method are minimal compared to the numerical concurrent technique since only the DAG and two SOEs (nominal and faulty) need to be stored simultaneously. A further advantage of the symbolic method concerns numerical stability: the inclusion of

shorts and opens into a netlist may cause the system matrix to be ill-conditioned which can lead to iteration problems within a numerical simulation. Such situations can be prevented by using a symbolic approach instead of numerical iterations.

The cost for the advantages of the symbolic approach is that only circuits with linear input-output relation can be handled. Additionally, fault clustering is not implemented. The FaultMaxx technique can handle general non-linear circuits, however, the parameter-performance dependencies are only linearly approximated using sensitivity functions. The SOE approach is opposite in the sense that it can handle only linear circuit behaviour (linear with respect to the input-output relationship) but can deal with non-linear parameter-performance dependencies.

A general important question is how effective is the SOE approach compared to numerical simulation? Table 5.12 [138] compares two symbolic methods (determinant decision diagrams (DDD) and SCAPP) with the numerical simulator SPICE concerning their CPU time requirements. Subcircuits #1 to #4 refer to cascading the circuit blocks shown in

# subcircuit	DDD		SCAPP			SPICE	
	constr.	sim.	analy.	comp.	sim.	setup.	sim.
1	0.37	2.09	0.81	13.1	2.60	1.10	5.34
2	1.01	4.75	2.09	33.3	7.49	2.70	8.98
3	2.42	6.91	3.69	44.2	10.37	3.12	15.58
4	12.75	9.19	5.54	64.7	12.06	3.42	22.10

Table 5.12: Comparison of CPU time requirements [138]

Figure 5.10 row by row such that subcircuit #1 refers to $T1 - T3$, #2 to $T1 - T6$, #3 to $T1 - T9$ and #4 to $T1 - T13$ (whole filter circuit). In column 2 and 3 of Table 5.12, the CPU time required for DDD construction and numerical evaluation of the DDD is stored respectively. In columns 4 to 6, the time for SOE analysis, compilation and numer-

ical evaluation is given respectively. Finally, column 7 and 8 show the matrix setup and simulation time of SPICE respectively.

Obviously, in terms of simulation/evaluation time, the symbolic methods are faster than numerical simulation. The speed-up lies typically between 1.5 and 2.5. Since the SOE procedures presented in section 5.3.5 further accelerate the SOE evaluation, the advantage of SOE fault simulation compared to sequential numerical fault simulation is even more significant. Actually, the overall speed-up of SOE fault simulation with respect to sequentially numerical fault simulation is a product of the speed-up reported in Table 5.12 and the speed-up reported in section 5.3.5 (see Table 5.9, 5.10 and 5.11).

A drawback of the SOE approach is the overhead related to SOE analysis and compilation. However, the SOE generation and compilation is a one time cost whereas the SOE evaluations and SPICE simulations need to be repeated for each fault in the fault list. When the number of faults is above 10, which is typically the case, simulation time and not analysis/compilation time is relevant. Then, the SOE approach is faster compared to straight forward numerical simulation. Additionally, it can be seen from Table 5.12 that the DDD approach is more effective than the SCAPP SOE method. Therefore it may be interesting to investigate whether the SOE fault simulation procedures presented in this thesis can be combined with the symbolic DDD representation to further accelerate symbolic fault simulation.

Having investigated numerical and symbolic fault simulation, the question is how the SOE sensitivity procedure compares with numerical sensitivity procedures, especially with the numerical adjoint approach [161] used in FaultMaxx [175]. The numerical adjoint technique derives the sensitivities with respect to all parameters in parallel by two circuit simulations and is consequently very effective for multi-parameter sensitivity analysis. Before performing such a comparison, it is investigated how the SOE can be used to

calculate sensitivity functions in parallel.

5.4 Parallel Symbolic Sensitivity Analysis

The SOE sensitivity procedures presented up to now calculate the sensitivities of the network function for each circuit parameter separately. In this section, a method is described which derives from the SOE the network sensitivities with respect to all parameters simultaneously [182]. Experimental results show that the benefit of the new *parallel* procedure is a further substantial reduction of computing time of multi-parameter sensitivity analysis, which improves the applicability of symbolic sensitivity methods to large scale circuits.

5.4.1 Bottom-up Approach

The approaches in [167] – [169] and [171] use a *bottom-up* procedure to derive the sensitivity functions. For each circuit parameter, the calculations are carried out bottom-up along the paths from the respective leaves to the root of the weighted DAG. As a consequence a lot of DAG edges are crossed several times. For example, consider the SOE and DAG shown in Figure 5.6 (Page 101). The vertex $N10$ lies on paths starting at the leaf $N1$, as well as on paths starting at the other leaves $N2$, $N3$ and $N4$. Calculating the sensitivities with respect to G_1, G_2, G_3 and G_4 causes the edge between $N10$ and $N11$ to be crossed four times and the edge weight $w(10, 11)$ occurs four times in the respective sensitivity equations shown in Figure 5.8 (Page 103). In each sensitivity equation set, $w(10, 11)$ is to be multiplied with the actual node weight $n_{10} = \frac{\partial H_{10}}{\partial G_i}, i = 1 \ldots 4$. In circuits with many parameters, the situation becomes even worse and a lot of paths and edges are crossed many times which causes many terms to be produced by the respective edge weights. This results in high redundancy and finally in large computational expense.

5.4.2 Top-down Approach

To improve the situation described above a *top-down* approach will be applied. By this means the sensitivities of a network function H with respect to all parameters are obtained in parallel within one traversal through the DAG, crossing each edge only once. The top-down procedure works as follows: starting with the root of the DAG, partial derivatives of the network function H with respect to each expression H_j of the SOE are calculated, proceeding top-down until the leaves of the DAG are reached. Referring to the example of the output admittance $Y_{out} = H = H_7$ of the 2-stage ladder circuit the top-down procedure yields

$$
\begin{aligned}
H_7 &= H_6 + H_5 & \frac{\partial H_7}{\partial H_7} &= 1 \\
H_6 &= \frac{H_3 H_4}{H_3 + H_4} & \frac{\partial H_7}{\partial H_6} &= \sum_{(6,i)\in DAG} \frac{\partial H_7}{\partial H_i}\frac{\partial H_i}{\partial H_6} = \frac{\partial H_7}{\partial H_7}\frac{\partial H_7}{\partial H_6} = 1\cdot 1 \\
H_5 &= G_4 & \frac{\partial H_7}{\partial H_5} &= \sum_{(5,i)\in DAG} \frac{\partial H_7}{\partial H_i}\frac{\partial H_i}{\partial H_5} = \frac{\partial H_7}{\partial H_7}\frac{\partial H_7}{\partial H_5} = 1\cdot 1 \\
H_4 &= G_3 & \frac{\partial H_7}{\partial H_4} &= \sum_{(4,i)\in DAG} \frac{\partial H_7}{\partial H_i}\frac{\partial H_i}{\partial H_4} = \frac{\partial H_7}{\partial H_6}\frac{\partial H_6}{\partial H_4} = \frac{\partial H_7}{\partial H_6}\cdot\frac{H_3^2}{(H_3+H_4)^2} \\
H_3 &= H_1 + H_2 & \frac{\partial H_7}{\partial H_3} &= \sum_{(3,i)\in DAG} \frac{\partial H_7}{\partial H_i}\frac{\partial H_i}{\partial H_3} = \frac{\partial H_7}{\partial H_6}\frac{\partial H_6}{\partial H_3} = \frac{\partial H_7}{\partial H_6}\cdot\frac{H_4^2}{(H_3+H_4)^2} \\
H_2 &= G_2 & \frac{\partial H_7}{\partial H_2} &= \sum_{(2,i)\in DAG} \frac{\partial H_7}{\partial H_i}\frac{\partial H_i}{\partial H_2} = \frac{\partial H_7}{\partial H_3}\frac{\partial H_3}{\partial H_2} = \frac{\partial H_7}{\partial H_3}\cdot 1 \\
H_1 &= G_1 & \frac{\partial H_7}{\partial H_1} &= \sum_{(1,i)\in DAG} \frac{\partial H_7}{\partial H_i}\frac{\partial H_i}{\partial H_1} = \frac{\partial H_7}{\partial H_3}\frac{\partial H_3}{\partial H_1} = \frac{\partial H_7}{\partial H_3}\cdot 1
\end{aligned}
\tag{5.25}
$$

The SOE describing the nominal admittance and the top-down sensitivity expressions are shown on the left hand and right hand side respectively. According to the chain rule of differentiation each equation in this sequence has the structure

$$
\frac{\partial H}{\partial H_j} = \sum_{(j,i)\in DAG} \frac{\partial H}{\partial H_i}\frac{\partial H_i}{\partial H_j}. \tag{5.26}
$$

Thereby the summing condition is a consequence of the fact the DAG edges represent the explicit dependencies between SOE expressions. In order to evaluate (5.26) for a given j,

both partial derivatives on the right hand side, $\frac{\partial H}{\partial H_i}$ and $\frac{\partial H_i}{\partial H_j}$ need to be known already. The first term, $\frac{\partial H_i}{\partial H_j}$, refers to explicit dependencies between the expressions in the SOE. This partial can be calculated directly as all nominal values for the expressions H_i and H_j are known from nominal analysis. Since there exist only backward dependencies within a SOE (equation 5.6) the following relation holds:

$$(j, i) \notin \text{DAG}, \quad \forall i \leq j. \tag{5.27}$$

Consequently, the calculation of (5.26) for a given j only requires the knowledge of $\frac{\partial H}{\partial H_i}$ with $i > j$. Therefore, proceeding top-down in the DAG and in the respective SOE as proposed in (5.25) yields all partial derivatives without any recurrence in calculations. According to Section 5.2, the circuit parameters x_l correspond to the DAG leaves and the respective SOE leaf expressions $H_{i(l)}$ of the SOE:

$$\text{leaf expression:} \qquad H_{i(l)} = x_l. \tag{5.28}$$

Consequently, the sensitivities of the network function are given as the partial derivatives of H with respect to the leaf expressions:

$$\text{sen}(H, x_l) = \frac{\partial H}{\partial x_l} = \frac{\partial H}{\partial H_{i(l)}}. \tag{5.29}$$

As a result, evaluating a SOE top-down as demonstrated in (5.25) generates all sensitivities in parallel. So, the sensitivities $\frac{\partial H}{\partial G_1}$, $\frac{\partial H}{\partial G_2}$, $\frac{\partial H}{\partial G_3}$ and $\frac{\partial H}{\partial G_4}$ are obtained in example (5.25) as the partials $\frac{\partial H}{\partial H_1}$, $\frac{\partial H}{\partial H_2}$, $\frac{\partial H}{\partial H_4}$ and $\frac{\partial H}{\partial H_5}$ respectively.

5.4.3 Signal Graph Interpretation

Similarly to the bottom-up approach, the top-down approach can be formulated as a signal graph problem [182]. For this purpose, the Top-Down Graph of a SOE is defined as follows:

Definition 5.2 (Top-down graph) *The Top-Down Graph (TDG) of a SOE is a directed weighted graph where*

- *each expression H_i is represented by a* **node** *N_i and*

- *the dependencies between the expressions are represented by* **edges***: there exists a directed edge (i, j) from node N_i to node N_j if and only if expression H_i depends explicitly on H_j.*

- *associated with each edge (i, j), is an edge weight $w(i, j) = \frac{\partial H_i}{\partial H_j}$.*

The TDG is very similar to the weighted expression graph DAG (Definition 5.1) with edge weights according to equation (5.10). The only difference is that the direction of the edges in the TDG is reversed compared to the DAG. Consequently the paths in the TDG start at the root of the graph and terminate at the leaves. The TDG of the SOE for the output admittance of the 2-stage ladder circuit is shown in Figure 5.14. By defining the node weight

$$n_i := \frac{\partial H}{\partial H_i} \tag{5.30}$$

$$\begin{aligned} H = H_7 &= H_6 + H_5 \\ H_6 &= \frac{H_3 H_4}{H_3 + H_4} \\ H_5 &= G_4 \\ H_4 &= G_3 \\ H_3 &= H_1 + H_2 \\ H_2 &= G_2 \\ H_1 &= G_1 \end{aligned}$$

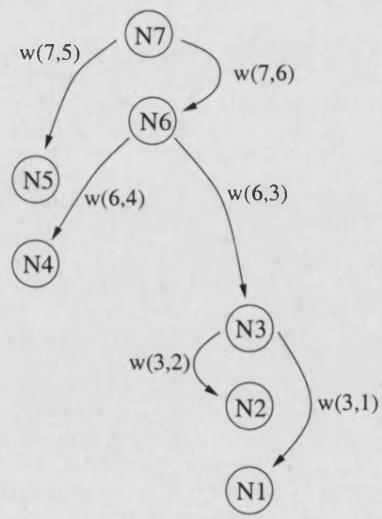


Figure 5.14: SOE and its corresponding TDG

the key equation of top-down sensitivity analysis (5.26) can be written purely in terms of the TDG:

$$n_j = \sum_{(i,j)\in TDG} n_i \, w(i,j). \tag{5.31}$$

Equation (5.31) is equivalent to the node equation of a signal flow graph [170]. Consequently, the top-down sensitivity analysis can be interpreted as the solution of a corresponding signal graph problem. The corresponding signal graph is the weighted TDG, the source node is the root H with node weight 1. The sensitivities are obtained as the weights of the respective leaves (leaves=output nodes), after the node weights have been propagated through the TDG according to equation (5.31). For illustration, the top-down sensitivity equations (5.25) are written in terms of the TDG:

$$\begin{aligned}
n_7 &= 1 \\
n_6 &= n_7\, w(7,6) \\
sen(H,G_4) = n_5 &= n_7\, w(7,5) \\
sen(H,G_3) = n_4 &= n_6\, w(6,4) \\
n_3 &= n_6\, w(6,3) \\
sen(H,G_2) = n_2 &= n_3\, w(3,2) \\
sen(H,G_1) = n_1 &= n_3\, w(3,1)
\end{aligned} \tag{5.32}$$

$$\text{where} \quad w(3,1) = w(3,2) = 1, \; w(6,3) = \frac{H_4^2}{(H_3+H_4)^2},$$
$$w(6,4) = \frac{H_3^2}{(H_3+H_4)^2}, \; w(7,5) = w(7,6) = 1.$$

5.4.4 Computational Aspects

For estimation of the computational expense of the top-down sensitivity analysis procedure, the assumption is made that practical circuits with sparse topology are considered.

In such a case, the subsequent relationships hold [89]:

$$N_n = O(n), \tag{5.33}$$

$$N_e = O(n), \tag{5.34}$$

where N_n and N_e are the number of TDG (or DAG) nodes and edges respectively. The heart of the top-down procedure is equation (5.31). In this equation, a term $n_i\, w(i,j)$ is generated for each edge ending at the node N_j. Since within the top-down procedure all TDG nodes are visited once, the overall number of terms is given by the number of all TDG edges. This yields

$$\text{expense(top-down):} \qquad O(N_e) = O(n). \tag{5.35}$$

According to these estimations, the expense of the top-down sensitivity algorithm grows only linearly with circuit complexity. The expected speed-up of the top-down approach is

- n compared to the unbalanced bottom-up approach,
- $\log_2 n$ compared to the balanced bottom-up approach.

This makes the top-down procedure effective, especially for the analysis of large-scale systems.

5.4.5 Experimental Results

The top-down procedure has been joined to the other SOE sensitivity procedures in Maple V [172]. To examine the efficiency of the top-down procedure two practical circuits have been analysed. For comparison with previous SOE sensitivity procedures, the same examples as in Section 5.3.4 are investigated.

Ladder Networks

Firstly, the resistive 2-stage ladder network according Figure 5.2 (Page 93) is considered. The sensitivities of the transfer function with respect to all 4 parameters are to be calculated. The edge weights $w(i,j)$ shown in Figure 5.7 (Page 102) are the same for both the bottom-up and top-down procedure. The bottom-up procedure generates a separate set of node weight equations for each parameter (see Figure 5.8, Page 103). In contrary, the top-down procedure generates only one equation set which is shown in Figure 5.15. The

$$
\begin{aligned}
n_{11} &= 1 \\
n_{10} &= n_{11}\, w(11,10) \\
n_9 &= n_{11}\, w(11,9) \\
n_8 &= n_9\, w(9,8) + n_{10}\, w(10,8) \\
n_7 &= n_8\, w(8,7) \\
n_6 &= n_{10}\, w(10,6) \\
n_5 &= n_7\, w(7,5) \\
sen(H,G_4) = n_4 &= n_6\, w(6,4) \\
sen(H,G_3) = n_3 &= n_6\, w(6,3) + n_7\, w(7,3) + n_8\, w(8,3) + n_{10}\, w(10,3) \\
sen(H,G_2) = n_2 &= n_5\, w(7,2) \\
sen(H,G_1) = n_1 &= n_9\, w(9,1) + n_5\, w(5,1)
\end{aligned}
$$

Figure 5.15: Top-down sensitivity (TDG node weight) equations

quality of SOE sensitivity analysis will again be measured based on the required number of arithmetic operations. Calculation of the edge weights requires 6 multiplications and

2 additions (Figure 5.7). The evaluation of the top-down sensitivity equations requires 6 multiplications[7] and 5 additions (Figure 5.15). Together with the edge weight equations the expense is only 12 multiplications and 7 additions as opposed to 18 multiplications and 9 additions required by the bottom-up approach.

Now, ladder circuits with different numbers of ladder stages are analysed. The computational expense of three different methods for multi-parameter sensitivity analysis are compared:

method 1: bottom-up procedure using unbalanced BPT

method 2: bottom-up procedure using balanced BPT

method 3: top-down (parallel) procedure

Table 5.13 and Table 5.14 show the number of additions and multiplications required by these methods. Generally, the top-down method is significantly faster than both bottom-up approaches. Moreover, this advantage is growing with circuit size. In the case of a ladder circuit with 4 stages the gain in speed is a factor of 3 compared to method 1 and a factor of 2 compared to method 2 referring to the number of multiplications. In the case of a circuit with 128 stages the gain is a factor of 70 compared to method 1 and a factor of 6 compared to method 2. Figure 5.16 illustrates the number of multiplications required by the top-down procedure in dependence on the number of circuit nodes n of the ladder network. The points fit well on a straight line which verifies that the top-down procedure has only linear complexity in the case of sparse circuits (see equation 5.35).

[7]Some of the node and edge weights are identical to 1, no matter what frequency is applied to the circuit and irrespective of the actual parametric values. In the above examples such weights are, n_{11}, $w(10,6)$, $w(7,5)$, $w(6,4)$, $w(6,3)$, $w(7,3)$, $w(5,2)$ and $w(5,1)$. Multiplications with these quantities are trivial and therefore not counted.

number of stages	number of parameters	method 1 N_{adds}	method 2 N_{adds}	method 3 N_{adds}
2	4	9	9	7
4	8	61	44	30
8	16	261	141	79
16	32	1045	398	184
32	64	4149	1039	401
64	128	16501	2576	842
128	256	65781	6161	1731

Table 5.13: Sensitivity analysis of ladder circuit with respect to all parameters: number of additions performed

number of stages	number of parameters	method 1 N_{mults}	method 2 N_{mults}	method 3 N_{mults}
2	4	18	18	12
4	8	131	92	43
8	16	619	317	113
16	32	2651	938	263
32	64	10939	2527	573
64	128	44411	6404	1203
128	256	178939	15561	2473

Table 5.14: Sensitivity analysis of ladder circuit with respect to all parameters: number of multiplications performed

Large Scale Bandpass Filter

As a second example the bandpass filter of Figure 5.10 is examined. Table 5.15 shows the cost for sensitivity analysis of this circuit. Again, the top-down method is considerably faster than both bottom-up procedures.

	N_{adds}	N_{mults}
method 1	657	1828
method 2	258	822
method 3	108	219

Table 5.15: Expense of bandpass sensitivity analysis

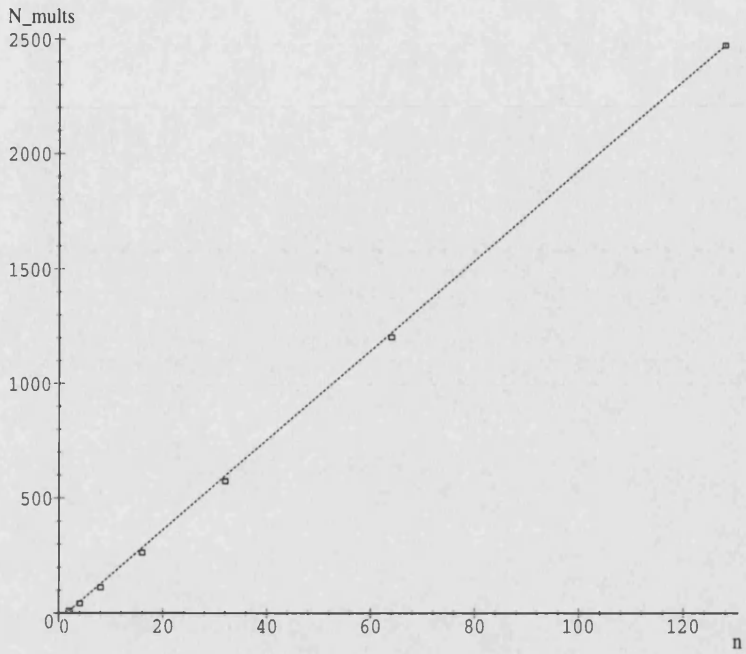


Figure 5.16: Sensitivity analysis of ladder circuit with respect to all parameters: number of multiplications of the top-down method in dependence on the number of circuit nodes

5.5 Summary & Conclusions

In this chapter, hierarchical symbolic procedures have been presented which are appropriate for the sensitivity analysis of large scale analogue circuits. Two methods,

- the *balanced* approach,
- the *parallel* approach

are proposed to organize SOE sensitivity analysis most effectively. Especially in the case of larger circuits a speed up of multi-parameter sensitivity analysis of factors up to 70 compared to previous approaches has been achieved. The balanced method can be applied both to differential sensitivity analysis and to examinations concerning large parametric deviations, for example to parametric fault simulation. The parallel approach can be applied merely to differential sensitivity analysis. On the other hand, the parallel approach

is the faster one of the two procedures. Another advantage of the parallel method is that it doesn't rely on a hierarchical SOE generation algorithm and consequently can be combined with any SOE. The capabilities of the two sensitivity methods are summarized in Table 5.16. Due to their low requirements, the small- and large-change SOE sensitivity

	balanced method	parallel method
differential sensitivity	√	√
large change deviations, e.g. fault simulation	√	–
speed-up compared to previous SOE procedures	1...20	1...70
computational expense in dependence on circuit complexity n	$O(n \log_2 n)$	$O(n)$
algorithmic requirements	hierarchical partitioning based SOE generation	no requirements, arbitrary SOE

Table 5.16: Assessment of SOE sensitivity analysis algorithms

procedures presented in this thesis can be efficiently applied to tolerance analysis and optimization procedures tackling large scale analogue circuits.

An interesting question is how the symbolic sensitivity procedures compare with numerical approaches. Concerning multi-parameter sensitivity analysis, the numerical counterpart of the parallel SOE procedure is the adjoint matrix approach [161]. In this approach, two equation systems, one referring to the original network and one referring to the adjoint network, are solved. By combining the results of both simulations appropriately, the sensitivities with respect to all parameters are derived simultaneously.

To compare the effectivity of the adjoint approach with the parallel SOE method, a 128-stage ladder circuit according to Figure 5.2 is considered. Within the numerical approach, the ladder circuit is described by a band matrix. The total cost to solve the band matrix, i.e. the LU decomposition and the forward and backward substitutions, is 636

multiplications and 381 additions[8]. The adjoint network is described by the transposed matrix of the matrix describing the original network. For the transposed matrix the LU decomposition needs not to be repeated and only the forward and backward substitutions are performed which requires 382 multiplications and 254 additions. Combining the results of both systems to derive the network sensitivities requires 256 multiplications and 256 additions. The overall expense of the numerical adjoint approach is shown in the upper row of Table 5.17. In the lower row, the respective cost of the parallel SOE procedure is given.

method	# adds	# mults
numerical adjoint	891	1274
parallel SOE	1731	2473

Table 5.17: Operations for all-parameter sensitivity analysis of 128-stage ladder circuit

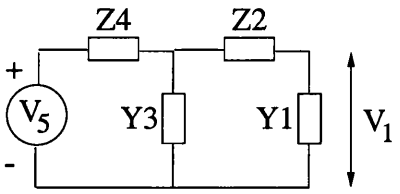
The parallel SOE requires approximately two times more arithmetic operations than an effective numerical sparse matrix solver when the similarity between the original and the adjoint network is fully exploited. Based on these results, the parallel SOE procedure seems not be advantageous compared to the numerical adjoint approach. However, there is one important degree of freedom which can be exploited to significantly improve the performance of the parallel SOE procedure: the generation of the nominal SOE.

The results presented in this thesis were based on a MAPLE implementation of the basic features of SCAPP. More involved symbolic techniques, like term sharing between different SOE expressions, for example, have not been implemented yet. Furthermore, Table 5.12 indicates that new symbolic procedures have been developed during the last two years with significantly increased effectivity compared to SCAPP. These new procedures hold therefore promise for a combination with the sensitivity techniques described in this

[8]It is assumed that an effective sparse matrix solver is applied.

thesis.

To show that there is room for further significant improvement, an alternative nominal SOE for ladder circuits is investigated. In Figure 5.17 a two stage ladder circuit is illustrated. On the right hand side the new SOE is shown which can be used to calculate



$$
\begin{aligned}
i_1 &= y_1 v_1 \\
i_2 &= i_1 \\
v_2 &= z_2 i_2 \\
v_3 &= v_1 + v_2 \\
i_3 &= y_3 v_3 \\
i_4 &= i_2 + i_3 \\
v_4 &= z_4 i_4 \\
v_5 &= v_3 + v_4.
\end{aligned}
$$

Figure 5.17: 2-stage ladder circuit and new SOE

the transfer function $H = v_1/v_5$. This SOE is a more compact alternative than the old SOE of Figure 5.6.

When extending the new SOE structure to ladder circuits with more stages, k-th stage contributes the following 4 equations with 2 multiplications and additions:

$$
\begin{aligned}
i_{2k-1} &= y_{2k-1} v_{2k-1} \\
i_{2k} &= i_{2k-2} + i_{2k-1} \\
v_{2k} &= z_{2k} i_{2k} \\
v_{2k+1} &= v_{2k-1} + v_{2k}.
\end{aligned}
\tag{5.36}
$$

For comparison, the old SOE according to Figure 5.6 requires 4 multiplications and 3 additions per stage. Altogether, using the new SOE, the expense of nominal analysis of a n-stage ladder circuit requires $2n$ multiplications and $2n-1$ additions (first stage requires

only 1 addition).

To investigate the expense of the parallel sensitivity procedure using this new SOE, the respective DAG is constructed. In Figure 5.18, a part of this DAG is illustrated. Between

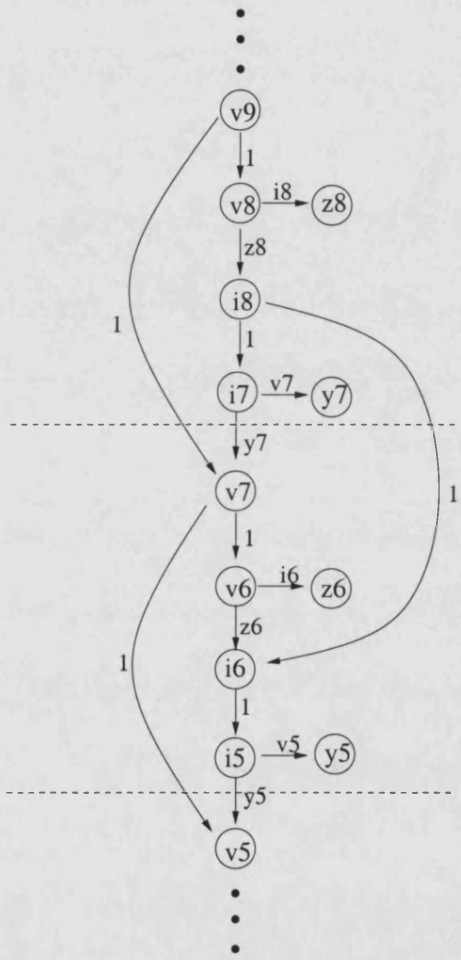


Figure 5.18: Part of DAG referring to the new SOE of ladder circuit

the dashed lines that part of the DAG referring to the 3rd stage is shown. The symbols attached to the edges refer to the edge weights. From this DAG, the parallel sensitivity SOE can be derived. The part which refers to the 3rd stage is:

$$
\begin{aligned}
n_{v7} &= y_7 n_{i7} + n_{v9} \\
n_{v6} &= n_{v7} \\
n_{z6} &= i_6 n_{v6} \\
n_{i6} &= z_6 n_{v6} + n_{i8} \\
n_{i5} &= n_{i6} \\
n_{y5} &= v_5 n_{i5}.
\end{aligned}
\tag{5.37}
$$

This contributes 4 multiplications and 2 additions. Since every stage is represented by the same DAG structure, $4n$ multiplications and $2n$ additions are required for evaluating the parallel sensitivity SOE of a n-stage ladder circuit. Together with nominal analysis this yields $6n$ multiplications and $4n - 1$ additions.

For comparison of the expense of the numerical adjoint approach and the parallel SOE method, $n = 128$ is assumed again. The costs are compared in Table 5.18. These results

method	# adds	# mults
numerical adjoint	891	1274
new parallel SOE	512	768

Table 5.18: Sensitivity analysis of 128-stage ladder circuit, new SOE

show that the parallel procedure is faster than the numerical adjoint approach under the condition that an optimal nominal SOE is provided. The parallel procedure can be further accelerated by compiling the derived parallel sensitivity SOE.

Chapter 6

Symbolic Tolerance Analysis

In the last chapter, sensitivity analysis techniques for large scale networks have been presented. These techniques are useful for circuit optimization and to get quickly a first insight into the tolerance behaviour of a circuit. However, to investigate the effects of parametric deviations more accurately, additional methods are required. Unfortunately, these accurate methods are usually extremely time intensive, mostly in an extent that their application to larger circuits during the design process is prohibitive.

The goal of this chapter is to propose an effective tolerance analysis procedure which meets the analysis time restrictions and yields results of improved accuracy compared to previous fast approaches. This allows for tolerance considerations in an early design stage.

After reviewing the tolerance analysis methods currently available, the Quantile Arithmetic of [183] is adopted and modified for higher precision and speed. The developed Modified Quantile Arithmetic [184] is combined with the SOE approach of [89] to make possible tolerance analysis of large scale analogue circuits. Practical circuit examples are analysed to examine the applicability and benefit of the new method.

6.1 Introduction

Tolerance analysis predicts the performance spreads of the circuit based on the knowledge of the component statistics and thereby allows predicting yield before fabrication is started. Yield is defined as

$$Y_{yield} := \frac{\text{number of devices which meet all specifications}}{\text{number of all fabricated devices}}. \tag{6.1}$$

Yield is an important factor for product cost assessment and economic planning. If the estimated yield turns out to be unacceptably low, the results of tolerance analysis help during optimizing the circuit with respect to reduced performance variations.

There exist several procedures for tolerance analysis in literature, reviews are given for example in [153, 185, 186, 187]. A classification of tolerance analysis procedures is shown in Figure 6.1. In the following, the benefits and limitations of the different methods are discussed.

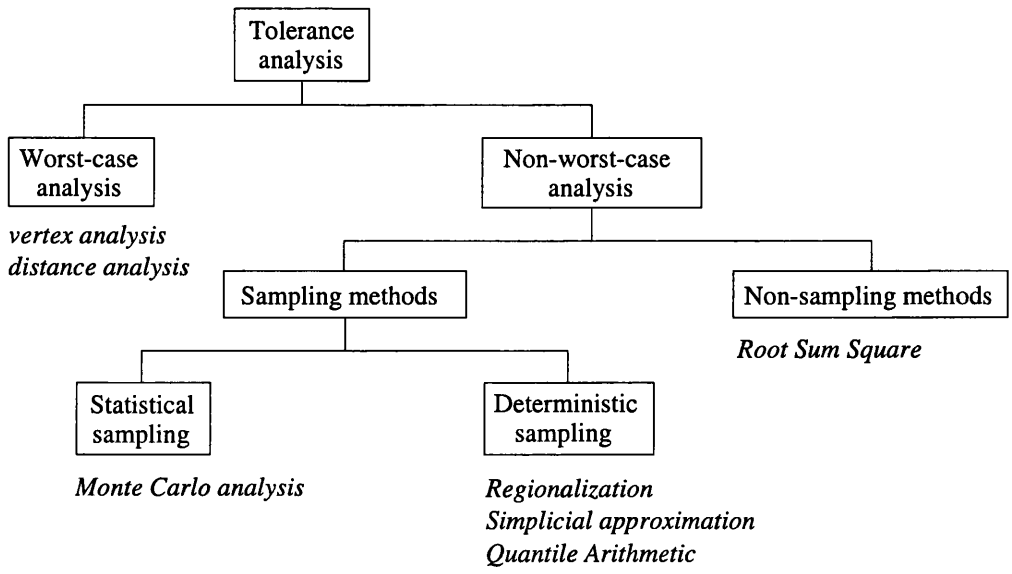


Figure 6.1: Classification of tolerance analysis procedures [185]

6.1.1 Worst-Case Analysis

The goal of worst-case analysis is to determine the worst-case (the largest) deviation of the circuit's performance that might be caused by the underlying parameter component deviations. Mathematically, this task can be formulated as multi-dimensional non-linear optimization problem. As such problems cannot be solved unambiguously [188], heuristic solutions have been proposed. One example of these methods is the worst-case vertex [189] analysis. This technique tries to find the worst case deviation by simulating the circuit using corner values in the parameter space. This method is improved in [190] by reducing the number of simulations based on Taguchi's technique and fractional factorial experimental design. However, these methods work only for monotonic parameter-performance dependencies. The information obtained by worst-case vertex analysis is whether the worst-case deviation lies within the specification limit or not, or in other words, whether yield is 100% or not. Therefore, worst-case analysis is of little use in cases where 100% yield either cannot be achieved or where yield is being traded off against cheaper (wider tolerance) components [185].

In [191] the so-called worst-case distances are determined. The worst-case distances are defined as the minimal distance between the nominal point and the border of the area of acceptability in the parameter space. These distances are derived by searching iteratively in the parameter space for extreme performance spreads based on the results of sensitivity analysis. The obtained worst-case distances are used in [192] for optimization of parametric yield. However, certain mathematical conditions need to be satisfied that the iterative search for the worst case distances in the parameter space converges. Moreover, no information about the concrete statistical distibutions of the circuit performances is obtained by these methods.

6.1.2 Non-Worst-Case Analysis

The non-worst-case methods are applicable to the more general case where yield is less than 100%. According to Figure 6.1, these procedures can be classified into sampling and non-sampling approaches. The sampling methods are based on circuit analysis at sample points in component parameter space whereas the non-sampling methods use purely deterministic techniques to derive the circuits' tolerances.

Non-Sampling Methods

The most popular non-sampling approach is the methods of moments, also called Root Sum Square technique [193, 194]. This method is based on a first order Taylor series expansion of the circuit performance H in the circuit parameters $X = \{x_1, \ldots, x_{N_X}\}$:

$$\begin{aligned} H(x_1, \ldots, x_{N_X}) &\approx H(X_0) + \sum_{i=1}^{N_X} \left.\frac{\partial H}{\partial x_i}\right|_{X_0} \cdot \Delta x_i \\ &= H(X_0) + \sum_{i=1}^{N_X} sen(H, x_i)(X_0) \cdot \Delta x_i \end{aligned} \tag{6.2}$$

where X_0 is the nominal design point in parameter space and Δx_i are the actual parameter deviations. Important statistical measures characterising the statistics of a parameter x_i are the pdf (probability density function) $\tilde{P}(x_i)$ and the derived quantities

$$\text{mean value:} \quad \overline{x_i} = \int_{-\infty}^{\infty} \tilde{P}(x_i)\, x_i\, dx_i \tag{6.3}$$

$$\text{variance:} \quad \sigma_{x_i}^2 = \int_{-\infty}^{\infty} \tilde{P}(x_i)\, (x_i - \overline{x_i})^2\, dx_i \tag{6.4}$$

where the mean value is identical to the nominal value in case of a symmetric pdf. The variance is a measure for the widths of the parameter pdf and therefore is an indication for the parameter deviations which typically occur. A statistical measure of the dependencies of two parameters is the linear correlation coefficient defined as follows

$$\text{correlation:} \quad \rho_{x_i,x_j} = \frac{1}{\sigma_{x_i}\sigma_{x_j}} \int_{-\infty}^{\infty} \tilde{P}(x_i, x_j)\, (x_i - \overline{x_i})(x_j - \overline{x_j})\, dx_i dx_j \tag{6.5}$$

where $\tilde{P}(x_i, x_j)$ is the statistical joint-pdf of x_i and x_j. Using the Taylor expansion (6.2) the so-called Root Sum Square (RSS) formula describing the variation of the performance H can be established [195]:

$$\begin{aligned}\sigma_H^2 &= \sum_{i=1}^{N_X} (sen(H, x_i)(X_0))^2 \, \sigma_{x_i}^2 \\ &\quad +2 \sum_{i=1}^{N_X} \sum_{i=j}^{N_X} sen(H, x_i)(X_0) \, sen(H, x_j)(X_0) \, \rho_{x_i, x_j} \sigma_{x_i} \sigma_{x_j}. \end{aligned} \tag{6.6}$$

The RSS formula allows for fast determination of performance deviations, when there is an effective sensitivity analysis procedure available (see for example Chapter 5), and is used in several approaches to tolerance design [194, 196]. However, since RSS is based on a first order Taylor expansion, the technique is restricted to linear parameter-performance dependencies and may be inaccurate when larger deviations are considered. This point can be improved partially by including higher order derivatives in the Taylor expansion of the circuit performance [197]. To summarize, RSS is not very accurate but fast and provides a useful first approximate indication of tolerance effects.

Sampling Methods

There exist diverse sampling methods for tolerance analysis. Basically, these methods divide into two classes (Figure 6.1): the first one chooses the sampling points in parameter space statistically, the second applies deterministic sampling.

Statistical Sampling: The Monte Carlo analysis The Monte Carlo method is one of the most popular tolerance analysis technique. It works by selecting S random samples of parameter values with the help of a random number generator. The samples need to be in accordance with the real statistical pdfs of the parameters. The circuit is simulated at each of the parameter samples which leads to S results for the performance H. Applying suitable post processing, the requested tolerance information can be derived. Concerning

yield for example, the following estimation is obtained:

$$Y_{yield} \approx \frac{\text{number of simulation results which meet all specifications}}{S}. \tag{6.7}$$

Moreover, being a statistical method, the Monte Carlo analysis provides a measure of statistical confidence of the obtained estimates [185, 198]. Theoretically, by increasing the number of sample points, any arbitrary confidence level can be achieved which leads to arbitrary high precision in Monte Carlo predictions. Unfortunately, to achieve high precision, the number parameters samples S is normally large. As a result, many circuit simulations need to be performed which often causes unrealistic long computing times, especially when large scale circuits are considered [41]. To improve this situation, regression analysis, piece-wise linear and spline approximation techniques are used to model the performance of a hierarchical block or of the whole circuit [199, 200, 201, 202, 203, 204]. These models are used instead of tedious repetitive simulations during the Monte Carlo analysis. However, these approaches require significant modelling effort which mostly cannot be automated and there always remains a risk of lack of accuracy.

Deterministic Sampling Deterministic methods choose the parameter samples with help of some deterministic strategy. One of the best known representative of these methods is the regionalization technique [205, 206] which chooses the sample points to be the vertices of a multi-dimensional grid in parameter space. While these methods are theoretically very accurate, they typically require very high computational cost, especially when the number of circuit parameters N_X is large. This is caused by the fact that the number of grid vertices increases rapidly with the number of circuit parameters. Suppose that each parameter's tolerance range is represented by N grid vertices, the overall number of grid vertices is given as N^{N_X}. As a result, the number of required circuit simulations is N^{N_X} and the analysis expense grows exponentially with the number of circuit parameters

which causes unrealistic high computing time even with only, say, 10 circuit parameters. However, in the case of integrated circuits where the circuits statistics may be described by the influence of a few dominant process parameters, the regionalization technique can be an appropriate alternative to the Monte Carlo analysis [206]. Improved deterministic sampling methods, e.g. the simplicial approximation [207] or the ellipsoidal technique [208], show slightly better performance, however, similar to regionalization, the number of required simulations grows very rapidly with the number of circuit parameters. These techniques are therefore not of practical use.

6.1.3 Wanted: A Compromise between Computing Time and Accuracy

The methods presented so far are either *not accurate* or *very time intensive* making these techniques prohibitive for today's large scale circuits and short time to market requirements. A sensible compromise between these two contrary requirements would significantly enhance the applicability of tolerance analysis within the design process.

An early approach into this direction was proposed in [130, 183, 209] and adopted in [210]. The authors used Quantile Arithmetic which is based on discretization of random variables and a suitable symbolic description of the network function H. An acceleration of typically one order of magnitude compared to Monte Carlo analysis was achieved. The precision of this method for the prediction of a yield in the region of 95% is good. However, in the case of low yield ($< 90\%$) or very high yield ($> 98\%$), Quantile Arithmetic turns out to be unacceptably inaccurate.

In this chapter, the Quantile Arithmetic approach is adopted and modified

- for better accuracy in all regions of possible yield ($0 \leq Y_{yield} \leq 1$),
- for increased analysis speed.

The chapter is organized as follows. Firstly, the technique of Quantile Arithmetic will

be presented and the limitations stated above explained. Thereafter, two modifications are introduced to improve the performance of Quantile Arithmetic. The new developed Modified Quantile Arithmetic [184] is then combined with the SOE analysis of [89] for tolerance analysis of large scale circuits. Finally, practical circuit examples are presented to benchmark the method. Comparisons with the two most popular tolerance analysis procedures presented above, RSS and Monte Carlo analysis, are made to assess the accuracy and effectivity of Modified Quantile Arithmetic for application in the design process.

6.2 Symbolic Tolerance Analysis by Quantile Arithmetic

The task of tolerance analysis is to determine the statistic of the network function $H(s, X)$ based on the knowledge of the statistical properties of the circuit parameters $X = \{x_1, \ldots, x_{N_X}\}$. Quantile Arithmetic (QA) derives the network tolerances using discretization of random variables [183]. A continuous random variable $\tilde{Y}$ with its continuous pdf $\tilde{P}(\tilde{Y})$ is approximated by a N-point discrete random variable

$$Y = (Y_1, Y_2, \ldots, Y_N) \tag{6.8}$$

with discrete pdf

$$P(Y) = (P_1, P_2, \ldots, P_N). \tag{6.9}$$

Using Y and P instead of $\tilde{Y}$ and $\tilde{P}$ makes tolerance analysis much easier due to operations on discrete random variables as described below. Figure 6.2 illustrates the structure of QA tolerance analysis. The upper two boxes show the input to the procedure: the statistics of the circuit parameters and the symbolic representation of the network behaviour. The symbolic network description needs to be in the so-called "Quantile Sequence Of Expressions" (QSOE) format which has the following structure:

$$
\begin{aligned}
z_1 &= f_1(s, X) \\
z_2 &= f_2(s, X, z_1) \\
&\vdots \\
H(s, X) = z_m &= f_m(s, X, z_1, \ldots, z_{m-1})
\end{aligned}
\tag{6.10}
$$

where the last expression of (6.10) represents the network function of interest. QSOE is very similar to the SOE format presented in the last chapters. The only difference is that each expression z_k in (6.10) depends on maximally two expressions calculated in previous steps, so that each expression can be expressed as $z = u \circ v$ where u and v are either pre-

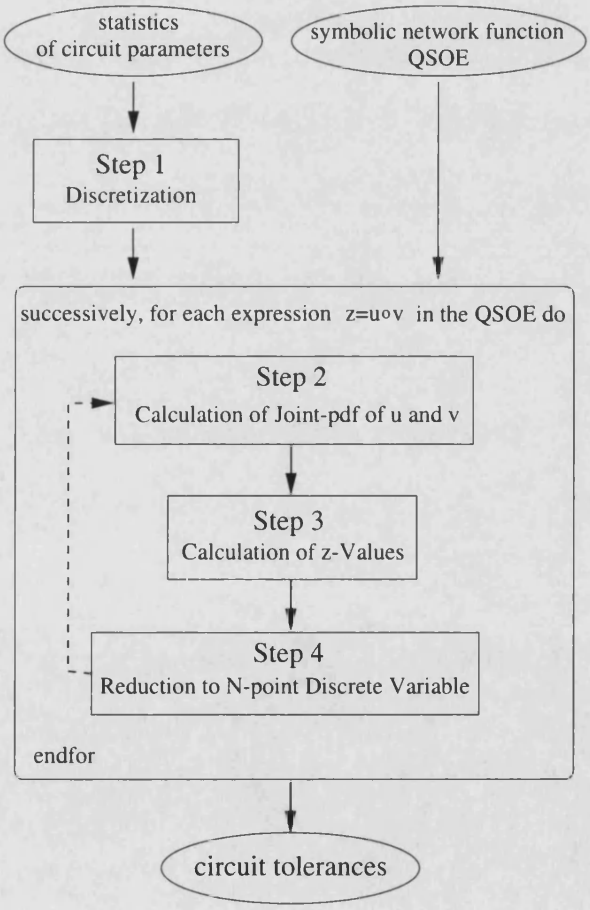


Figure 6.2: Tolerance analysis by Quantile Arithmetic [183]

vious expressions of the QSOE or circuit parameters. "$\circ$" means one of the basic binary arithmetic operations addition, subtraction, multiplication or division, $\circ \in \{+, -, \cdot, /\}$. An example for the QSOE format clarifies the situation: consider the resistive voltage di-

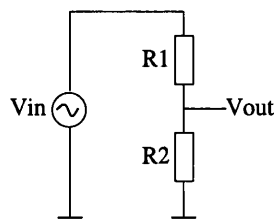


Figure 6.3: Resistive voltage divider

vider of Figure 6.3. Its transfer function $H = \frac{V_{out}}{V_{in}} = \frac{R_2}{R_1+R_2}$ is described by the following QSOE:

$$\begin{aligned} z_1 &= R_1 + R_2 \\ H = z_2 &= R_2/z_1 \end{aligned} \tag{6.11}$$

The first step of QA is the discretization of the statistical circuit parameter pdfs. Afterwards, the steps 2, 3 and 4 shown in the central box of Figure 6.2 are applied successively to each QSOE expression $z = u \circ v$. By this means, the pdf of each expression z is calculated using the statistical information of the QSOE predecessor expressions u and v. Finally, the requested network tolerance can be derived from the pdf of the last QSOE expression z_m. All four steps of QA are now explained in greater depth.

6.2.1 Step 1: Discretization of Random Variables

Let $\tilde{Y}$ be a continuous random variable with its pdf $\tilde{P}(\tilde{Y})$. The quantile $Q(w)$ with respect to a probability w $(0 \leq w \leq 1)$ is defined implicitly by the relation

$$w = \int_{-\infty}^{Q(w)} \tilde{P}(\tilde{Y})\, d\tilde{Y}. \tag{6.12}$$

For illustration, the random variable $\tilde{Y}$ is smaller than the quantile $Q(w)$ with a probability of w. This situation is illustrated in Figure 6.4. Discretization in QA is based on

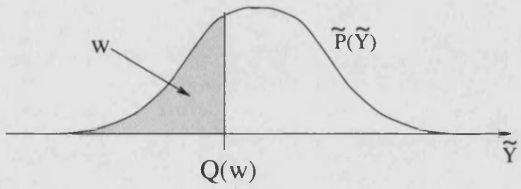


Figure 6.4: Probability w and respective quantile $Q(w)$

finding the quantiles $q_1, \ldots, q_N$ of a continuous random variable $\tilde{Y}$ with respect to a given set of probability values $w_1, \ldots, w_N$. Figure 6.5 illustrates the discretization procedure. Starting from the continuous pdf $\tilde{P}(\tilde{Y})$ the probability cumulative function (pcf) $\tilde{P}_{cum}(\tilde{Y})$ is calculated by integration:

$$\tilde{P}_{cum}(\tilde{Y}) = \int_{-\infty}^{\tilde{Y}} \tilde{P}(\tilde{y})\, d\tilde{y} \tag{6.13}$$

The pcf $\tilde{P}_{cum}(\tilde{Y})$ intersects the probabilities w_i at the quantiles $q_i = Q(w_i)$. These quantiles divide the value range of the continuous random variable $\tilde{Y}$ into N separate intervals. The probability for $\tilde{Y}$ being in the ith interval $[q_{i-1}, q_i]$ is $w_i - w_{i-1}$. This probability is now represented by a discrete probability peak with the weight

$$P_i = w_i - w_{i-1}. \tag{6.14}$$

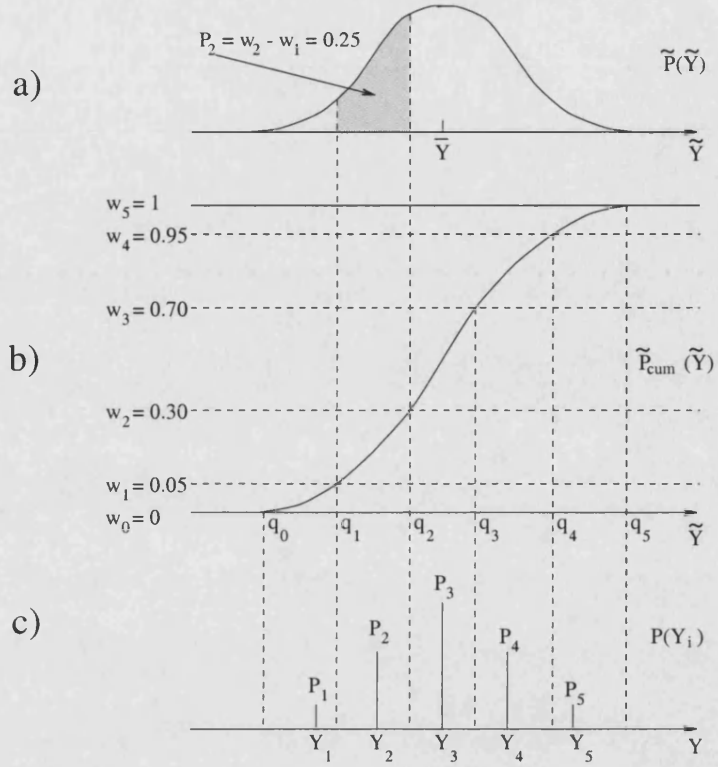


Figure 6.5: Discretization of a random variable $\tilde{Y}$ [211]
a) continuous pdf $\tilde{P}(\tilde{Y})$
b) continuous probability cumulative function (pcf) $\tilde{P}_{cum}(\tilde{Y})$
c) discrete pdf $P(Y_i)$

A sensible choice for the position Y_i of the peak is the mean value of $\tilde{Y}$ within the interval $[q_{i-1}, q_i]$:

$$Y_i = \frac{\int_{q_{i-1}}^{q_i} \tilde{P}(\tilde{Y})\, \tilde{Y}\, d\tilde{Y}}{\int_{q_{i-1}}^{q_i} \tilde{P}(\tilde{Y})\, d\tilde{Y}}. \tag{6.15}$$

The result of this procedure is a N-point discrete random variable

$$Y = (Y_1, Y_2, \ldots, Y_N), \tag{6.16}$$

with its discrete pdf

$$P(Y) = (P_1, P_2, \ldots, P_N). \tag{6.17}$$

It is important to note that using this discretization procedure, the discrete pdf $(P_1, \ldots, P_N)$ is the same for all random variables because of equation (6.14). Only the positions Y_i need

to be calculated for each random variable Y individually.

Finally, the question about the choice of the probabilities w_i arises. These probabilities determine the quantiles $q_i = Q(w_i)$ and consequently the lengths of the discretization intervals $[q_{i-1}, q_i]$. For best approximation of a continuous random variable, all discretization intervals should have similar lengths[1]. Since the quantiles are influenced by the actual pdf, our problem can be formulated as follows: *how to choose the probabilities $w_1, \ldots w_N$ that for most random variables the quantiles $q_1, \ldots, q_n$ are approximately equidistant?*

A heuristic solution of this problem relies on the observation that in the case that a lot of influences interact statistically, the result is mostly described by a normal (or at least by a nearly normal) pdf[2]. For this reason, the probabilities w_i are chosen with reference to a normal pdf with the following two steps.

Definition of Cut-Off Points

The normal pdf with mean value $\overline{y}$ and standard deviation σ is given by

$$\tilde{P}_{norm}(\tilde{Y}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\tilde{Y}-\overline{y})^2}{2\sigma^2}}. \tag{6.18}$$

The low and high cut-off points for discretization are defined by

$$y_{low} = \overline{y} - d \cdot \sigma \tag{6.19}$$

$$y_{high} = \overline{y} + d \cdot \sigma \tag{6.20}$$

where in most cases, $d = 3$ is a sensible choice.

[1] An alternative to the quantile based discretization described above is to apply directly equidistant discretization. The advantage of the quantile based discretization is, that by this means, non-linear dependencies between QSOE expressions can be better described than by equidistant discretization (see [211]).

[2] The mathematical background of this observation is the central limit theorem of statistics [212].

Definition of Probabilities w_i with help of $\tilde{P}_{norm}$

The probabilities w_i are defined by sampling the normal pcf at equidistant points within the interval $[y_{low}, y_{high}]$:

$$\text{sampling points:} \quad y_i = (i-1) \cdot \frac{y_{high} - y_{low}}{N-1} + y_{low}, \qquad i = 1 \ldots N, \tag{6.21}$$

$$\text{probabilities:} \quad w_i = \tilde{P}_{norm,cum}(y_i) = \int_{-\infty}^{y_i} \tilde{P}_{norm}(\tilde{Y})\, d\tilde{Y}, \qquad i = 1 \ldots N. \tag{6.22}$$

It can be shown that as a result of this procedure the probability peaks $P_i = w_i - w_{i-1}$ are given by a discrete normal pdf:

$$P_i = \frac{1}{M} e^{-\frac{(i-\bar{i})^2}{2\sigma_d^2}}, \qquad i = 1 \ldots N \tag{6.23}$$

$$\text{with} \quad M = \sum_{i=1}^{N} e^{-\frac{(i-\bar{i})^2}{2\sigma_d^2}}, \tag{6.24}$$

$$\bar{i} = \frac{N+1}{2}, \tag{6.25}$$

$$\sigma_d = \frac{N-1}{2 \cdot d}. \tag{6.26}$$

Example for the Discretization Procedure

For example of the discretization procedure, consider a resistor R with nominal value $\overline{R} = 10k\Omega$ and the standard deviation $\sigma_R = 500\Omega$. Assuming that R is normal distributed and choosing $N = 13$, $R_{low} = \overline{R} - 3\sigma_R$ and $R_{high} = \overline{R} + 3\sigma_R$ results in the discretized pdf shown in Table 6.1.

6.2.2 Step 2: Calculation of Joint-pdf of u and v

Once the statistical pdfs of all circuit parameters have been discretized, QA applies successively step 2 to step 4 for all QSOE expressions to derive their respective pdfs. Given a continuous QSOE expression of the form $z = u \circ v$, the respective discretized relation is $z_{ij} = u_i \circ v_j$, $i, j = 1 \ldots N$. The statistic of this binary combination is described by a 2-dimensional N^2-point pdf P_{ij} which can be interpreted as the discrete joint-pdf $P(u_i, v_j)$

i	0	1	2	3	4	5	6
pcf w_i	0	0.002	0.011	0.038	0.103	0.224	0.400
pdf P_i	-	0.002	0.009	0.027	0.065	0.121	0.176
quantile R_i	-	8.5 kΩ	8.75 kΩ	9.0 kΩ	9.25 kΩ	9.5 kΩ	9.75 kΩ
i	7	8	9	10	11	12	13
pcf w_i	0.600	0.776	0.897	0.962	0.989	0.998	1.000
pdf P_i	0.200	0.176	0.121	0.065	0.027	0.009	0.002
quantile R_i	10.0 kΩ	10.25 kΩ	10.5 kΩ	10.75 kΩ	11.0 kΩ	11.25 kΩ	11.5 kΩ

Table 6.1: Discretized statistical pdf of resistor R with $\overline{R} = 10k\Omega$, $\sigma_R = 500\Omega$, $N = 13$, $R_{low} = \overline{R} - 3\sigma_R$ and $R_{high} = \overline{R} + 3\sigma_R$

of the variables u and v

$$P_{ij} = P(u_i, v_j). \tag{6.27}$$

Generally, the joint-pdf $P(u, v)$ must be in accordance with the respective marginal pdfs $P_u(u)$ and $P_v(v)$ describing the random variables u and v separately. Introducing the notations $P_u(i)$ and $P_v(j)$ for the discrete pdfs of u and v respectively yields

$$P_u(i) = \sum_{j=1}^{N} P(u_i, v_j) = \sum_{j=1}^{N} P_{ij} \tag{6.28}$$

$$P_v(j) = \sum_{i=1}^{N} P(u_i, v_j) = \sum_{i=1}^{N} P_{ij} \tag{6.29}$$

Moreover, realistic tolerance analysis needs to take into consideration the statistical dependencies between the expressions u and v. These dependencies can be described by the linear correlation coefficient

$$\rho_{uv} = \frac{\sum_{i,j=1}^{N} P_{ij}(u_i - \overline{u})(v_j - \overline{v})}{\sigma_u \, \sigma_v}, \tag{6.30}$$

where $\overline{u}$, $\overline{v}$ are the mean values and σ_u, σ_v the standard deviations of the variables u and v respectively. The mean value $\overline{u}$ and standard deviation σ_u of a discrete random variable u are defined in a similar way to the continuous case:

$$\overline{u} = \sum_{i=1}^{N} P_u(i) \, u_i, \tag{6.31}$$

$$\sigma_u^2 = \sum_{i=1}^{N} P_u(i)\,(u_i - \overline{u})^2. \tag{6.32}$$

Altogether, for realistic tolerance analysis the joint-pdf P_{ij} needs to satisfy the three equations (6.28) - (6.30). The conditions do not determine the joint-pdf P_{ij} completely[3] and there remain some degrees of freedom. In [183] the following solution has been selected

$$P_{ij} = (1 - |\rho_{uv}|) \cdot W_0(i,j) + |\rho_{uv}| \cdot W_1(i,j)\,, \tag{6.33}$$

where

$$W_0(i,j) = P_u(i)P_v(j)\,, \tag{6.34}$$

$$W_1(i,j) = \begin{cases} P_u(i)\,\delta(i-j) & : \quad \rho_{uv} \geq 0 \\ P_u(i)\,\delta(N+1-i-j) & : \quad \rho_{uv} < 0 \end{cases}\,, \tag{6.35}$$

and $\delta(i)$ is the Kronecker delta

$$\delta(i) = \begin{cases} 1 & : \quad i = 0 \\ 0 & : \quad \text{otherwise} \end{cases}\,. \tag{6.36}$$

This solution can be interpreted as a superposition of the pdfs W_0 and W_1 describing uncorrelated and completely correlated variables respectively. The determination of this solution requires the knowledge of the correlation coefficient ρ_{uv}. The correlation coefficients of the circuit parameters are supposed to be known from fabrication data. The correlations between QSOE expressions require detailed analysis which is discussed in Appendix A.1.

6.2.3 Step 3: Calculation of z-Values

Step 3 of QA is to calculate the values of the random variable

$$z_{ij} = u_i \circ v_j \tag{6.37}$$

at the N^2 discretization points $i, j = 1 \ldots N$. The results of step 2 and step 3 are N^2 pairs (z_{ij}, P_{ij}), describing the values of the random variable z by a N^2-point discrete pdf.

[3] Conditions (6.28) - (6.30) provide $2N+1$ equations while P_{ij} has N^2 degrees of freedom. Consequently, for $N \geq 3$ the degree of freedom is larger than the number of conditions.

6.2.4 Step 4: Reduction to N-point Discrete Variable

In step 4, the N^2-point pdf obtained by step 2 and step 3 is reduced to a corresponding N-point pdf. This reduction will be accomplished following the route:

- sorting of the N^2 pairs (z_{ij}, P_{ij}) with respect to increasing z_{ij}.
- Combination of neighboured probability peaks.

The sorting of the N^2 pairs (z_{ij}, P_{ij}) is accomplished with help of the Quick Sort or Heap Sort algorithm [213]. The first one requires $O(K^2)$, the second one $O(K \log_2 K)$ computational steps to sort K elements. Since $K = N^2$, the complexity of Quick Sort is $O(N^4)$ and of Heap Sort is $O(N^2 \log_2 N^2)$. In practical examples, it turns out that for small N Quick Sort is the faster algorithm. However, for larger N, Heap sort becomes faster due to the lower O-complexity. The result of the sorting is a sequence of probability peaks (z'_l, P'_l), $l = 1 \ldots N^2$, where $z'_{l_1} \leq z'_{l_2}$ for $l_1 < l_2$. Neighboured peaks are now combined in a suitable way to reduce the N^2 peaks to a N-point pdf of the form

$$Z = (Z_1, \ldots, Z_N), \tag{6.38}$$

$$P(Z) = (P_1, \ldots, P_N). \tag{6.39}$$

The details concerning the combination of neighboured probability peaks are presented in Appendix A.2.

6.3 Limitations of Quantile Arithmetic

Quantile Arithmetic as presented in the last section is approximately one order of magnitude faster than Monte Carlo analysis and shows in many cases comparable results [211]. However, QA has two limitations: on the one hand, QA reveals a lack of accuracy and on the other hand QA still requires long computation time. These limitations are now explained.

6.3.1 Accuracy

The QA approach described in the last section still has a lack of accuracy. For demonstration purposes, the resistive voltage divider circuit of Figure 6.3 (Page 146) will be analysed. The circuit parameters R_1 and R_2 are assumed to be statistically uncorrelated, normal distributed with the nominal value $\overline{R} = 10\,k\Omega$ and the standard deviation of $\sigma = 500\,\Omega$. Applying a 13-point discretization with the cut-off points $R_{low} = \overline{R} - 3\sigma_R$ and $R_{high} = \overline{R} + 3\sigma_R$ yields for R_1 and R_2 the discrete pdf already shown in Table 6.1.

QA has been implemented in MAPLE V [172] and applied to the QSOE (eq. 6.11, Page 146) describing the transfer function of the voltage divider. A Monte Carlo analysis using $S = 5000$ samples is performed for comparison. The results of tolerance analysis are summarized in Table 6.2 and illustrated in Figure 6.6. Two conclusions can be derived from this comparison:

1. The QA predictions for the quantiles q_i close to the centre of the pdf ($i = 5, 6, 8, 9$) reveal a too small deviation $\Delta H = H - \overline{H}$ from the mean value $\overline{H} = 0.5$. As a result, the centre peak of the pdf of H is predicted by QA as too high and too narrow.

2. The quantiles q_i which refer to the outer corners of the pdf ($i = 1, 2, 12, 13$) are predicted by QA to have a too large deviation from the mean value $\overline{H}$. As a result, the deviations of the outer corners of the pdf of H are predicted to be too large.

By applying QA to some circuit examples, it has been found that the above observations are almost generally valid. In that consequence, QA of [211] is not very precise in the prediction of high yield (specification limit intersects the outer corners of the pdf) and low yields (specification limit is near to the centre of the pdf).

i	discrete pdf P_i	discrete pcf w_i	quantiles $Q(w_i)$	
			Monte Carlo	QA
1	0.002	0.002	$[-0.057, -0.051]$	-0.078
2	0.009	0.011	$[-0.045, -0.043]$	-0.059
3	0.027	0.038	$[-0.036, -0.034]$	-0.041
4	0.065	0.103	$[-0.027, -0.025]$	-0.023
5	0.121	0.224	$[-0.018, -0.017]$	-0.011
6	0.176	0.400	$[-0.009, -0.008]$	-0.005
7	0.200	0.600	$[-0.000, 0.000]$	0.000
8	0.176	0.776	$[0.008, 0.009]$	0.005
9	0.121	0.897	$[0.017, 0.018]$	0.010
10	0.065	0.962	$[0.026, 0.028]$	0.022
11	0.027	0.989	$[0.034, 0.037]$	0.043
12	0.009	0.998	$[0.043, 0.049]$	0.064
13	0.002	1.000	$[0.054, 0.064]$	0.088

Table 6.2: Statistical results for transfer function H of voltage divider
Quantiles $Q(w_i)$ refer to the deviation from mean value: $\Delta H = H - \overline{H}$
Monte Carlo results are given as 95% confidence intervals [214]

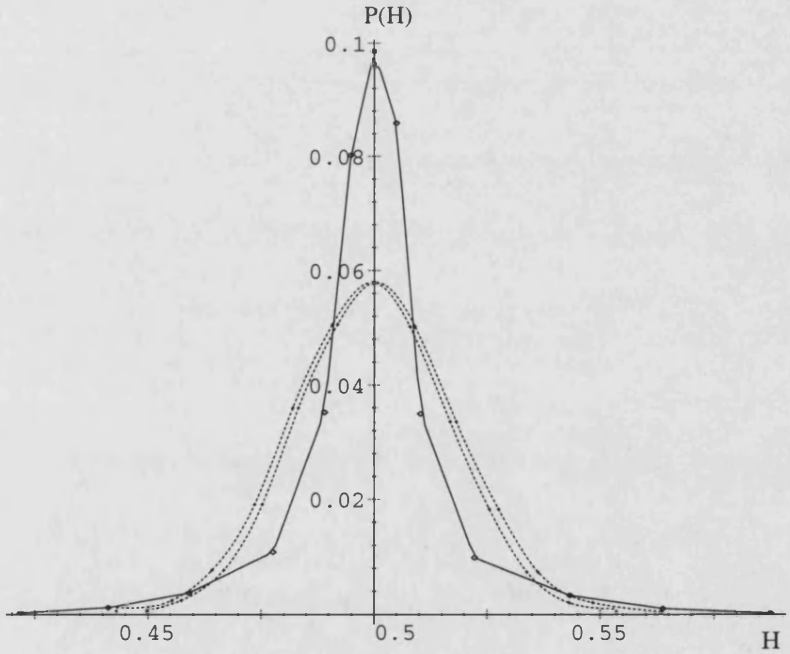


Figure 6.6: Pdf of transfer function H of voltage divider
solid line: Quantile Arithmetic [183]
dotted lines: Monte-Carlo results (95% confidence limits)

6.3.2 Computation Time

QA is approximately 10 times faster than Monte Carlo analysis. Nevertheless, QA as defined in [183] still requires considerable computation time. A balance has been made up concerning the computational requirements of the different steps of QA. The balance sheet is shown in Table 6.3. Obviously, the bottleneck of QA is step 4 because of the

QA Step	Computation time in dependence on number of discretization points N	Part of total computation time (for $N = 13$)
1	$O(N)$	$< 1\%$
2	$O(N^2)$	$\approx 4\%$
3	$O(N^2)$	$\approx 4\%$
4	$O(N^2 \log_2 N^2) \ldots O(N^4)$	$> 90\%$

Table 6.3: Balance sheet for computation time required by the different QA steps

computational expense of sorting. Any improvement concerning the speed of QA needs therefore to by-pass the straight forward sorting approach within step 4.

6.4 Modified Quantile Arithmetic (MQA)

In this section, QA is modified to improve its performance concerning the two limitations presented above. Firstly, the reason for the lack of inaccuracy of QA is explained. After that, a modification of QA will be proposed to increase precision of tolerance analysis. The derived Modified Quantile Arithmetic will be applied to an example to demonstrate the achieved improvement. Finally, a QA specific sorting algorithm is developed to speed up the computations of step 4.

6.4.1 Origin of Inaccuracy of Quantile Arithmetic

In the approach of [183] to QA, the joint-pdf $P_{ij} = P(u_i, v_j)$ of two statistical variables u and v is assumed to be given by a superposition of the uncorrelated and fully correlated case (eq. 6.33). This assumption is correct for $\rho_{uv} = \pm 1$ and $\rho_{uv} = 0$. However, in the cases $-1 < \rho_{uv} < 0$ and $0 < \rho_{uv} < 1$, equation (6.33) is just an approximation. To analyse the quality of this approximation, the voltage divider example is considered. The joint-pdf of the statistical variables R_2 and $z_1 = R_1 + R_2$ is analysed. These variables are combined in the QSOE (eq. 6.11) to the transfer function $H = z_2 = R_2/z_1$. Assuming the same statistical properties for R_1 and R_2 as before, the correlation coefficient of R_2 and z_1 is $\rho_{R_2 z_1} = \frac{1}{\sqrt{2}} \approx 0.71$. In Figure 6.7, scatter plots of the R_2-z_1 joint-pdf are illustrated. The

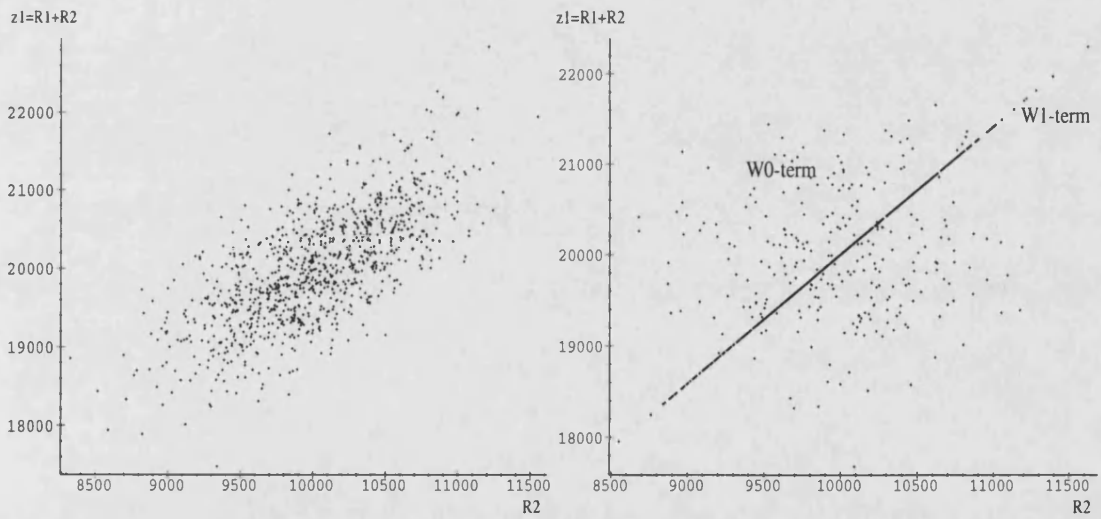


Figure 6.7: Scatter plots of $R2$-z_1 joint-pdf
left hand side: Monte Carlo analysis
right hand side: Quantile Arithmetic

plot on the left hand side has been generated by Monte Carlo analysis, the one on the right hand side shows a scatter plot according to the QA joint-pdf of equation (6.33). The comparison shows that QA strongly overweights the straight line which refers to ideal linear

correlation. The statistical weights of this line originate from the W_1-term in equation (6.33). Additionally, it can be seen that regions which are near to the ideal correlation line are underweighted and peripheral regions are overweighted by the W_0 contributions compared to the Monte Carlo analysis.

The consequences for deriving the pdf of $H = z_2 = R_2/z_1$ are explained with help of Figure 6.8. The solid lines are the equi-value contours which refer to constant values of

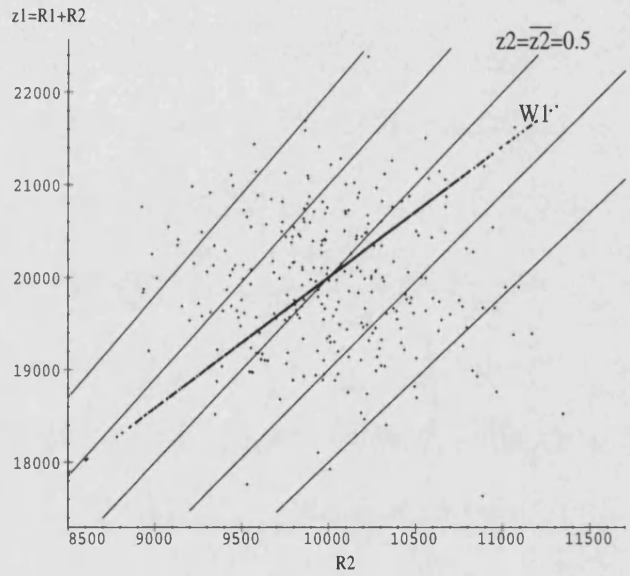


Figure 6.8: QA scatter plot of $R2$-z_1 joint-pdf and equi-value contours of z_2 (solid lines)

z_2. These contours are almost parallel to the line which refers to ideal correlation and the statistical weights of the W_1-term mostly contribute to z_2 values near to the mean value $z_2 = \overline{z_2} = 0.5$ (middle contour line). As a result, the centre peak of the pdf of H (see Figure 6.6) is predicted to high by QA. The other inaccuracies of QA may be clarified similarly.

6.4.2 Modification of Quantile Arithmetic

Improvements to the classical QA are achieved by a modification of this procedure concerning a more realistic approximation for the joint-pdf $P_{ij} = P(u_i, v_j)$. Within QA, this joint-pdf shows a strong discontinuity as it strongly overweights a singular line referring to ideal correlation. A more realistic, i.e. a more continuous, approximation of the joint-pdf can be generated using properties of multivariate normal (Gaussian) pdfs. According to the central limit theorem [212], the statistics of many random processes can be described by a normal or at least nearly normal pdf, especially in the case when many influences interact statistically. This situation applies well to the analysis of large scale circuits where the respective QSOE will exhibit many expressions which are influenced by many circuit parameters. The newly derived Modified Quantile Arithmetic (MQA) [184] is therefore based on multivariate normal pdfs.

Referring to equations (6.28) - (6.30), the joint-pdf $P_{ij} = P(u_i, v_j)$ must fulfil the condition set

$$\begin{aligned} \sum_{j=1}^{N} P_{ij} &= P_u(i), \\ \sum_{i=1}^{N} P_{ij} &= P_v(j), \\ \frac{\sum_{i,j=1}^{N} P_{ij}(u_i - \overline{u})(v_j - \overline{v})}{\sigma_u \, \sigma_v} &= \rho_{uv}. \end{aligned} \tag{6.40}$$

Assuming that u and v are normal distributed, a multivariate pdf obeying these conditions can be found using an axis transformation in the parameter space

$$\begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{A} \begin{pmatrix} x \\ y \end{pmatrix}, \tag{6.41}$$

where $\mathbf{A}$ is a 2×2 matrix. An example for such a transformation is the "Cholesky Decomposition" [215] which chooses the matrix $\mathbf{A}$ in such a way, that the new statistical variables (x, y) are described by an uncorrelated multivariate normal pdf. The Cholesky

Decomposition is best suited for dealing with continuous normal pdfs. For application to discrete pdfs, this approach is modified. Since the discrete random variables u_i and v_j are described in terms of the indices i and j, any transformation concerning discrete random variables acts in the (i, j)-index space. The following approach will be considered:

$$\begin{aligned} \begin{pmatrix} i \\ j \end{pmatrix} &= \mathbf{A} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \qquad \text{with} \quad \mathbf{A} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \\ \begin{pmatrix} \alpha \\ \beta \end{pmatrix} &= \mathbf{A}^{-1} \begin{pmatrix} i \\ j \end{pmatrix}, \qquad \text{with} \quad \mathbf{A}^{-1} = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \end{aligned} \tag{6.42}$$

This transformation yields new indices α and β and can be interpreted as a 45° rotation in index space as illustrated in Figure 6.9. Similarly to the Cholesky Decomposition, the

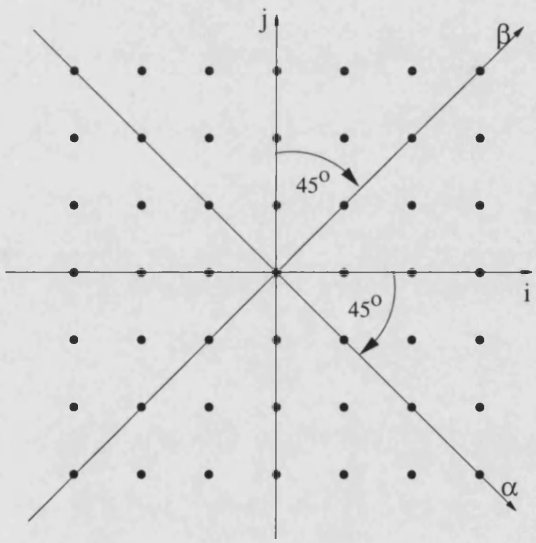


Figure 6.9: Transformation of indices i and j to the new indices α and β

abstract random variables referring to the new indices α and β are assumed to be statistically uncorrelated with the discrete pdfs P_α and P_β respectively. Then, the joint-pdf P_{ij} of the old variables u_i and v_j is given by

$$\begin{aligned} P_{ij} &= P_\alpha \cdot P_\beta \\ &= P_{\alpha(i,j)} \cdot P_{\beta(i,j)}, \end{aligned} \tag{6.43}$$

where $\alpha(i,j) = i - j$ and $\beta(i,j) = i + j$ according equation (6.42). The question is how to choose P_α and P_β to fulfil the condition set (6.40).

To find the solution to this problem equation (6.43) is used in (6.40). After some algebraic manipulation, which is reported in Appendix A.3, the following condition set for the new pdfs P_α and P_β is obtained:

$$P_u(i) = \sum_{j=1}^{N} P_{ij} = \sum_{j=1}^{N} P_{\alpha(i,j)} \cdot P_{\beta(i,j)} = \cdots = (P_\alpha * P_\beta)\,(2i), \tag{6.44}$$

$$P_v(j) = \sum_{i=1}^{N} P_{ij} = \sum_{i=1}^{N} P_{\alpha(i,j)} \cdot P_{\beta(i,j)} = \cdots = (P_{\overline{\alpha}} * P_\beta)\,(2j), \tag{6.45}$$

$$\rho_{uv} = \frac{\sum_{i,j=1}^{N} P_{ij}(u_i - \overline{u})(v_j - \overline{v})}{\sigma_u\,\sigma_v} = \cdots = \frac{\sigma_\beta^2 - \sigma_\alpha^2}{\sigma_\beta^2 + \sigma_\alpha^2}, \tag{6.46}$$

where $P_{\overline{\alpha}}(k) = P_\alpha(-k)$ and "$*$" means convolution. Herein, the mean value $\overline{\alpha}$ and the standard deviation σ_α of an index are defined as $\overline{\alpha} := \sum_\alpha P_\alpha\,\alpha$ and $\sigma_\alpha^2 := \sum_\alpha P_\alpha(\alpha - \overline{\alpha})^2$ respectively. The solution of the condition set is based on the knowledge of the original pdfs $P_u(i)$ and $P_v(j)$ and of the correlation ρ_{uv}. According to Section 6.2.1, all random variables are described by the same set of probability peaks $(P_1, \ldots, P_N)$. Consequently $P_u(i)$ and $P_v(j)$ are of the form

$$P_u(i) = P_v(i) = P_i, \tag{6.47}$$

where P_i is a discretized normal pdf according to equation (6.23) (Page 150). Since the convolution "$*$" of two normal pdfs results again in a normal pdf, equations (6.44) and (6.45) can be solved by choosing P_α and P_β as normal pdfs. The details of the solution of the whole equation set (6.44) – (6.46) are described in Appendix A.4. In Table 6.4 the properties of the solutions P_α and P_β are summarized.

pdf	discretization points	mean value	standard deviation	shape
P_α	$\alpha = -N \ldots N$	$\overline{\alpha} = 0$	$\sigma_\alpha = \sigma_i\sqrt{2(1-\rho_{uv})}$	normal
P_β	$\beta = -N \ldots N$	$\overline{\beta} = 0$	$\sigma_\beta = \sigma_i\sqrt{2(1+\rho_{uv})}$	normal

Table 6.4: Solutions P_α and P_β of the condition set (6.44)-(6.46)

Considerations for Highly Correlated Variables

Direct application of MQA is revealed to be inaccurate in the case that the two random variables u and v are strongly correlated, i.e. $\rho_{uv} > 0.9$ or $\rho_{uv} < -0.9$. For illustration, the correlation ρ_{uv} is assumed to have a target value $\rho_t = 0.95$. By applying MQA to determine $P_{ij} = P(u_i, v_j)$ according to equation (6.43) and using P_α and P_β according to Table 6.4 results in a "real" correlation of $\rho_r = 0.99$. The difference of 0.04 seems not to be significant. However, under certain conditions the pdf of $z = (u, v)$ may depend severely on correct correlation.

Example: $\overline{u} = \overline{v}$, $\sigma_u = \sigma_v$ and $z = u - v$.

Since $z = u - v$ is a purely linear relation the resulting standard deviation for z can be determined by using the RSS formula (6.6). After some algebraic manipulations, the following result is obtained:

$$\sigma_z = 2\sigma_u\sqrt{1-\rho_{uv}}. \tag{6.48}$$

In the case of $\rho_{uv} = 0.95$ this leads to $\sigma_z = 0.2\sigma_u$ while in the case of $\rho_{uv} = 0.99$ the result is $\sigma_z = 0.04\sigma_u$. The tolerances differ by a factor 5!

The reason for correlation inaccuracy is as follows: in the case of high correlation $\rho_{uv} \approx 1$ the standard deviation $\sigma_\alpha = \sigma_i\sqrt{2(1-\rho_{uv})}$ of the index α becomes very small, i.e. smaller than 1 which is the sampling distance of P_α. In such a case, the pdf P_α is narrow compared to the discretization distance which causes a significant discretization error. As a result of this discretization error the target and real standard deviation of α differ $\sigma_r \neq \sigma_t$. This difference causes ρ_{uv} to become too large (see equation 6.46). In

Appendix A.5, a method is developed which removes this discretization error and allows correct treatment of highly correlated variables within MQA.

Improvement of MQA: Voltage Divider Example

To demonstrate the improvements of MQA the voltage divider circuit will be analysed again. The joint-pdf of R_2 and z_1 obtained by Monte Carlo analysis and MQA are shown in Figure 6.10. The MQA scatter plot is very similar to that generated with Monte Carlo

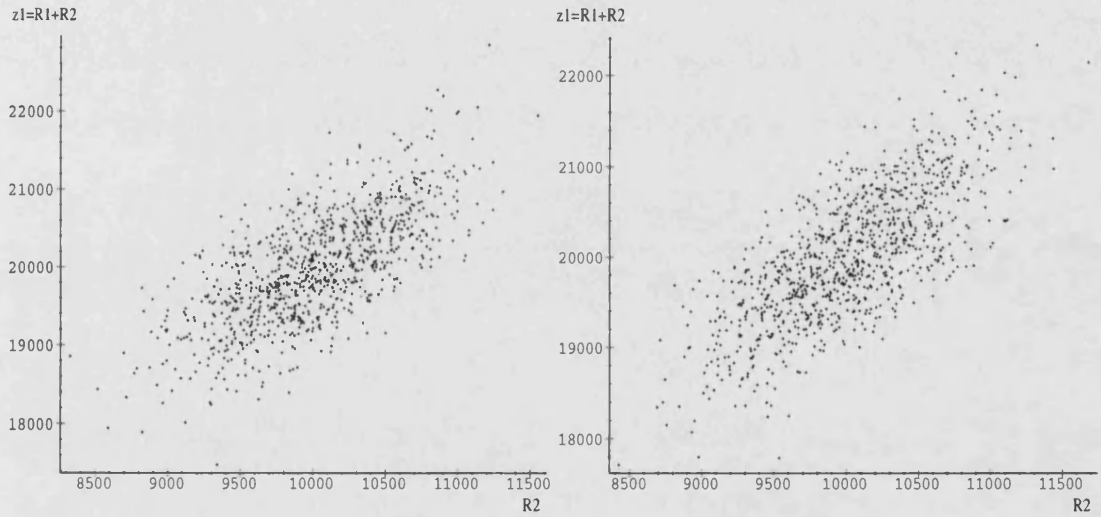


Figure 6.10: Scatter plots of $R2$-z_1 joint-pdf
left hand side: Monte Carlo analysis
right hand side: Modified Quantile Arithmetic

analysis. In Figure 6.11 the statistics of the transfer function H is illustrated. The pdf predicted by MQA is completely within the 95% confidential limits of Monte Carlo analysis as opposed to the pdf predicted by QA (see Figure 6.6, Page 155). Consequently, using MQA instead of QA, a significant improvement in accuracy of yield prediction can be expected.

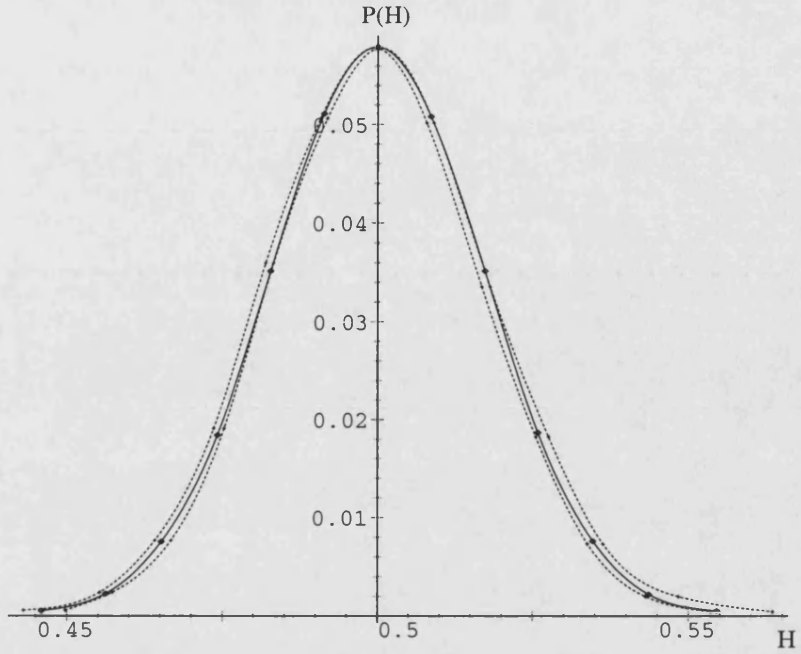


Figure 6.11: Pdf of transfer function H of voltage divider
dotted lines: Monte Carlo analysis (95% confidence limits)
solid line: Modified Quantile Arithmetic

6.4.3 Fast Sorting in Modified Quantile Arithmetic

In Section 6.3.2 it has been found that the bottleneck of MQA is step 4: the sorting of N^2 pairs (z_{ij}, P_{ij}) with respect to increasing z_{ij}. The key to fast sorting within MQA is to take advantage of the monotony of the arithmetic operations $z_{ij} = u_i \circ v_j$. Assuming that the arithmetic operation is an addition, z_{ij} increases with u_i and v_j. As u_i and v_j increase with i and j respectively, z_{ij} increases also with i and j. This property can be used to accelerate the heap sort algorithm. By applying the heap sort directly to MQA, a binary tree with all N^2 z_{ij}-values is created [213]. In this tree, each element is smaller than its two successors. An example is shown in Figure 6.12. After all N^2 z_{ij}-elements have been inserted into the heap, they are successively extracted from the heap in increasing order of the z_{ij}-values. The insertion (and also the extraction) of one element into (from) the heap with N^2 elements is of $\log_2 N^2$ complexity [213]. Consequently, the creation of the

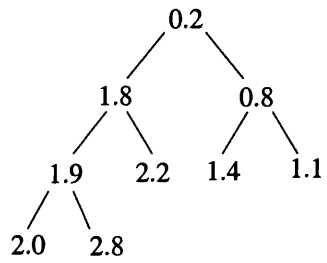


Figure 6.12: A heap (binary tree) for sorting

whole heap is of $N^2 \log_2 N^2$ complexity and the extraction of the elements from the heap is again of $N^2 \log_2 N^2$ complexity. The overall complexity is then given as

$$\text{complexity(heapsort)} = 2N^2 \log_2 N^2 = 4N^2 \log_2 N. \tag{6.49}$$

Compared to other sorting algorithms this can be weighted as rather fast. However, by using the monotony properties of z_{ij}, the heap sort algorithm can be accelerated. Not all N^2 z_{ij}-elements need to be in the heap at once. Obviously, for $z_{ij} = u_i + v_j$ the smallest element is z_{11}, which is known without creating any heap at all. The next larger candidates are z_{12} and z_{21}. These elements are now inserted in a heap. The smaller element is extracted from the root of that heap as the next element of the sorted sequence (e.g. z_{21}) yielding the actual sequence (z_{11}, z_{21}). Then the next candidates which are in accordance with monotony (in our example only z_{31} since z_{13} is smaller than z_{12} in any case because of monotony) are inserted in the remaining heap. This yields a new intermediate heap which is constituted of the elements z_{12} and z_{31}, the smaller one being the root. The procedure

- extraction of the root
- insertion of the next candidates

is successively continued until all elements are sorted and the heap is empty. An intermediate situation within the application of this algorithm is illustrated in Figure 6.13. The

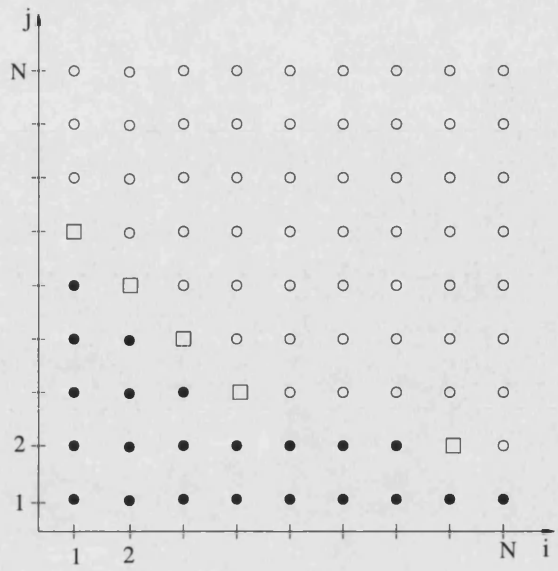


Figure 6.13: Sorting in MQA using monotony with respect to i and j
filled circles: already in ordered sequence
squares: actual heap elements = candidates for next element in sequence
empty circles: not in heap yet and not in sequence yet

maximal number of simultaneous candidates is N. Consequently, the maximal number of elements in the heap is given by N (instead of N^2) using this procedure. As a result, the overall complexity of the accelerated sorting algorithm is

$$\text{complexity(heapsort,accelerated)} = 2N^2 \log_2 N \tag{6.50}$$

which is 2 times faster than the straight forward heap sort algorithm. For other arithmetic operations than addition, similar algorithms have been developed within MQA and always an acceleration by a factor 2 has been achieved.

6.5 Implementation of MQA within Hierarchical Symbolic Analysis

For applying the MQA approach to tolerance analysis of analogue circuits, a symbolic description of the network behaviour is required. Since steps 2 to 4 (see Figure 6.2) are

successively applied to each arithmetic operation, the expense of tolerance analysis is proportional to the number of terms in the symbolic network function.

The hierarchical symbolic approach of [89] can help to provide a highly compact network description, even in the case of large scale circuits. Moreover, the generated SOE is very similar to the required QSOE format introduced in Section 6.2. However, two points need to be considered:

1. MQA operates on real random variables. Consequently, the SOE generated by [89] needs to be decomposed such that the real and imaginary parts of the expressions can be calculated separately. The real and imaginary parts are then treated as two separate random variables as described in [216].

2. MQA operates only on binary symbolic expressions. For this reason the SOE needs to be converted into a suitable format.

   Example: the expression $z = a \cdot b \cdot c$ is converted to the sequence $z_1 = a \cdot b$, $z = z_1 \cdot c$.

A SOE-QSOE interface has been written in MAPLE V [172]. Herein, the SOE generated by the hierarchical symbolic analysis [89] is converted into a QSOE according the two steps described above. Since all arithmetic operations within the SOE are additions, subtractions, divisions and additions, the conversion can be performed automatically. The MQA algorithm as described in the previous sections has also been implemented in MAPLE V. The inputs to this algorithm are the QSOE generated by the SOE-QSOE converter and the parameter pdfs provided by the user. The output of the MQA procedure is the tolerance information of the network function(s) requested by the user. The overall architecture of the proposed symbolic tolerance analysis procedure [184] is shown in Figure 6.14.

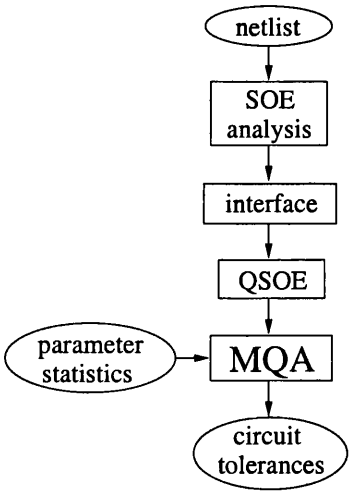


Figure 6.14: Architecture of symbolic tolerance analysis [184]

6.6 Experimental Results

The symbolic MQA approach developed in the previous sections is now applied to practical circuits to examine its efficiency and limitations. For this purpose, comparisons of the circuit tolerances derived with MQA, QA, Monte Carlo analysis and RSS will be performed. Two circuit examples are chosen for benchmarking: an active biquad circuit and the large scale bandpass circuit already examined in Chapter 5. Important criteria for benchmarking are precision on the one hand and efficiency, i.e. computational cost, on the other hand.

6.6.1 Active Biquad Circuit

The biquad circuit to be analysed is shown in Figure 6.15. The operational amplifiers are treated as ideal amplifiers with infinite amplification, zero output and infinite input resistance. Then, the resonant frequency f_{res}, the filter amplification A at the resonant

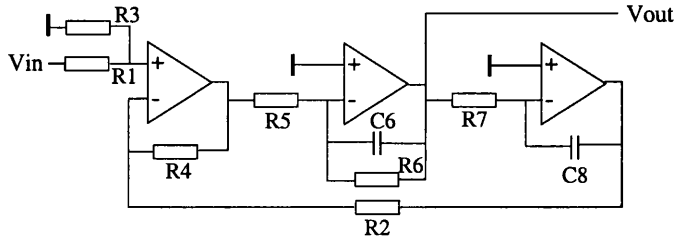


Figure 6.15: Biquad circuit

frequency and the filter Q-value are given as

$$f_{res} = \frac{1}{2\pi}\sqrt{\frac{R_4}{R_5 R_7 R_2 C_6 C_8}} \tag{6.51}$$

$$A = \frac{(R_4 + R_2)R_3 R_6}{(R_1 + R_3)R_5 R_2} \tag{6.52}$$

$$Q = R_6\sqrt{\frac{R_4 C_6}{R_5 R_7 R_2 C_8}} \tag{6.53}$$

The nominal values of the resistors and capacitors are chosen as shown in Table 6.5, which

R_1	R_2	R_3	R_4	R_5	R_6	R_7	C_6	C_8
$10k\Omega$	$10k\Omega$	$10k\Omega$	$10k\Omega$	$10k\Omega$	$10k\Omega$	$10k\Omega$	$1nF$	$1nF$

Table 6.5: Filter parameter values for $f_{res} = 16$kHz, $A = 1$ and $Q = 1$

refers to a resonant frequency of $f_{res} = 16$kHz, amplification $A = 1$ and $Q = 1$. Tolerance analysis is performed assuming the following situation:

- circuit parameters are uncorrelated, normal distributed with a standard deviation of 5% of the respective nominal value.
- MQA and as well QA are applied using a 13-point discretization[4] with the cut-off points $y_{low} = \overline{y} - 3\sigma$ and $y_{high} = \overline{y} + 3\sigma$.
- Monte Carlo analysis with $S = 5000$ samples is performed for comparison purposes.

[4]It has been found out that a discretization with $N = 13$ is sufficient in the case of cut-off points of $\overline{y} \pm 3\sigma$. By using a larger N the accuracy is not improved significantly, however, computing time increases.

All tolerances derived by Monte Carlo analysis will be given in terms of the respective 95% confidence interval according to [214].

Pdf of Transfer Function H at one specific Frequency

Firstly, the magnitude of the transfer function $|H|$ at one specific frequency (30 kHz) is considered. The statistical pdfs predicted by Monte Carlo analysis, RSS[5], QA and MQA are compared. The quantiles $q_i, i = 1 \ldots 13$, of the pdf of the transfer function magnitude $P(|H|)$ predicted by these four methods are shown in Table 6.6. The pdfs obtained by

i	discrete pdf P_i	discrete pcf w_i	quantiles q_i			
			Monte Carlo	RSS	QA	MQA
1	0.002	0.002	$[-0.178, -0.158]$	-0.184	-0.215	-0.150
2	0.009	0.011	$[-0.143, -0.130]$	-0.153	-0.159	-0.127
3	0.027	0.038	$[-0.115, -0.107]$	-0.123	-0.117	-0.105
4	0.065	0.103	$[-0.086, -0.082]$	-0.092	-0.078	-0.081
5	0.121	0.224	$[-0.060, -0.057]$	-0.061	-0.043	-0.056
6	0.176	0.400	$[-0.031, -0.028]$	-0.031	-0.025	-0.030
7	0.200	0.600	$[-0.001, 0.002]$	0.000	-0.002	-0.002
8	0.176	0.776	$[0.030, 0.034]$	0.031	0.023	0.029
9	0.121	0.897	$[0.062, 0.067]$	0.061	0.047	0.061
10	0.065	0.962	$[0.097, 0.103]$	0.092	0.093	0.095
11	0.027	0.989	$[0.132, 0.142]$	0.123	0.154	0.133
12	0.009	0.998	$[0.163, 0.177]$	0.153	0.235	0.175
13	0.002	1.000	$[0.198, 0.238]$	0.184	0.368	0.222

Table 6.6: Comparison of Monte Carlo analysis, RSS, QA and MQA: quantiles of transfer function magnitude $q_i(|H|)$ (deviation from mean value)

RSS, QA and MQA plotted and compared with Monte Carlo analysis in Figure 6.16, 6.17 and 6.18 respectively.

[5]The required sensitivity information is derived using the parallel sensitivity analysis approach of Chapter 5.

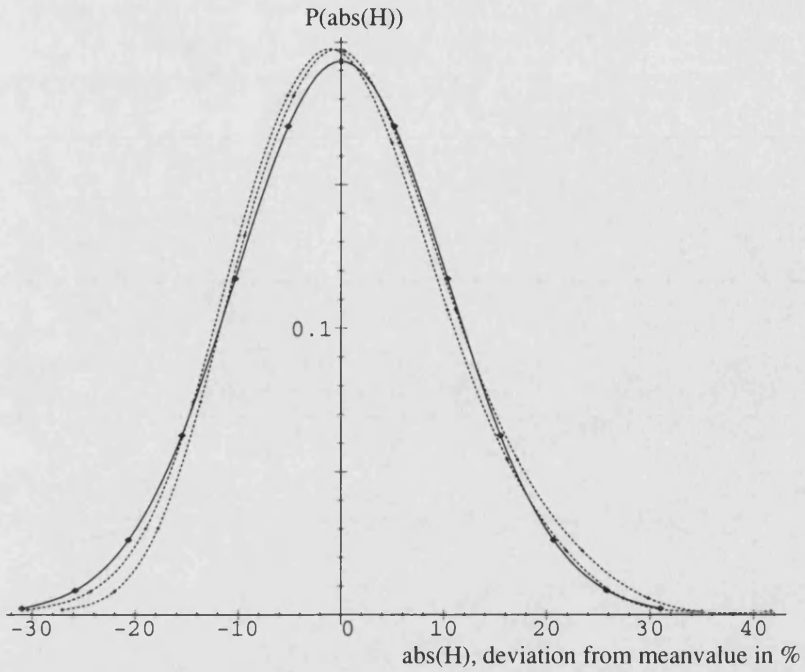


Figure 6.16: Comparison of RSS and Monte Carlo analysis: $P(|H|)$ at 30 kHz
solid line: RSS
dotted lines: Monte Carlo 95% confidence limits

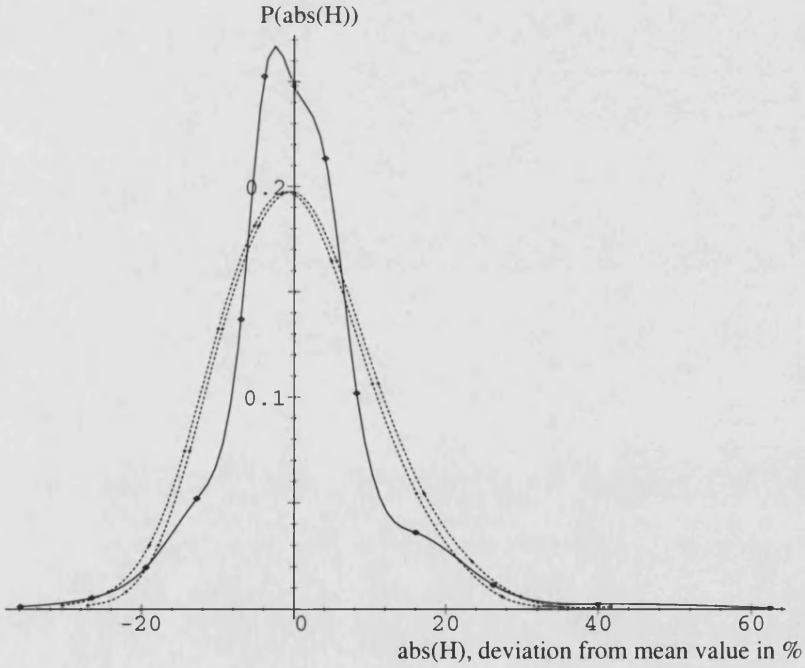


Figure 6.17: Comparison of QA and Monte Carlo analysis: $P(|H|)$ at 30 kHz
solid line: QA
dotted lines: Monte Carlo 95% confidence limits

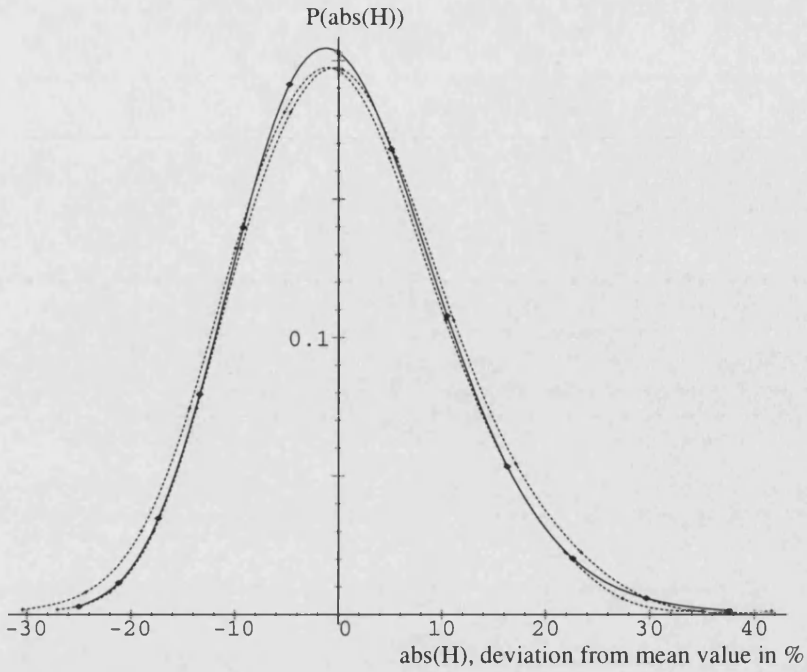


Figure 6.18: Comparison of MQA and Monte Carlo analysis: $P(|H|)$ at 30 kHz
solid line: MQA
dotted lines: Monte Carlo 95% confidence limits

The following observations are made:

- The RSS pdf is in accordance with Monte Carlo analysis for small deviations (up to 15%, quantiles $i = 5, 6, 7, 8, 9$). However, for large deviations ($> 20\%$, $i < 4$ or $i > 10$), RSS and Monte Carlo analysis yield different results. For examples, the RSS quantiles $q_1 = -0.184$ and $q_{13} = 0.184$ lie outside the respective Monte Carlo 95% confidence interval.

- QA shows the over-/undershoot character already reported.

- MQA is in good accordance with Monte Carlo analysis and describes the larger deviations (q_1 and q_{13}) almost exactly.

The reason why RSS is not able to predict the larger deviations of $|H|$ correctly lies in the fact that RSS is restricted to purely linear parameter-performance dependencies. In the above example however, non-linearities occur which can be seen from the fact that

Monte Carlo analysis yields a nonsymmetric performance pdf $P(|H|)$ ($|q_1| \neq |q_{13}|$) in spite of completely symmetric (normal) parameter pdfs. MQA is able to predict this nonsymmetry correctly which shows that it can handle these nonlinearities.

Tolerances of Transfer Function H over Frequency Range

Next, the tolerance behaviour of the transfer function magnitude $|H|$ over varying frequency is analysed. Herein, the following definitions will be used:

- The **negative** and **positive 1σ-tolerances** are defined as the deviations given by the quantiles q_5 and q_9 respectively.
- The **negative** and **positive 2σ-tolerances** are defined as the deviations given by the quantiles q_3 and q_{11} respectively.
- The **negative** and **positive 3σ-tolerances** are defined as the deviations given by the quantiles q_1 and q_{13} respectively.

The names 1σ-, 2σ- and 3σ-tolerances refer to the situation that in the case of normal pdfs, the respective quantiles describe the 1σ-, 2σ- and 3σ deviations (see Section 6.2.1). The 1σ-, 2σ- and 3σ-tolerances of the magnitude of the biquad transfer function versus frequency $|H(freq)|$ are plotted in Figure 6.19, 6.21 and 6.23 respectively. Thereby the QA and MQA results are compared with the Monte Carlo 95% confidence limits. Figure 6.20, 6.22 and 6.24 show the relative 1σ-, 2σ- and 3σ-tolerances respectively. Relative deviations are defined by

$$d(abs(H)) := \frac{|H| - \overline{|H|}}{\overline{|H|}} \tag{6.54}$$

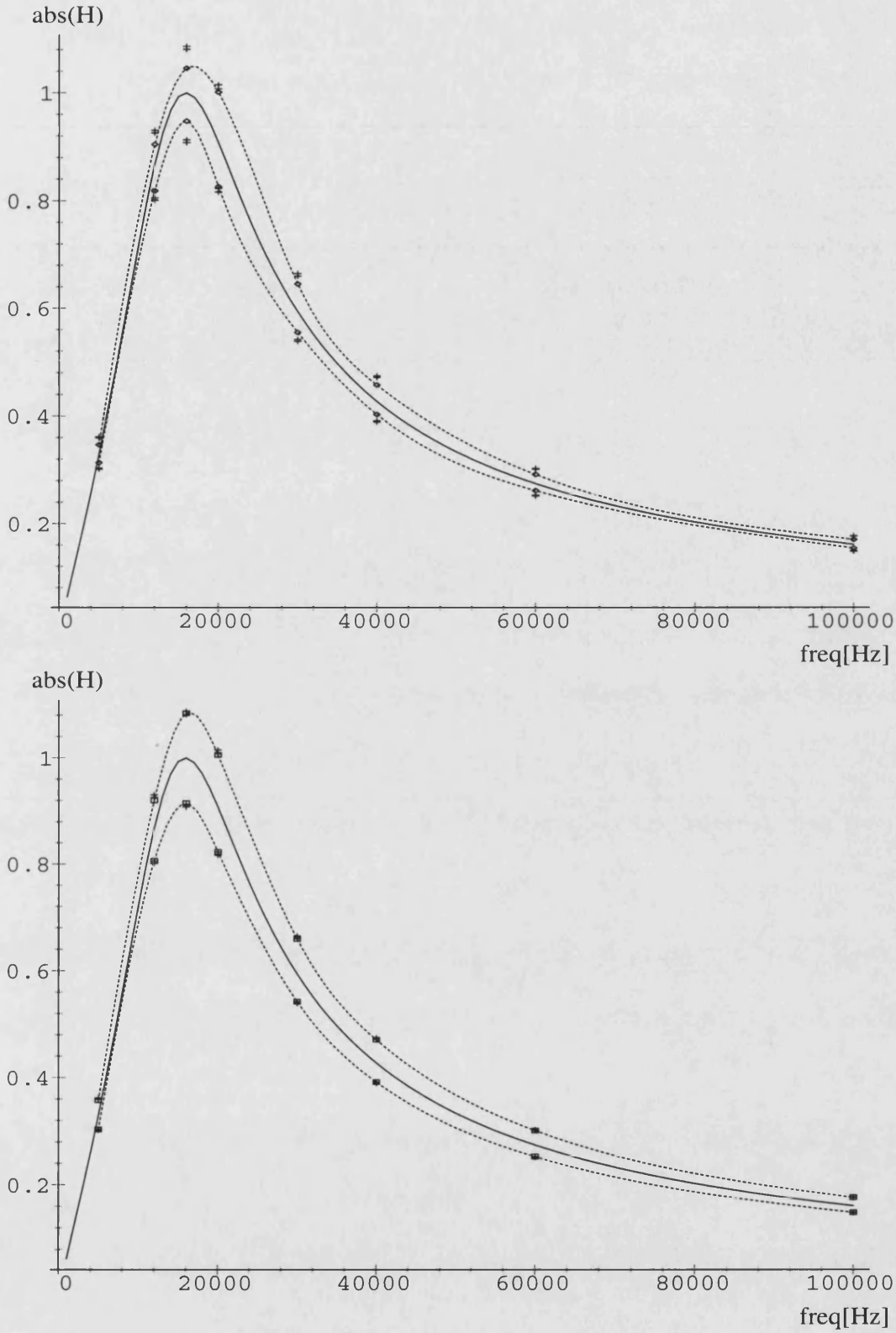


Figure 6.19: **1σ-Tolerances** of $|H|$ derived from QA (above) and MQA (below)

solid line:	nominal value
dotted lines:	1σ-tolerances
crosses:	Monte Carlo 95% confidence limits of 1σ-tolerances

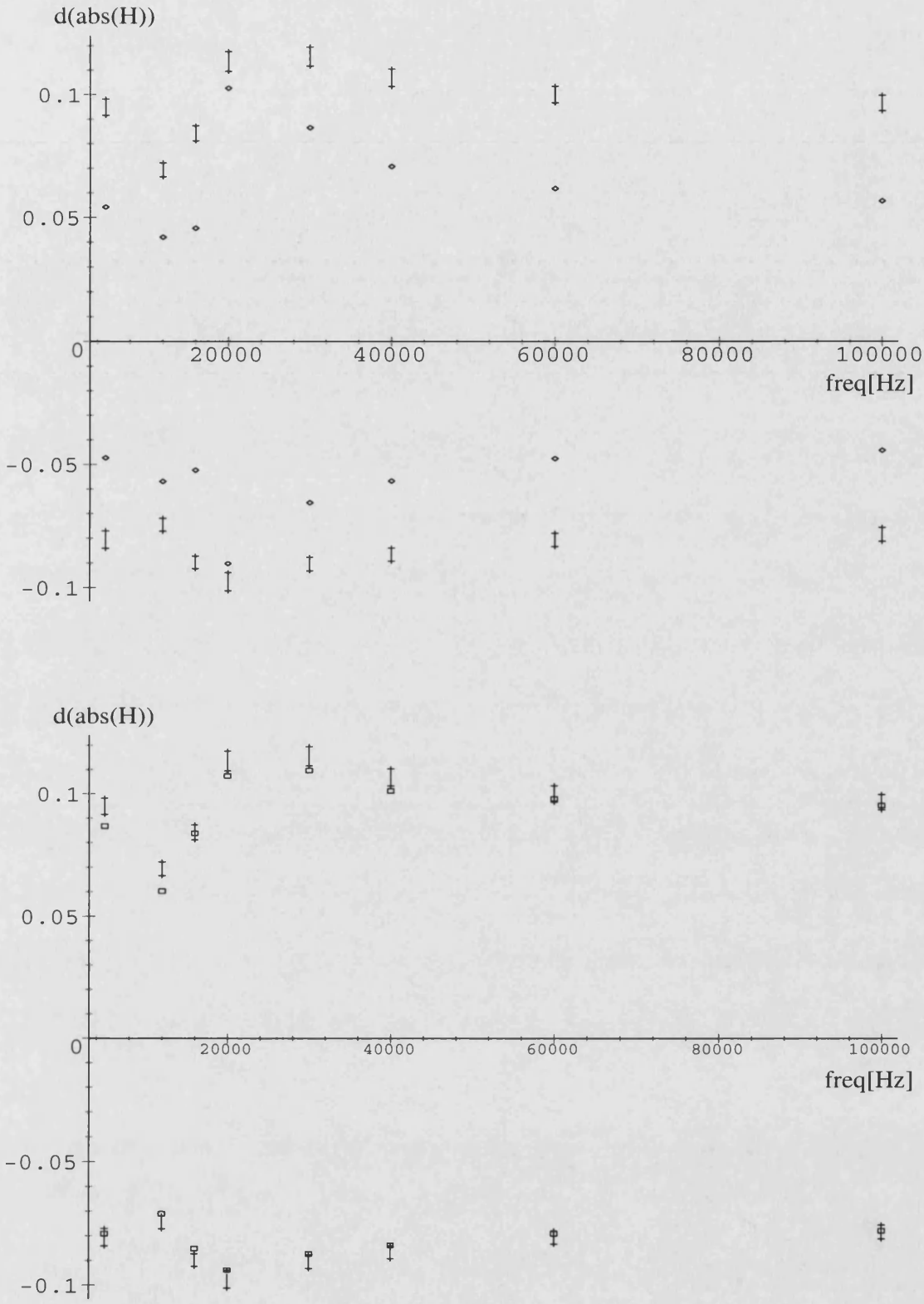


Figure 6.20: Relative **1σ-Tolerances** of $|H|$ derived from QA (above) and MQA (below)
diamonds (above): QA relative 1σ-tolerances
boxes (below): MQA relative 1σ-tolerances
vertical lines: Monte Carlo 95% confidence intervals of 1σ-tolerances

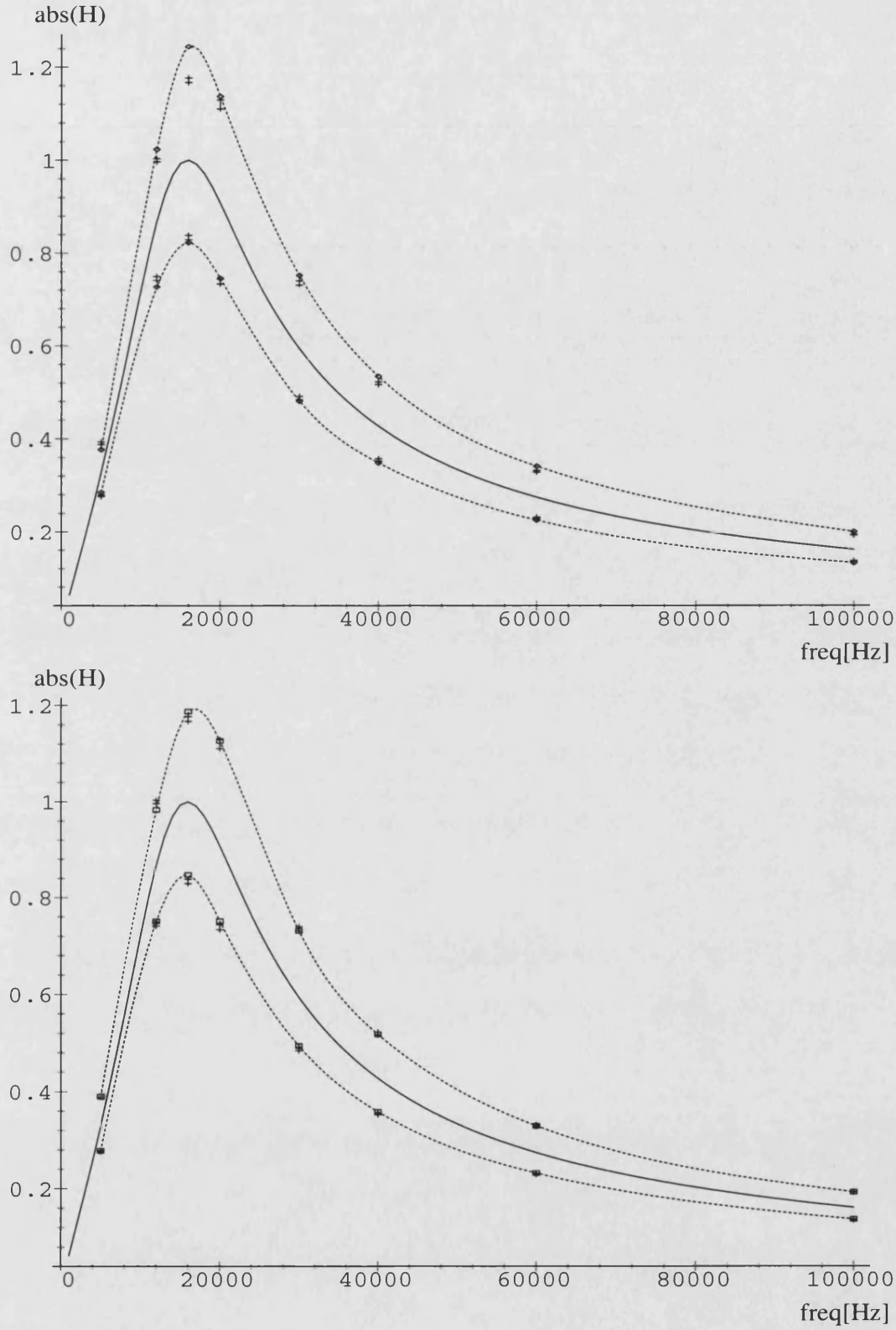
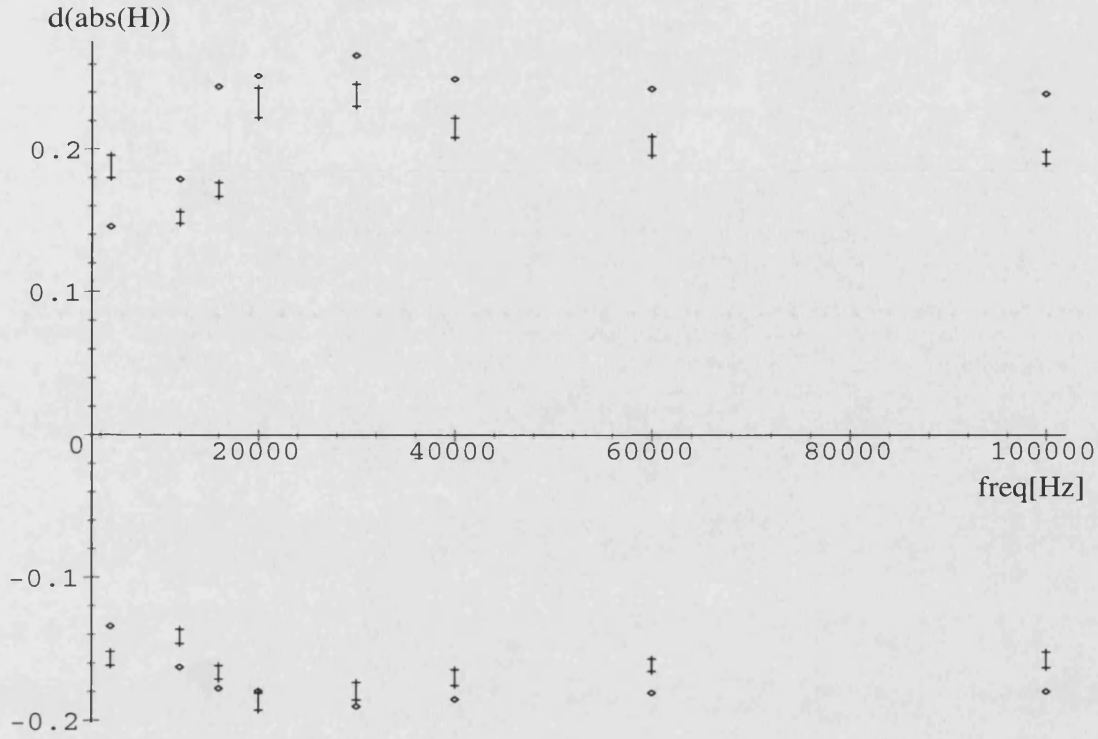


Figure 6.21: **2σ-Tolerances** of $|H|$ derived from QA (above) and MQA (below)

solid line:	nominal value
dotted lines:	2σ-tolerances
crosses:	Monte Carlo 95% confidence limits of 2σ-tolerances



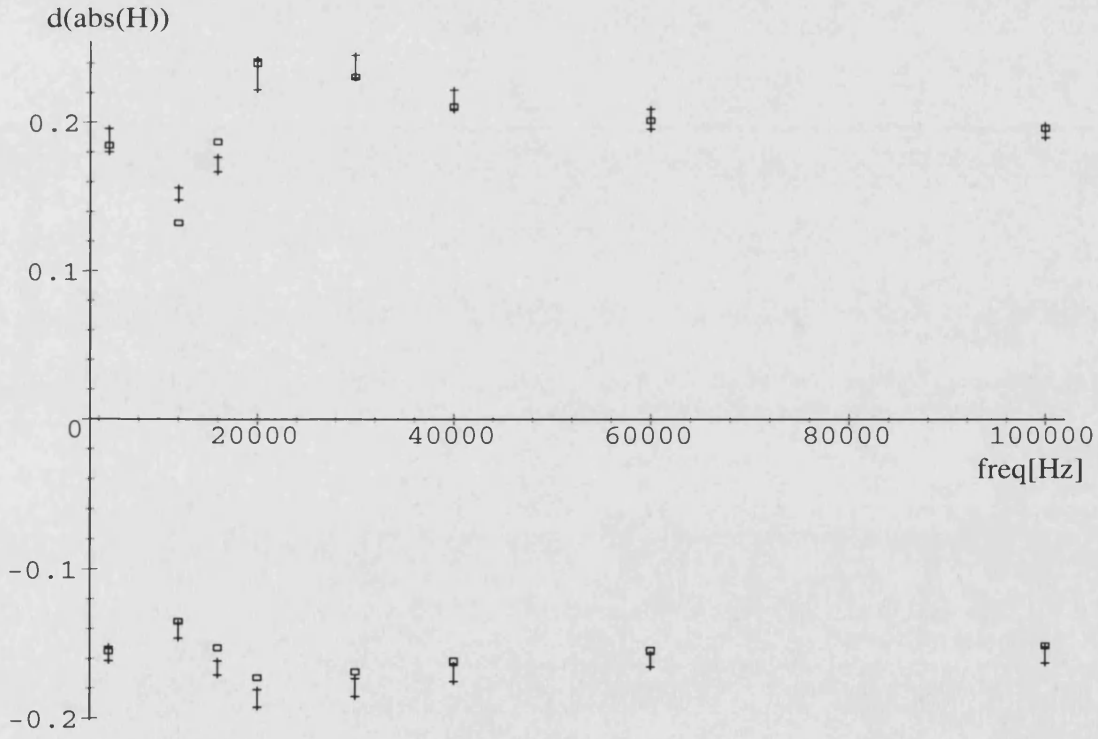


Figure 6.22: Relative **2σ-Tolerances** of $|H|$ derived from QA (above) and MQA (below)
diamonds (above): QA relative 2σ-tolerances
boxes (below): MQA relative 2σ-tolerances
vertical lines: Monte Carlo 95% confidence intervals of 2σ-tolerances

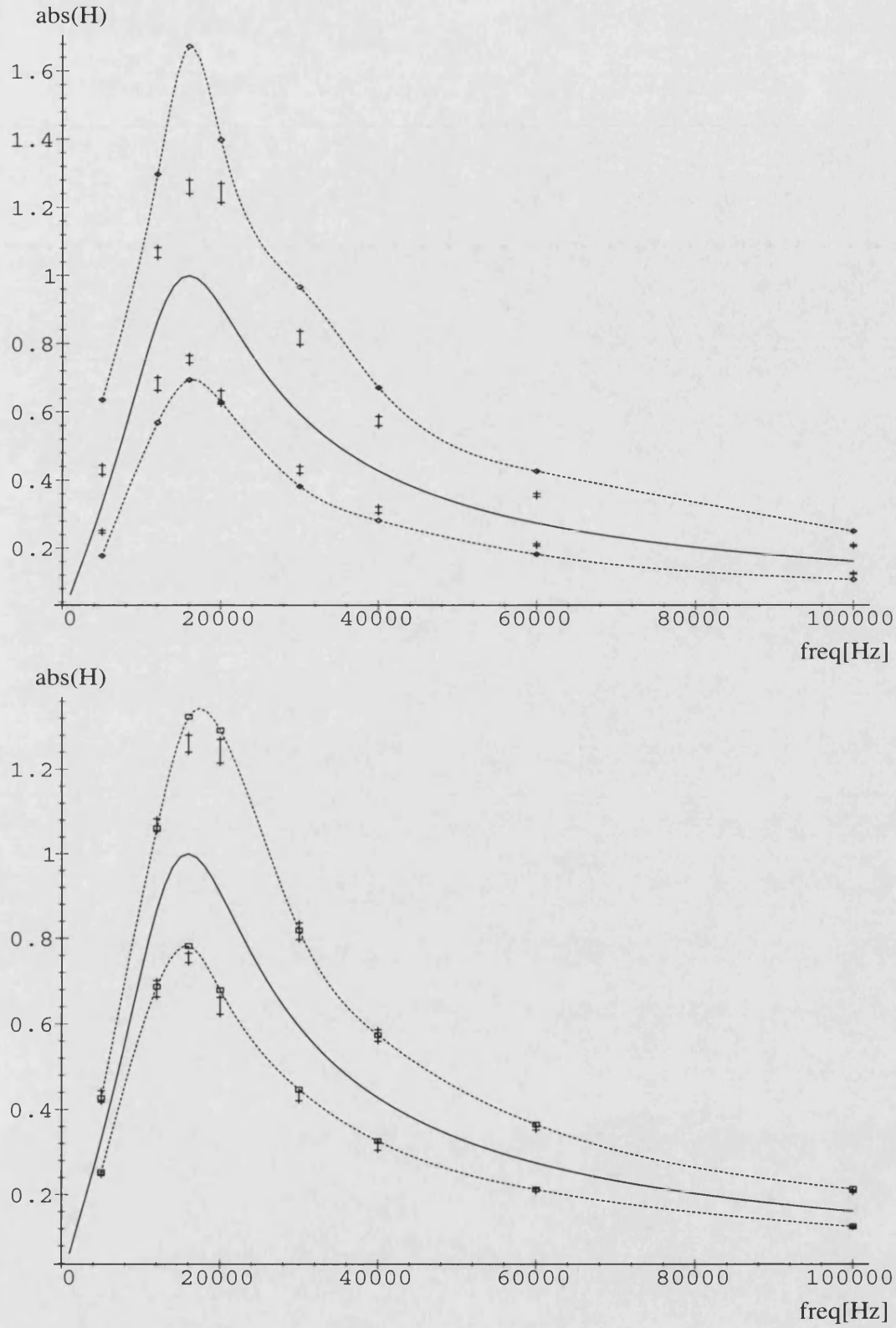


Figure 6.23: **3σ-Tolerances** of $|H|$ derived from QA (above) and MQA (below)

solid line:	nominal value
dotted lines:	3σ-tolerances
crosses:	Monte Carlo 95% confidence limits of 3σ-tolerances

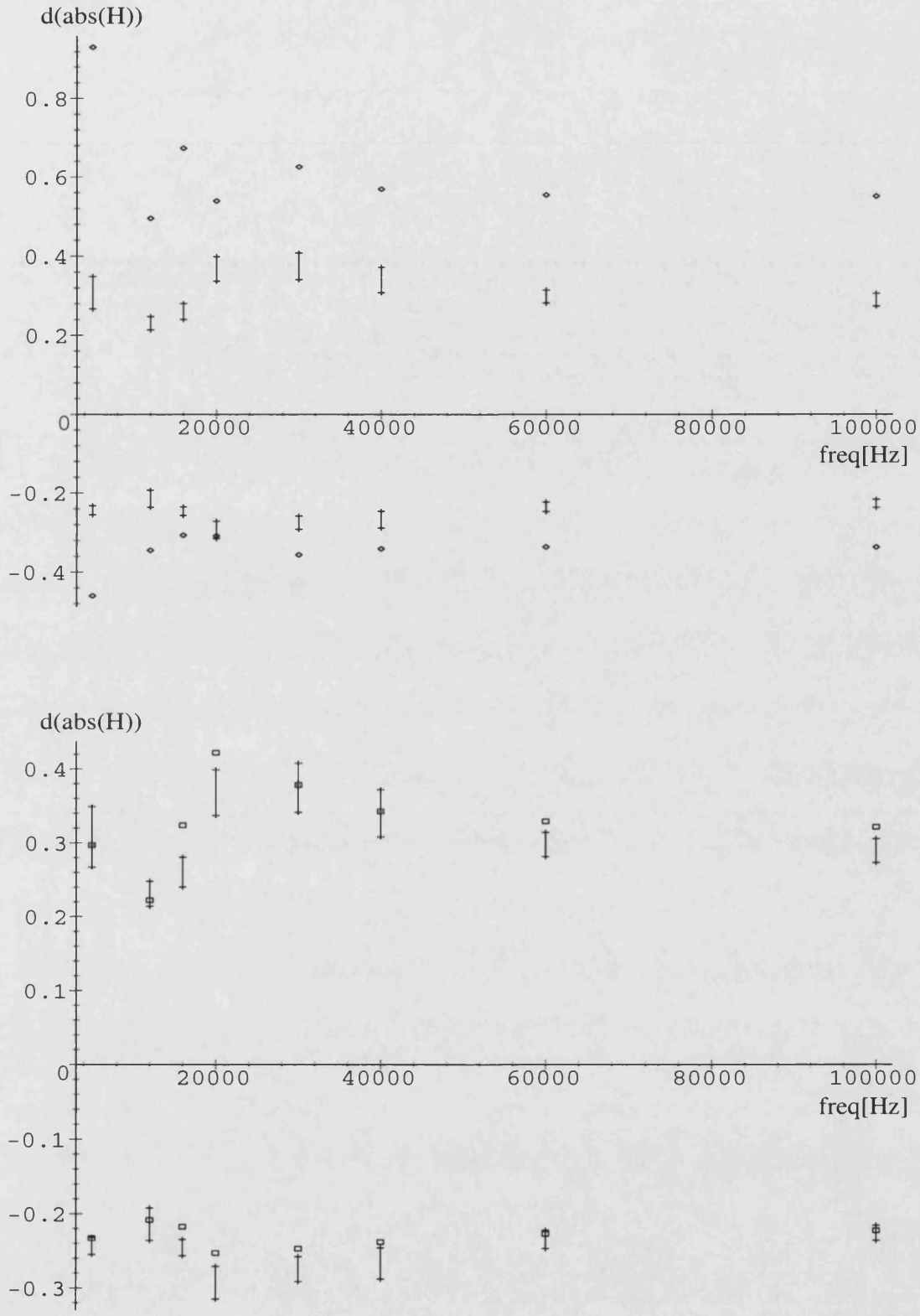


Figure 6.24: Relative **3σ-Tolerances** of $|H|$ derived from QA (above) and MQA (below)

diamonds (above):	QA relative 3σ-tolerances
boxes (below):	MQA relative 3σ-tolerances
vertical lines:	Monte Carlo 95% confidence intervals of 3σ-tolerances

The following conclusions can be drawn from the comparison with Monte Carlo analysis:

- While QA predicts the 2σ-tolerances with moderate accuracy, the 1σ-tolerances are predicted too small. On the other hand, the 3σ-tolerances are predicted too large. This is in accordance with the over/-undershoot character reported previously.

- The tolerances predicted by MQA are in good agreement with Monte Carlo analysis: in most cases the MQA results for the 1σ-, 2σ- and 3σ-tolerances are within or nearby the respective Monte Carlo 95% confidence interval. The reason for the good precision is the realistic approximation of joint-pdfs within MQA.

MQA provides an accurate prediction of the circuit tolerances over the whole range of deviations. Consequently, MQA is suited for yield prediction even if yield is low (specification limit intersects the pdfs near to the mean value) or high (specification limit intersects the outer corners of the pdf).

Comparison of Computing Time Requirements

The plots showing the relative tolerances indicate that even in the case of a Monte Carlo analysis with $S = 5000$ samples, the 95% confidence intervals can have significant extension, especially in case of the 3σ-tolerances. For example, consider the positive 3σ-tolerance at 30 kHz. According Table 6.6, the respective 95% confidence interval $[0.198, 0.238]$ has an extension $d = 0.238 - 0.198 = 0.040$. With respect to the total 3σ-deviation of ≈ 0.2 this results in a relative uncertainty of $0.04/0.20 = 20\%$. For other frequencies, comparable uncertainties occur, e.g. 26% at 5 kHz. Reducing S would cause even larger confidence intervals. Therefore, to get reliable results, it is not wise to reduce S significantly, say below 1000. This is the reason why Monte Carlo analysis is very time-consuming. The main advantage of quantile arithmetic, QA but especially MQA, is speed of analysis. The computing times required for tolerance analysis of the biquad circuit are shown in Ta-

ble 6.7. QA runs 18 times and MQA 26 times faster than a Monte Carlo analysis with $S = 5000$ samples. MQA requires less computing time than QA because of the accelerated sorting described in Section 6.4.3.

method	computing time
Monte Carlo	250.0 secs
QA	14.2 secs
MQA	9.5 secs

Table 6.7: Computing times[6] for tolerance analysis of the biquad circuit

Limitations of MQA

In the region of the resonant frequency $f_{res} = 16$kHz a difference between the 3σ-tolerances predicted by MQA and Monte Carlo analysis occurs (see Figure 6.23). The reason for this difference is now investigated. Basically, there are two sources of inaccuracy within MQA:

- discretization error
- non-linear correlations between QSOE expressions

With respect to the first item, a 13-point discretization has been chosen to minimize discretization error. It turns out that increasing N beyond a value of 13 doesn't improve accuracy significantly, but increases computational expense.

Concerning the second item, it is important to note that, in principle, MQA takes into account statistical correlations between different QSOE expressions. However, the statistical correlation between two expressions u and v are measured with help of a linear

[6]All computing times were measured on a sparc ultra 10 workstation in a large computer network. MAPLE runs in an interpreter mode which results in rather long computing times. Within our comparison, the Monte Carlo analysis uses also the SOE description for circuit simulation. In most cases, SOE analysis is faster than numerical simulation [138]. Applying a straight forward Monte Carlo analysis with numerical simulation would therefore be even slower.

correlation coefficient ρ_{uv} according to equation (6.30) (Page 151). Consequently, MQA (but also QA) accounts only for linear correlations. In case of non-linear correlations between QSOE expressions, MQA is only an approximation.

Since the discretization error has been minimized, it can be supposed that the inaccuracy observed above is caused by non-linear QSOE correlations. To prove this assumption, the QSOE describing the transfer function magnitude $|H|$ of the biquad has been analysed in detail. The QSOE consists of 120 expressions. It turns out that the "troublemaker" is expression $z_{111} = z_{110}^2 + z_{109}^2$. The problem is that the tolerance interval $[q_1, q_{13}]$ of the expression z_{110} has its centre approximately at zero. As z_{110}^2 is non-monotonic around $z_{110} = 0$, z_{111} depends on z_{110} in a strongly non-linear fashion. As a result, the statistical correlation between z_{110} and z_{111} is strongly non-linear. Some of the successor expression $z_{112}, z_{113}, \ldots, z_{120}$ depend on both the expressions z_{110} and z_{111}. Within MQA, the statistical pdfs of such expressions is derived by calculating the joint-pdf of z_{110} and z_{111} according to equation (6.43) (Page 161) where P_α and P_β depend on the correlation coefficient $\rho_{110,111}$ according to Table 6.4 (Page 162). Since the linear correlation coefficient $\rho_{110,111}$ is only a very rough approximation of the real nonlinear correlation between z_{110} and z_{111}, the statistics predicted by MQA for the respective successor expressions may be not accurate. Indeed, comparing MQA with Monte Carlo analysis reveals that the tolerances of all z_i, $i < 112$, are predicted correctly by MQA while the tolerances for some z_i, $i > 112$, are inaccurate.

From the results presented so far, it can be concluded that MQA is able to handle "moderately non-linear" parameter-performance dependencies correctly. This has been verified by examining the biquad transfer function at 30 kHz (see Table 6.6). Here MQA is superior to RSS which is restricted to purely linear relations. MQA becomes inaccurate when the correlation of two QSOE expressions is strongly non-linear, i.e. non-monotonic.

Tolerances of Biquad with different Q-values

Next, the biquad circuit with different Q-values is analysed. Table 6.8 shows the nominal parameter values yielding Q-values of 3 and 10 respectively. For tolerance analysis, uncor-

Q	R_1	R_2	R_3	R_4	R_5	R_6	R_7	C_6	C_8
3	$50k\Omega$	$10k\Omega$	$10k\Omega$	$10k\Omega$	$10k\Omega$	$30k\Omega$	$10k\Omega$	$1nF$	$1nF$
10	$190k\Omega$	$10k\Omega$	$10k\Omega$	$10k\Omega$	$10k\Omega$	$100k\Omega$	$10k\Omega$	$1nF$	$1nF$

Table 6.8: Filter parameters for f_{res} = 16kHz, $A = 1$ and different Q-values

related normal distributed parameters are assumed with a standard deviation of 1% with respect to the nominal values. The results of MQA and Monte Carlo analysis are shown in Figure 6.25 and 6.26 respectively. Since for $Q = 3$ and $Q = 10$ the resonant peaks are more concentrated around f_{res} = 16kHz than in the $Q = 1$ case, these plots reach only to the frequency $f = 30kHz$.

For most frequencies, the MQA results are in good accordance with Monte Carlo analysis. Only at the resonant frequency f_{res} = 16kHz, a difference concerning the positive 3σ-tolerances occurs between MQA and Monte Carlo analysis. The origin of this inaccuracy is the same as reported before. The reason why the difference is larger for $Q = 10$ than for $Q = 3$ lies in the strong non-linear influence of the parameters on the transfer function at the narrow resonant peak.

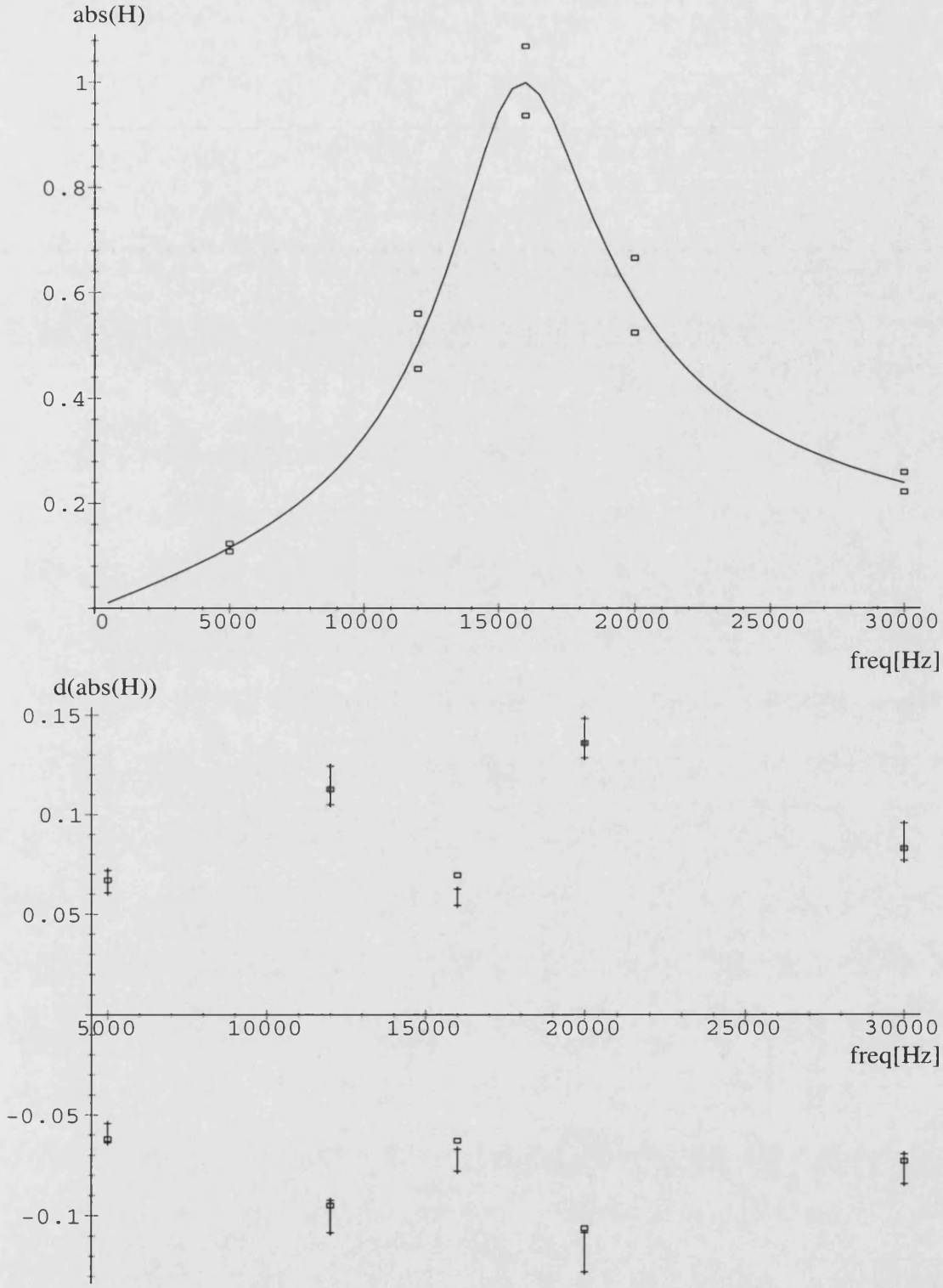


Figure 6.25: 3σ-tolerances of transfer function of biquad with $Q = 3$,
above: nominal value (solid line) and MQA 3σ-tolerances (boxes)
below: relative 3σ-tolerances, MQA (boxes),
Monte Carlo 95% confidence intervals (vertical lines)

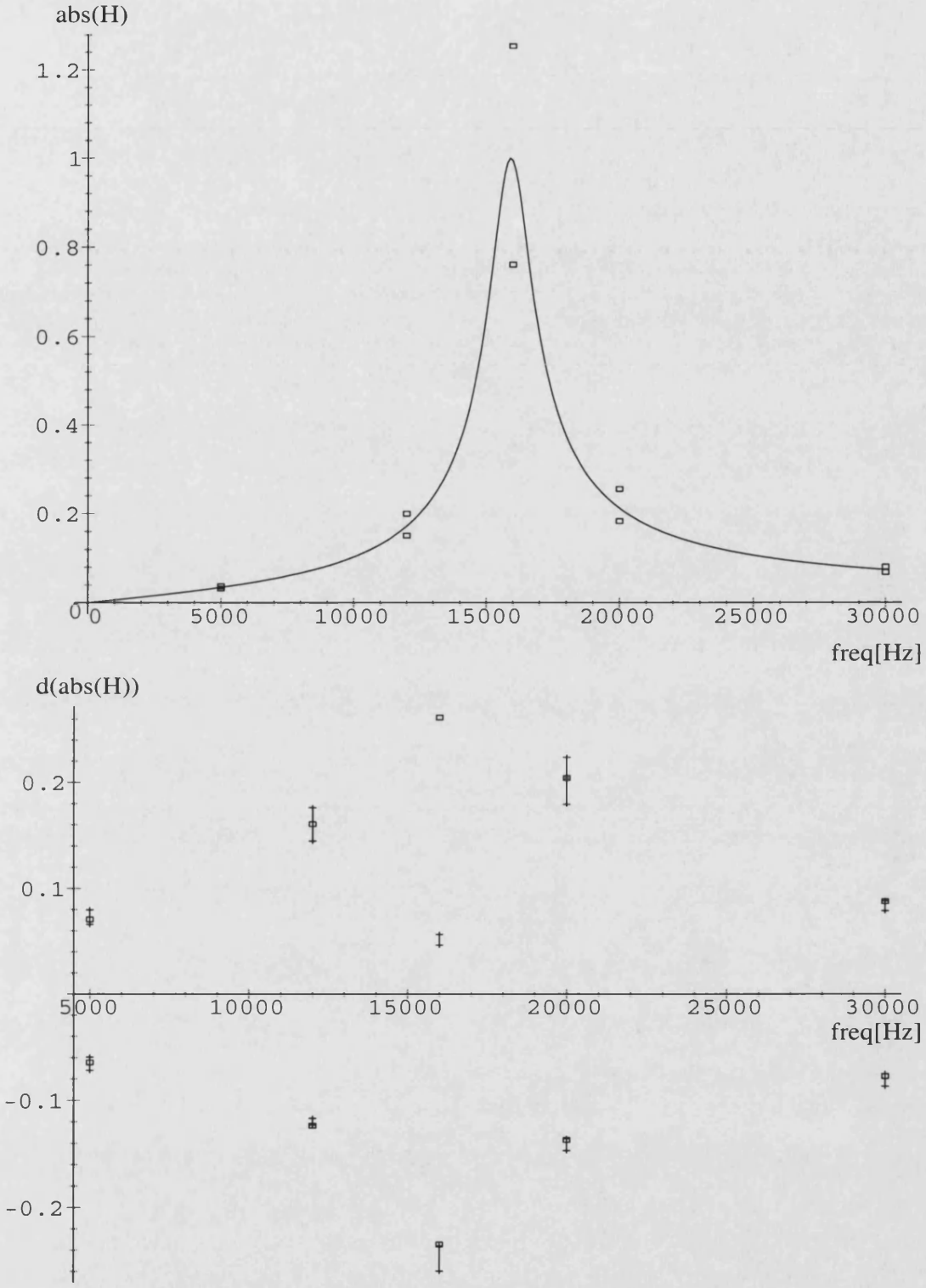


Figure 6.26: 3σ-tolerances of transfer function of biquad with $Q = 10$,
above: nominal value (solid line) and MQA 3σ-tolerances (boxes)
below: relative 3σ-tolerances, MQA (boxes),
Monte Carlo 95% confidence intervals (vertical lines)

6.6.2 Large Scale Bandpass Filter

The bandpass illustrated in Figure 6.27 is a common benchmark circuit for analysis

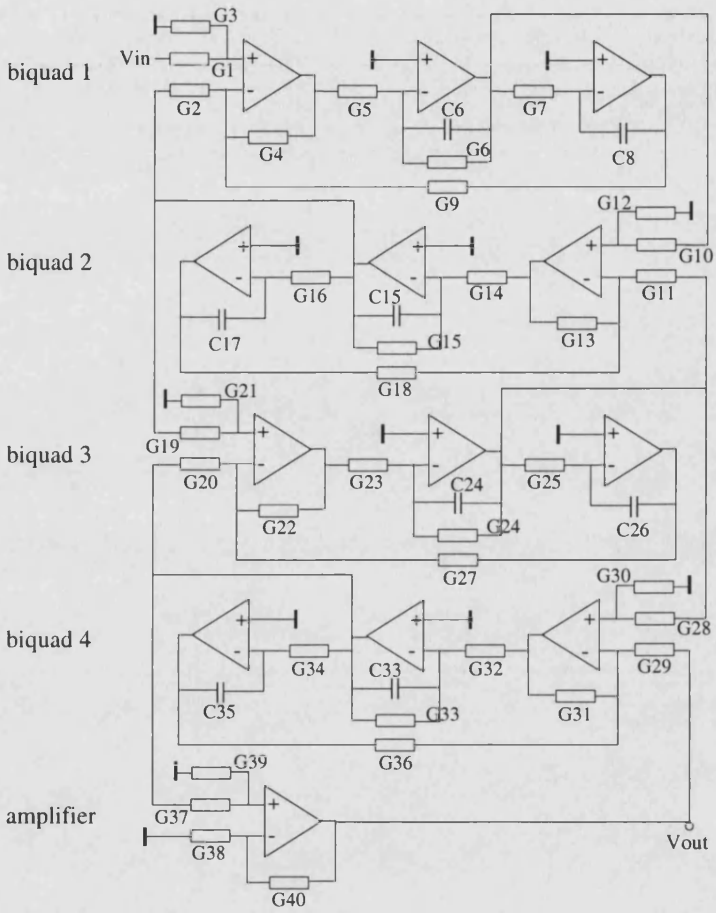


Figure 6.27: Bandpass circuit

stage	f_{res}	Q-value	A
biquad 1	8 kHz	3	3
biquad 2	8 kHz	3	3
biquad 3	16 kHz	3	2.2
biquad 4	16 kHz	3	2.2
amplifier	-	-	1

Table 6.9: Design of bandpass stages

stage	circuit parameters									
biquad 1	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_9	C_6	C_8
	$50k\Omega$	$50k\Omega$	$10k\Omega$	$10k\Omega$	$10k\Omega$	$30k\Omega$	$10k\Omega$	$10k\Omega$	$2nF$	$2nF$
biquad 2	R_{10}	R_{11}	R_{12}	R_{13}	R_{14}	R_{15}	R_{16}	R_{18}	C_{15}	C_{17}
	$50k\Omega$	$50k\Omega$	$10k\Omega$	$10k\Omega$	$10k\Omega$	$30k\Omega$	$10k\Omega$	$10k\Omega$	$2nF$	$2nF$
biquad 3	R_{19}	R_{20}	R_{21}	R_{22}	R_{23}	R_{24}	R_{25}	R_{27}	C_{24}	C_{26}
	$50k\Omega$	$50k\Omega$	$10k\Omega$	$10k\Omega$	$10k\Omega$	$22k\Omega$	$10k\Omega$	$10k\Omega$	$1nF$	$1nF$
biquad 4	R_{28}	R_{29}	R_{30}	R_{31}	R_{32}	R_{33}	R_{34}	R_{36}	C_{33}	C_{35}
	$50k\Omega$	$50k\Omega$	$10k\Omega$	$10k\Omega$	$10k\Omega$	$22k\Omega$	$10k\Omega$	$10k\Omega$	$1nF$	$1nF$
amplifier	R_{37}	R_{38}	R_{39}	R_{40}						
	$10k\Omega$	$10k\Omega$	$10k\Omega$	$10k\Omega$						

Table 6.10: Bandpass filter: nominal parameter values

procedures tackling large scale networks and has been considered for sensitivity analysis already. The bandpass circuit consists of one amplifier stage and four biquads which have the same structure as the one shown in Figure 6.15 (Page 169). These stages are designed to have the properties shown in Table 6.9. The respective parameter values of this design are listed in Table 6.10. Tolerance analysis is performed assuming the following situation:

- circuit parameters are uncorrelated, normal distributed with a standard deviation of 1% of their nominal value.
- MQA and as well QA are applied using a 13-point discretization with the cut-off points $y_{low} = \overline{y} - 3\sigma$ and $y_{high} = \overline{y} + 3\sigma$.
- Monte Carlo analysis with $S = 2500$ samples is performed for comparison purposes.

The tolerances of the transfer function magnitude predicted by QA and MQA are compared with Monte Carlo analysis in Figure 6.28. In Figure 6.29 the respective relative tolerances are plotted over frequency.

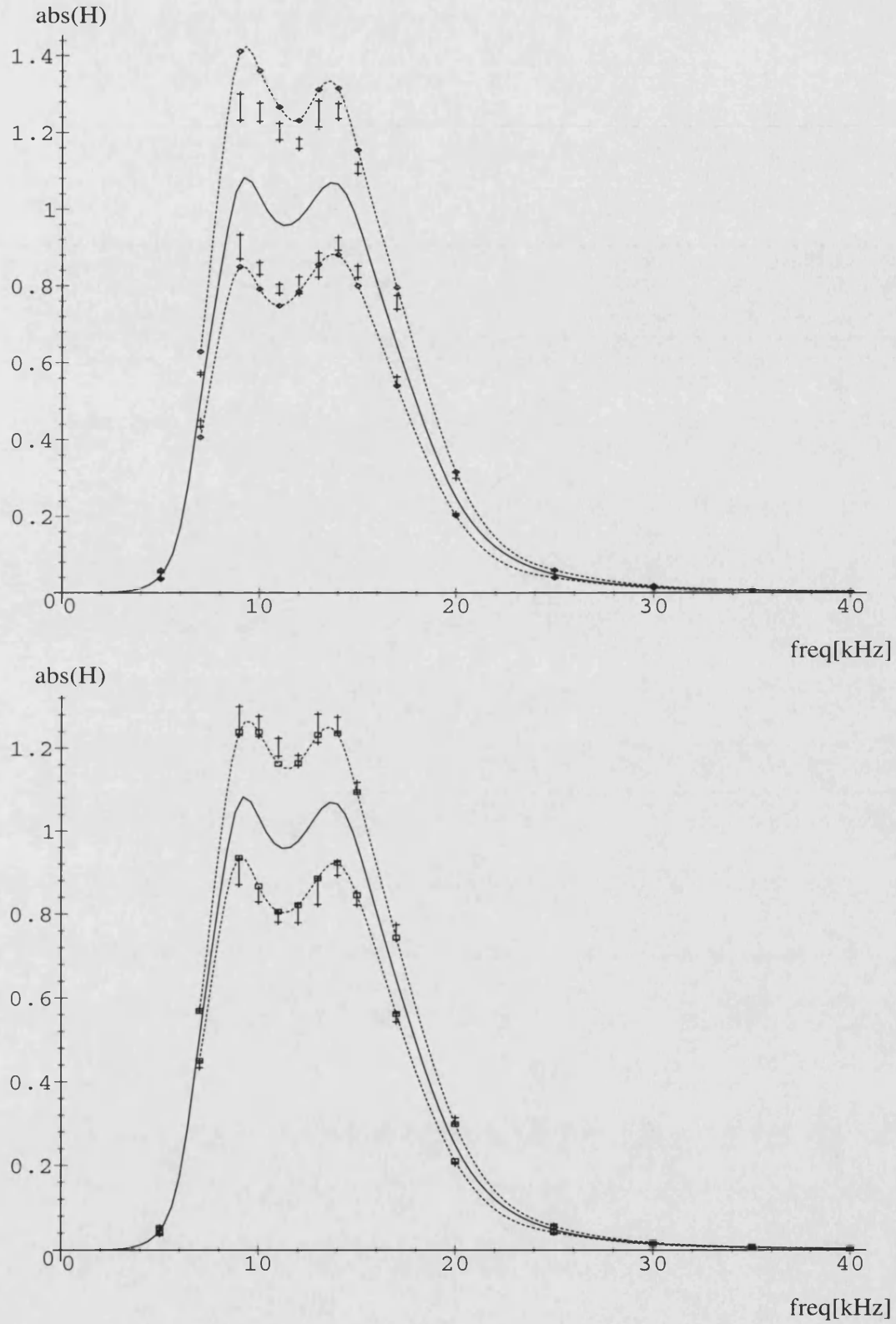


Figure 6.28: **3σ-Tolerances** of bandpass $|H|$ derived from QA (above) and MQA (below)
solid line: nominal value
dotted lines: 3σ-tolerances
vertical lines: Monte Carlo 95% confidence intervals

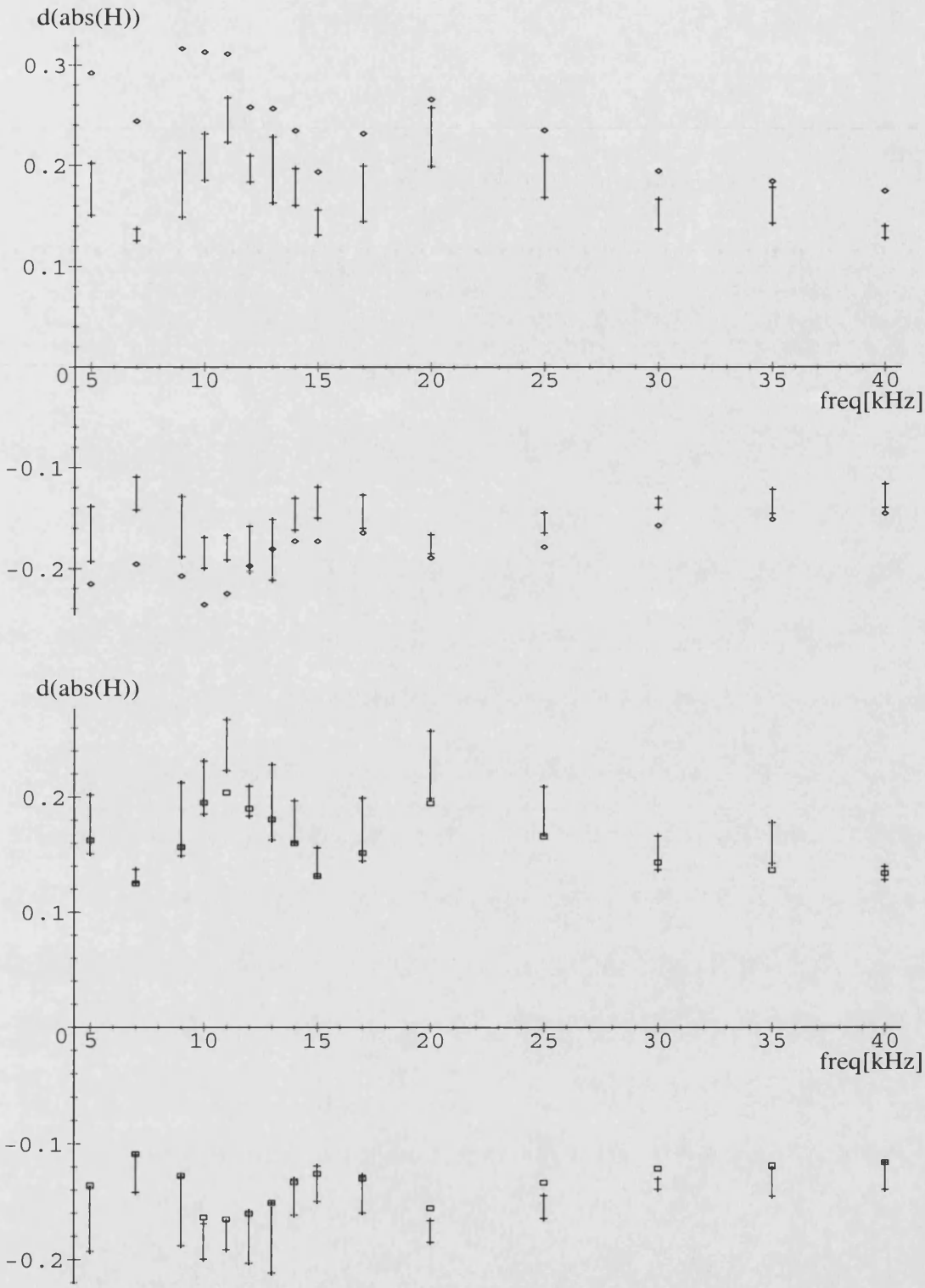


Figure 6.29: Relative **3σ-Tolerances** of $|H|$ derived from QA (above) and MQA (below)

diamonds (above):	QA relative 3σ-tolerances
boxes (below):	MQA relative 3σ-tolerances
vertical lines:	Monte Carlo 95% confidence intervals

The tolerances of MQA are mostly in good agreement with Monte Carlo analysis. Computation times of Monte Carlo analysis, QA and MQA are compared in Table 6.11. MQA

method	computing time
Monte Carlo	1980 secs
QA	163 secs
MQA	109 secs

Table 6.11: Computing times for tolerance analysis of the large scale bandpass circuit

runs 18 times faster than Monte Carlo analysis, despite the fact that the Monte Carlo samples have been reduced from $S = 5000$ to $S = 2500$. As can be seen in Figure 6.29, the reduction from $S = 5000$ (in the biquad tolerance analysis) to $S = 2500$ in the bandpass analysis results in relatively wide confidential intervals. Reducing S significantly below 2500 would cause the Monte Carlo results for the 3σ-tolerances to become very uncertain. Consequently, Monte Carlo analysis can only be accelerated when very low accuracy is acceptable. Here, MQA provides a sensible compromise with respect to accuracy and computation time.

6.7 Summary & Conclusions

A symbolic tolerance analysis procedure has been presented in this chapter. Starting from the Quantile Arithmetic approach of [183], modifications have been proposed which led to the development of a Modified Quantile Arithmetic. The achieved improvements with respect to the previous Quantile Arithmetic are

- significantly higher accuracy,
- acceleration of the analysis by up to a factor of two.

The combination of the Modified Quantile Arithmetic with the SOE method of [89] yields an automatic tolerance analysis procedure applicable to large scale analogue circuits. In the presentation of circuit examples, the Modified Quantile Arithmetic has been compared with the two currently most popular tolerance analysis methods: the RSS technique and the Monte Carlo analysis. Modified Quantile Arithmetic is approximately 20 times faster than Monte Carlo analysis and yields in most cases equivalent results, except there exist strong parameter-performance non-linearities. Compared to RSS, Modified Quantile Arithmetic is not so fast but provides better accuracy. In that consequence, Modified Quantile Arithmetic can be considered as a middle course between RSS and Monte Carlo analysis leading to a compromise concerning computing time requirements on the one hand and accuracy on the other hand. For these reasons, Modified Quantile Arithmetic holds promise for the application in the design process of today's large scale analogue circuits.

Chapter 7

Conclusions & Future Work

This chapter summarizes the work described in this thesis and discusses the results. Ideas for further improvements of the presented techniques are given and future directions for research in analogue design support and tolerance analysis suggested.

During the last decade, the development of the analogue part of ICs has become a bottleneck due to the lack of CAD tools. Especially the absence of an efficient tolerance analysis support causes low yield, increased product cost, decreased quality and longer time to market. The reasons for the impracticability of the currently available tolerance analysis techniques were traced back to the growing complexity of analogue circuits which results in unrealistically large computing times.

To improve this situation, two directions of work were presented in this thesis:

- **characterisation support** by providing a technique for structured design-for-testability (DFT) and *parametric fault diagnosis.* Such a technique alleviates the localization of circuit blocks responsible for performance deviations.

- **design support** by providing effective *tolerance analysis* techniques applicable to today's large scale analogue circuits.

7.1 Parametric Fault Diagnosis

Parametric fault diagnosis is strongly related to test issues. For this reason, an overview on testing digital, analogue and mixed-signal circuits was given in Chapter 2. The techniques for testing digital circuits are quite mature. This has become possible through the availability of easy to analyse fault models which can be used for automatic test pattern generation and which allow for a structural, defect oriented test approach.

Then, the difficulties related with analogue test and the reasons why mostly a functional test approach is chosen in the analogue domain were outlined. The analogue fault modelling techniques were classified into hard fault-, soft fault- and hierarchical fault modelling. The methods for analogue test generation were categorized into simulation before test (SBT) and simulation after test (SAT). The first category is primarily usable for hard fault testing and achieves fault detection with the help of a fault dictionary. The methods of the second category are suited to soft fault testing and aim at both fault detection and localization using either parameter identification, fault estimation or fault verification techniques. Finally, the different DFT suggestions to alleviate the testing of analogue and mixed-signal ICs were described.

Based on this overview, it has been decided to use a topological SAT method with fault verification as the basis for the development of a parametric fault diagnosis algorithm (FDA). The reasons for this choice were the following:

1. a SAT technique is more appropriate than a SBT method for diagnosing parametric deviations.

2. a parameter identification technique cannot be applied to integrated circuits due to the high number of test nodes required.

3. topological fault verification methods are computationally more efficient than estima-

tion methods and can also be applied to diagnosing multiple simultaneous parametric deviation faults.

From the available topological SAT techniques with fault verification Wey's fault diagnosis approach was chosen because

1. a testability condition is formulated that is merely based on the circuit's topology. This allows for the development of a structured DFT method which can be applied in an early design stage before the chip has been laid out.

2. given a maximal number of simultaneous parametric faults (input by the user), a minimal set of test points is selected which guarantees diagnosibility at minimal impact on circuit behaviour.

3. no additional on-chip circuitry is required.

4. all steps in the fault diagnosis procedure are algorithmically formulated. Therefore an automatic tool support can be programmed.

In Chapter 3, Wey's fault diagnosis procedure was described in detail. The underlying circuit description, the component connection model (CCM), and the self test (ST) algorithm were introduced. Then the limitations of this approach concerning the applicability to large scale ICs and switched capacitor (SC) circuits were discussed. To overcome the circuit size limitations, a hierarchical modelling strategy was proposed. Emphasis was drawn to the inclusion of hierarchical models into the CCM and the implications for the developed hierarchical fault diagnosis algorithm (HFDA). Furthermore, the CCM was extended to a voltage/charge based discrete-time description such that the HFDA becomes applicable to SC circuits. Finally, the ANSI C implementation of the HFDA was applied to a practical circuit example to diagnose single parametric deviation faults.

It is important to note that the development of the HFDA was organized as a collaboration between the University of Bath and the Robert Bosch company. The contributions of the author concentrated on the modelling aspects. For a description of HFDA properties which are not directly related to modelling and for a more comprehensive collections of experimental results, the reader is referred to [79].

The following conclusions can be drawn from the practical application of the HFDA:

1. under ideal conditions (high measurement precision, no tolerances of fault-free components) the HFDA provides correct fault diagnosis in most cases.

2. single and multiple faults can be diagnosed.

3. the benefits of the hierarchical approach are:

   - significantly reduced computing time and storage requirements making feasible the parametric diagnosis of large scale circuits. In [79] a hierarchical representation of a passive passive resistor circuit reduced the number of circuit edges from 18 to 14 and the number of test cycles required for the diagnosis from 248 to 5. This shows that large computing time reductions are achievable. The major resason for this improvement is the strong reduction of the number of required test cycles rather than accelerated matrix operations.

   - since the hierarchical approach reduces the number of variables in the circuit description the number of test points is also reduced which enhances the application of the HFDA to integrated circuits.

4. the drawbacks of the hierarchical approach are:

   - the diagnosis resolution is reduced because the HFDA can provide only go/no-go testing on a hierarchical circuit block.

- the convergence of the ST algorithm to reliable test results is sometimes deteriorated by a too high hierarchical description level. Firstly, the number of tester/testee-partitions is diminished by enclosing several lower level components into one hierarchical block. Secondly, the hierarchical approach causes the connection matrices to be more sparse which reduces the number of tester/testee-partitions for which the Pseudo Circuit equations exist.

- the generation of the hierarchical models inevitably adds some overhead in the application of the HFDA. This is especially true when a structural testing approach is chosen in which the tolerances of a hierarchical block need to be derived from the statistics of its components by tolerance analysis. However, as circuit complexity increases, the designer is also forced to a hierarchical design strategy in which behavioural models need to be developed and specified. In such a situation, the models developed by the designer can be used directly within the HFDA for the characterisation test. This is similar to the approaches adopted in [217, 218].

5. under real conditions, the HFDA in the current version is not applicable. Measurement inaccuracies as well as tolerances in fault-free components (which can not be avoided with analogue devices) cause the decision algorithm to work improperly.

Based on the experiences with the application of the HFDA the following recommendations for future enhancement can be made:

- Obviously, the HFDA becomes most effective when a sensible compromise in the hierarchical description level is accomplished. Investigations concerning the optimal hierarchical level would help. Important criteria in this respect are the diagnosibility on the one hand and computation time on the other hand.

- the main requirement for the practical applicability of the HFDA is the reliability of the diagnosis results under real conditions. Therefore, the decision algorithm of the ST approach of [57, 79, 86] has to be modified in a way that tolerances of fault-free components don't affect its performance (avoid tolerance masking) and that the HFDA becomes insensitive to measurement inaccuracies. Interesting is that the most critical point of the exact decision algorithm within the ST approach is the following: 'more than one testees are tested failed $\Rightarrow$ all testees are good'. Investigations why this statement causes most error and whether it is possible to remove this statement from the HFDA may be helpful.

- one observation during the evaluation of the HFDA was that many test cycles are not testable. This is especially true when a high hierarchical description level has been chosen. A consequence is that many additional test cycles need to be processed before a reliable diagnosis result is achieved which causes an overhead in test time. Alternatively, it may happen that the diagnosis capability is deteriorated and no reliable test result is achieved at all. Examples for this situation occured with the filter circuit (second and third line from below in Table 3.6). These observations are an indication that an improvement of the test point selection is required. Additionally, the circuit tree generation can be optimised with respect to an enhancement of the number of testable test cycles. Methods in those directions have been currently presented in [60].

- the application of the HFDA to hierarchical networks and SC circuits showed that the selection of the optimal tree may require a lot of time. This is caused by the fact that the HOTG (see page 48) firstly chooses a tree to achieve sparse system matrices. It is checked only afterwards whether the tree is in accordance with the hierarchical model structure. This procedure can be accelerated significantly if it is

possible to fulfill the hierarchical restrictions directly within the HOTG.

- at the moment, the next tester/testee-partition is chosen without taking into consideration the previous test results, except in the case that enough components are tested good to achieve reliable test results in the next partition. The convergence of the HFDA to correct diagnosis results can be improved significantly by appropriately storing the results of previous test cycles. This information helps to evaluate the results of the current test cycle and to choose the tester/testee-partition for the next test cycle in a more deterministic manner.

- the calculation of the next tester/testee-partition is currently also a computation time bottleneck. This is caused by the applied recursive algorithm which works as follows: when deriving the 100th partition, for example, the algorithm starts from the first one and recursively proceeds through all predecessors until arriving at the 100th partition. A faster algorithm should have the capability to derive the 100th partition directly from the 99th partition.

- in the case that a terminal of a hierarchical component is connected to the reference node, some dynamic degrees of freedom can be removed from the circuit description. The respective component matrix can be collapsed by deleting rows and columns which refer to the respective terminal and by removing the respective component edges from the circuit graph. This accelerates the HFDA performance. At the moment the matrix reductions must be done manually and an automatic procedure would make life much easier.

- one major natural continuation to the fault diagnosis work is to extend the HFDA to deal with non-linear analogue as well as mixed-signal circuits. The approach investigated in [85] may be suitable in this proposed research direction. The inclusion

of non-linear effects like opamp offset can even be achieved with slight additions to the actual HFDA. Offset can be modelled by just changing the component equation (3.1) from $b = Za$ to $b = Za + off$ where off represents the offset effects.

7.2 Symbolic Tolerance Analysis

The main purpose of the thesis was to develop and investigate techniques for efficient tolerance analysis which are applicable to large scale analogue circuits. An important focus of the work was to improve the accuracy of the developed methods and to reduce their computational expense.

Symbolic analysis has been chosen as a basis for the investigations in this thesis. The advantage of symbolic analysis is that only one simulation run is needed in which a symbolic expression for the circuit behaviour is generated. During tolerance analysis, successive evaluations of the compiled symbolic expression replace then the necessity for any extra time-consuming numerical iterations through the simulator.

In Chapter 4 the current state of the art in symbolic analysis of analogue circuits was reviewed. Symbolic analysis is useful for getting insight into circuit behaviour, for circuit optimization procedures and for iterative tasks which require repetitive formula evaluation. The capabilities and limitations of symbolic analysis were outlined. Currently, symbolic analysis is most appropriate for the frequency domain simulation of small signal behaviour. Then the various symbolic methods are classified in respect of their algorithmic aspects. A primary purpose of this chapter was to find out which technique is most efficient for tolerance analysis of large scale networks. Herein, important criteria were

- the number of terms in the symbolic network function should be as low as possible. This reduces numerical evaluation time during tolerance analysis.
- the possibility to analyse the influence of all circuit elements with arbitrary value

range of the element parameters.

- good accuracy over the whole frequency range.

A comparison of the available methods with respect to these criteria showed that the symbolic hierarchical decomposition is the best solution. SCAPP has been selected as a well-known hierarchical technique with high performance in the case of large scale networks. Finally, the reader was introduced to the three major steps of SCAPP, binary circuit partitioning, subcircuit analysis and upward hierarchical analysis. The outcome of SCAPP is a symbolic network function in the form of a "sequence of expressions" (SOE) in which the number of terms grows typically linearly with circuit size (instead of the traditional exponential growth). This makes the SOE approach attractive for the analysis of large scale circuits.

7.2.1 Symbolic Sensitivity Analysis

An important technique which helps in tolerance investigations is sensitivity analysis. To make this technique efficient for today's large scale analogue circuits, Chapter 5 was dedicated to hierarchical symbolic sensitivity analysis. At first, the role of sensitivity methods in tolerance analysis, tolerance design and circuit optimization was outlined. The different numerical and symbolic approaches were compared concerning their computational expense. Symbolic sensitivity analysis has its advantages when the behaviour at many frequency and parameter points needs to be investigated. Since the effectiveness of symbolic sensitivity analysis is directly related to the number of terms in the network function, the techniques developed in this thesis were based on the SOE approach.

Chapter 5 continued by discussing the previous symbolic sensitivity procedures using the SOE. The Directed Acyclic Graph (DAG) was introduced which represents dependencies between the expressions of the SOE and helps in implementing automatic symbolic

sensitivity analysis. The drawback of the SOE sensitivity techniques previously presented in literature is that they still require a large number of arithmetic operations when the sensitivities with respect to many or all parameters need to be determined. Efficient multi-parameter sensitivity analysis, however, is essential for tolerance analysis and tolerance design.

Two novel SOE methods for fast multi-parameter sensitivity were then presented:

- the *balanced* symbolic sensitivity analysis and
- the *parallel* symbolic sensitivity analysis.

The first technique was based on a structural similarity of the DAG and the binary partition tree (BPT) which models the circuit partitioning in the hierarchical symbolic analysis of SCAPP. An easy to handle heuristic for minimizing the computational expense of multi-parameter sensitivity analysis was introduced by proposing the use of a maximally balanced BPT to derive the SOE.

Additionally it was shown that it is possible to calculate sensitivity by traversing the SOE in the opposite direction than the previous approaches. By this means, the sensitivities with respect to all circuit parameters can be derived in parallel.

An estimation for the computational complexity of the novel techniques was given. The MAPLE implementations of the sensitivity procedures were then applied to large scale circuits to investigate the efficiency of the new methods in comparison with previous approaches. The experimental results can be summarized as follows:

- **balanced symbolic sensitivity analysis:**
  - speed-up compared to previous SOE sensitivity procedure: $1 \ldots 20$. The speed-up increases with circuit size.
  - computational expense in dependence on n (number of circuit nodes): $O(n \log_2 n)$.

  - applicable to both differential and large change sensitivity analysis and to parametric fault simulation.
  - algorithmic requirements: technique is based on a symbolic SOE which has been generated using hierarchical partitioning.

- **parallel symbolic sensitivity analysis:**
  - speed-up compared to previous SOE sensitivity procedure: $1 \ldots 70$. The speed-up increases with circuit size.
  - computational expense in dependence on n: $O(n)$.
  - applicable to differential sensitivity analysis.
  - no algorithmic requirement, the technique can be applied to any arbitrary SOE.
  - the parallel procedure can be faster than the numerical adjoint approach. The effectivity of the parallel approach strongly depends on the structure of the nominal SOE (see conclusions of chapter 5).

These results show the efficiency of the new techniques. To the best knowledge of the author, both developed techniques are faster than any symbolic multi-parameter sensitivity method presented so far in the literature. Since the speed-up grows with circuit complexity, both methods hold promise for the application to today's large scale analogue circuits. In combination with the symbolic simulator SCAPP, an automatic sensitivity analysis tool can be established.

Future research should investigate

- numerical stability: as a result of the compact nesting in the SOE inaccuracies may occur during numerical evaluation by the term cancellation phenomenon (terms with same symbols but opposite sign, see Chapter 4). There exist numerous SOE formats which have different nesting structures and consequently show different behaviour

concerning term cancellation. An examination of the different SOE formats concerning their numerical accuracy would therefore be useful[1].

- further acceleration: this can achieved by examining different SOE generation procedures. In the conclusions of chapter 5 it has been shown that the structure of the nominal SOE is essential for the performance of the parallel procedure. Since within the last years significant improvements in symbolic SOE analysis have been achieved, e.g. [138, 152], there is room for further improvement of SOE sensitivity analysis. A nominal SOE which has a minimal arithmetic operation count is a good candidate for additional reduction of the expense of the parallel SOE procedure [219]. However, a minimal nominal SOE is not necessarily the best solution for the parallel procedure, since different arithmetic operations in the nominal SOE cause different expense within the respective parallel sensitivity SOE. For example, divisions result in a high expense for edge weight calculation. Consequently, the number of divisions in the nominal SOE may be more significant than the number of multiplications. Altogether, the goal is to find a SOE which optimally fits the parallel sensitivity approach.

- extensions to non-linear circuits: symbolic analysis is primarily for simulating linear circuit behaviour in the frequency domain. In recent years however, symbolic techniques have successfully been applied to speed-up highly iterative tasks in non-linear circuit simulation, e.g. fault diagnosis [128, 132]. Even the first approaches to direct simulation of non-linear behaviour have been presented [131, 133]. These developments may allow the extension of the presented sensitivity techniques to non-linear analogue circuits in the future.

[1]No numerical inaccuracies were observed with the practical applications presented in this thesis.

7.2.2 Modified Quantile Arithmetic

Sensitivity analysis is useful to circuit optimization and to get a first insight into tolerance behaviour. To examine the effects of parameter variations in a more accurate manner, additional methods are required. In Chapter 6, the currently available tolerance analysis methods were reviewed and classified into worst-case and non-worst-case analysis, sampling and non-sampling approaches. It was outlined that these techniques are either not accurate or very time consuming. Popular examples are the root-sum-square (RSS) and Monte Carlo method. The first one is fast but inaccurate. The second one is accurate under the condition that a lot of circuit simulations are performed which causes high computing time requirements, especially for larger circuits. As a result, it is mostly impossible to apply the Monte Carlo method during the design process.

A tolerance analysis method was then proposed which provides a compromise with respect to sufficient accuracy on the one hand and low computational expense on the other hand. The developed method was based on Quantile Arithmetic (QA) which computes circuit tolerances using discretization of random variables. The main advantage of QA is that it runs approximately one order of magnitude faster than the Monte Carlo analysis. The various steps within QA were described in detail. By performing several experiments the following limitations of QA were found:

- the method is only accurate for yield prediction, if yield is in the region of 95%. In the case of low yield ($< 90\%$) or very high yield ($> 98\%$), QA is unacceptably inaccurate.

- the last step of QA, the reduction to a N-point variable, requires long computing time due to the expensive sorting process.

Both the limitations were then overcome by developing a novel Modified Quantile Arith-

metic (MQA). Improved precision in all regions of predicted yield was achieved by defining realistic joint-distributions for random variables. Increased analysis speed was obtained by taking advantage of monotonic properties within the sorting process. Finally, MQA was combined with the SOE technique to provide an efficient automatic technique for symbolic tolerance analysis of large scale analogue circuits. MQA was implemented in MAPLE and applied to practical circuit examples. Comparisons with the performances of other tolerance analysis methods were accomplished. The experimental results can be summarized as follows:

- MQA runs approximately 2 times faster than QA and provides results of significantly higher accuracy in all regions of predicted yield.
- MQA is not as fast as RSS, but more accurate.
- MQA runs approximately 20 times faster than the Monte Carlo analysis and provides in most cases similar results.
- MQA can handle moderate parameter-performance non-linearities. However, it shows inaccuracies in the case of strong non-linearities, i.e. non-monotonic correlations. The underlying reason is that the joint-distribution defined within MQA (but also QA) is only valid for monotonic correlations between random variables.

This summary shows that the development of MQA yielded a tolerance analysis with improved speed and accuracy compared to QA and a significant acceleration compared to Monte Carlo analysis.

To evaluate the practical applicability of MQA in the design process it is necessary to recall the following criteria for tolerance analysis methods:

1. precision

2. measure for the precision and the involved error (e.g. statistical confidence of the result)

3. computation time

4. computing time in dependence of number of parameters

5. obtained information for tolerance optimization

With repect to the first and third criteria, the experimental results indicate that MQA achieves a sensible compromise concerning the two contrary requirements of high accuracy and low computational cost, especially compared to the Monte Carlo analysis which mostly needs unrealistic huge computing times. Additionally, the computing time of MQA is solely dependent on the number of SOE expressions. As a result, the computational expense of MQA doesn't grow exponentially with the number of circuit parameters which was the killing factor for the application of previous deterministic sampling techniques like regionalization.

One drawback of the actual version of MQA, similarly as for other deterministic methods (e.g. RSS), is the missing measure of the statistical confidence of the obtained tolerance prediction. The main reason for the success of the Monte Carlo approach is that it directly provides this information. Furthermore, at the current status, MQA is merely a tolerance analysis approach and not a tolerance optimization method: techniques to optimize the tolerance behaviour of a circuit based on the MQA results need to be developed in future.

In the literature, techniques have been proposed to accelerate the Monte Carlo approach. One of the most significant approaches into this direction are hierarchical Monte Carlo methods. In [41], for example, the circuit is hierarchically decomposed and statistical behavioural models for the circuit blocks are generated. The tolerances of the circuit behaviour are then derived using behavioural simulation. The main advantage of such ap-

proaches compared to MQA are their general applicability to linear and non-linear circuits. However, the price for the general applicability is the overhead related to the generation and characterization (both nominal and statistical behaviour) of the hierarchical models which requires a lot of manual effort. MQA provides here the advantage of a complete automatic procedure by applying the steps shown in Figure 6.14.

Based on this evaluation, to improve the applicability of MQA in the design process the following ideas may be helpful:

- the major source of inaccuracies within MQA are non-linear correlations between two SOE expressions. This has been explained with the active biquad circuit example in Section 6.6.1. The "troublemaking" SOE expression in this example was $z_{111} = z_{110}^2 + z_{109}^2$ where the tolerance interval $[q_1, q_{13}]$ of the expression z_{110} has its centre approximately at zero. As a result of the squaring of z_{110}, z_{111} depends on z_{110} in a strongly non-linear, i.e. in a non-monotonic, fashion. Since the MQA procedure actually considers only linear correlations between SOE expressions, this non-monotonic correlation between z_{111} and z_{110} causes inaccuracies when both variables are statistically combined in MQA to calculate the pdf of a successor expression, i.e. $z_x = z_{110} \circ z_{111}$. In several examples it has been observed that non-monotonic dependencies are the main contributions to inaccuracies of MQA. For this reason, a heuristic indication for the precision of the tolerances predicted by MQA is to monitor whether quantile intervals of SOE expressions include zero and whether such variables are used in multiplications, divisions or squarings. If this is the case, there is a risk for precision and a warning signal can be given to the user.

- there exist a lot of different SOEs with different nesting structures. These different SOE structures have different numbers of each specific arithmetic operation. If it is possible to choose a SOE such that the number of multiplications, divisions

and squarings is diminished (perhaps at the cost of additional additions and subtractions), the risk that strongly non-linear and non-monotonic correlations occur during MQA analysis is reduced. This may improve the accuracy of MQA.

- the limitation of MQA to linear correlations originates from the usage of a linear correlation coefficient to describe statistical dependencies. This linear correlation coefficient refers to a linear approximation of the dependencies between two statistical variables (see Appendix A.1). One possible improvement would be to use a piece-wise linear approximation of dependencies between statistical variables within their respective tolerance region $[q_1, q_{13}]$. Then, it needs to be investigated how the procedure of Appendix A.1 can be modified such that correlation coefficients of the different variables can be calculated when MQA proceeds successively from one SOE expression to the next.

- within the actual MQA version the complex variables are described by their real and imaginary part. An alternative is to use polar coordinates. In the case of a division for example, the actual version of MQA requires 9 intermediate real-valued variables to calculate the pdfs of the real and imaginary part of the result of the division. For each of the 9 intermediate variables a pdf needs to be determined by applying MQA from step 2 to step 4 according to Figure 6.2. Using polar coordinates, only two pdfs need to be derived (for the expressions $r = r_1/r_2$ and $\phi = \phi_1 - \phi_2$). This certainly increases speed, but in addition may increase accuracy of the final result.

- actually, whenever the tolerance interval of a variable contains the zero and the variable is used as divisor, the calculations are aborted. A possible extension is to calculate the pdf of the division in spite of this situation. To achieve as much accuracy as possible the number of discretization points of the respective variable in the area around zero can be increased.

- having applied MQA to several circuit examples, it has been observed that the variances predicted by MQA show a tendency to be smaller than the respective ones predicted by Monte Carlo analysis. Though the difference is small (typically around 5%, e.g. Figure 6.29), the inspection of several MQA results has proven this tendency in the average. The reason behind this situation lies in the reduction of a N^2-point variable to the respective N-point variable (MQA step 4): Let (z_{ij}, p_{ij}), $i, j = 1 \ldots N$ correspond to the N^2 unreduced probability peaks and (Z_i, P_i), $i = 1 \ldots N$ are the respective reduced peaks where

$$Z_i = \frac{1}{P_i} \sum_{j=1}^{N} p_{ij} z_{ij}, \tag{7.1}$$

$$P_i = \sum_{j=1}^{N} p_{ij}. \tag{7.2}$$

For simplicity it is assumed that always N peaks of the unreduced variable are combined to one peak of the reduced variable. The variance of the unreduced probability variable is given by

$$\sigma^2 = \sum_{ij} (z_{ij} - \mu)^2 p_{ij} = \sum_{ij} z_{ij}^2 p_{ij} - \mu^2, \tag{7.3}$$

where μ is the mean value of the statistical variable. The variance of the reduced variable is

$$\sigma_{red}^2 = \sum_i (Z_i - \mu)^2 P_i = \ldots = \left(\sum_i \frac{(\sum_j z_{ij} p_{ij})^2}{P_i} \right) - \mu^2, \tag{7.4}$$

where equations (7.1) and (7.2) have been used. The difference between the variances of the statistical variable before and after the reduction is given as follows

$$\Delta = \sigma^2 - \sigma_{red}^2 = \sum_{ij} z_{ij}^2 p_{ij} - \left(\sum_i \frac{(\sum_j z_{ij} p_{ij})^2}{P_i} \right) \tag{7.5}$$

$$= \ldots = \sum_i \Delta_i P_i, \tag{7.6}$$

where

$$\Delta_i = \frac{\sum_j (z_{ij} - Z_i)^2 p_{ij}}{P_i}. \tag{7.7}$$

Since $\Delta_i \geq 0 \ \forall i$, it can be concluded that $\Delta \geq 0$, or in other words, that $\sigma^2 \geq \sigma_{red}^2$. This property is inherent in the reduction process because the combination of neighboured probability peaks to one representative peak always reduces variance. The straight forward way to minimize this effect is to increase the number of discretization points N. However, this can only be done at the cost of significantly increased computation time. A better idea is to monitor the loss in variance during the reduction process. After the reduction, the distances $Z_i - \mu \ \forall i = 1 \ldots N$, can be linearly scaled by the appropriate scaling factor $\frac{\sigma}{\sigma_{red}}$ to compensate for the loss of variance.

- Despite the development of an accelerated algorithm for sorting, this step is still the bottleneck. An idea for improvement is to do away with sorting completely and calculate the distribution of $z = u \circ v$ based on finding equi-value lines in the (u, v)-plane. This process can be supported by effective integral transforms (e.g. FFT for $z = u + v$ or $z = u - v$).

It is planned in the near future to combine the developed sensitivity and tolerance analysis procedures with the SCAPP program to provide tool support for the analogue designers at Bosch.

List of Publications

The research work of this thesis has resulted in the following publications:

1. C.-K. Ho, P. R. Shepherd, F. Eberhardt and W. Tenten, 'Hierarchical Approach to Analogue Fault Diagnosis', Proceedings 3rd IEEE International Mixed-Signal Testing Workshop, Seattle, pp. 25-30, 1997.

2. C.-K. Ho, P. R. Shepherd, W. Tenten and F. Eberhardt, 'Improvements to Circuit Diagnosis through Hierarchical Modelling', IEE Colloquium for Testing Mixed-Signal Circuits & Systems, 23rd October 1997, Savoy Place, London.

3. F. Eberhardt, W. Tenten, C.-K. Ho and P. Shepherd, 'A Structural Approach to Hierarchical Tolerance Modelling of Analogue CMOS Integrated Circuits', Proceedings 4th IEEE International Mixed-Signal Testing Workshop, The Hague, p. 288-297, 1998.

4. F. Eberhardt, W. Tenten and P. R. Shepherd, 'Improvements in Hierarchical Symbolic Tolerance & Sensitivity Analysis', Electronics Letters, Vol. 35, No. 4, pp. 261-263, Feb. 1999.

5. F. Eberhardt, W. Tenten and P. R. Shepherd, 'Symbolic Parametric Fault Simulation & Diagnosis of Large Scale Linear Analogue Circuits', Proceedings 5th IEEE International Mixed-Signal Testing Workshop, Vancouver, Canada, pp. 221-228, 1999.

6. F. Eberhardt, W. Tenten and P. R. Shepherd, 'Parallel Symbolic Sensitivity Analysis of Large-Scale Analogue Circuits', submitted to IEEE Transactions on Circuits and Systems.

7. F. Eberhardt, W. Tenten and P. R. Shepherd, 'Symbolic Tolerance Analysis of Large Scale Analogue Circuits', to be submitted to IEE Proceedings Circuits Devices Systems.

Appendix A

Quantile Arithmetic

A.1 Correlations between QSOE Expressions

Let $x_1, \ldots, x_{N_X}$ be statistical independent variables (not necessarily the circuit parameters) with standard deviations $\sigma_1, \ldots, \sigma_{N_X}$. Let the QSOE expressions be functions of these variables

$$z_\nu = z_\nu(x_1, \ldots, x_{N_X}), \qquad \nu = 1 \ldots m. \tag{A.1}$$

Furthermore, the covariance of two discretized variables u_i and v_j with joint-pdf P_{ij} is defined by

$$\mathrm{cov}(u, v) = \sum_{i,j} P_{ij}(u_i - \overline{u})(v_j - \overline{v}). \tag{A.2}$$

The covariances of the expressions z_ν and the variables x_i are stored in a covariance matrix which has the following structure

$$\begin{array}{c} \\ x_1 \\ x_2 \\ \vdots \\ x_{N_X} \\ z_1 \\ \vdots \\ z_m \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & \cdots & x_{N_X} \end{array} \\ \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & & 0 \\ \vdots & & \ddots & 0 \\ 0 & 0 & & \sigma_N^2 \\ \mathrm{cov}(z_1, x_1) & \mathrm{cov}(z_1, x_2) & \cdots & \mathrm{cov}(z_1, x_{N_X}) \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{cov}(z_m, x_1) & \mathrm{cov}(z_m, x_2) & \cdots & \mathrm{cov}(z_m, x_{N_X}) \end{pmatrix} \end{array} \tag{A.3}$$

During the calculation of $z_\lambda = z_\nu \circ z_\mu$ in QA, the row corresponding to the covariances with respect to z_λ is determined using

$$\mathrm{cov}(z_\lambda, x_i) = \left.\frac{\partial z_\lambda}{\partial z_\nu}\right|_{\overline{z_\nu},\overline{z_\mu}} \mathrm{cov}(z_\nu, x_i) \quad + \quad \left.\frac{\partial z_\lambda}{\partial z_\mu}\right|_{\overline{z_\nu},\overline{z_\mu}} \mathrm{cov}(z_\mu, x_i), \quad i = 1 \ldots N_X. \tag{A.4}$$

where the partial derivatives are taken at the mean values $\overline{z_\nu}$ and $\overline{z_\mu}$. In the case of non-linear dependencies between the expressions, equation (A.4) is only an approximation. The correlation coefficient $\rho_{\nu\mu}$ of the QSOE expressions z_ν and z_μ is calculated using previous entries of the covariance matrix:

$$\begin{aligned}
\rho_{\nu\mu} &= \frac{\mathrm{cov}(z_\nu, z_\mu)}{\sigma_{z_\nu}\,\sigma_{z_\mu}}, \\
\mathrm{cov}(z_\nu, z_\mu) &= \sum_{i=1}^{N_X} \frac{\mathrm{cov}(z_\nu, x_i)}{\sigma_{x_i}} \cdot \frac{\mathrm{cov}(z_\mu, x_i)}{\sigma_{x_i}}, \\
\sigma_{z_\nu}^2 &= \mathrm{cov}(z_\nu, z_\nu), \\
\sigma_{z_\mu}^2 &= \mathrm{cov}(z_\mu, z_\mu).
\end{aligned} \tag{A.5}$$

Further details and proofs of the relations (A.4) and (A.5) are given in [183].

A.2 Combination of Neighboured Probability Peaks

In this section, the sorted sequence of N^2 probability peaks (z'_l, P'_l) is reduced to a according sequence of N probability peaks (Z_i, P_i), $i = 1 \ldots N$. This is achieved by combining neighboured peaks of the original (z'_l, P'_l)-sequence as follows. Firstly, the respective pcf $P_{cum}(Z)$ is generated by adding up the probability peaks P'_l:

$$P_{cum}(Z) = \sum_{l=1}^{L} P'_l, \qquad \text{where} \quad z'_L \leq Z < z'_{L+1}. \tag{A.6}$$

$P_{cum}(Z)$ has the shape of a step function as illustrated in Figure A.1. The reduced pdf P_i, $i = 1 \ldots N$, can be derived from the pcf $P_{cum}(Z)$ as follows. The function $P_{cum}(Z)$ intersect the probabilities w_i at the quantiles q_i, thereby dividing the Z-axis into N intervals:

$$[-\infty, q_1],\ [q_1, q_2],\ [q_2, q_3],\ \ldots\ ,\ [q_{N-1}, \infty]. \tag{A.7}$$

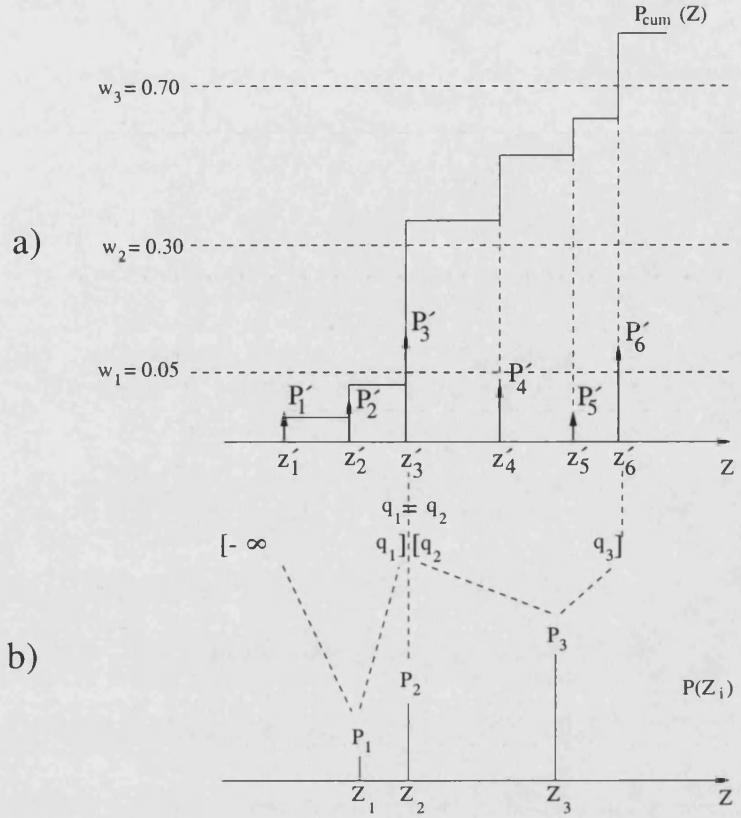


Figure A.1: Combination of probability peaks (z'_l, P'_l) according [211]
a) pcf $P_{cum}(Z)$
b) reduced discrete pdf $P(Z_i) = P_i$

Within the i-th interval there lies a set of probability peaks $(z'_{k_i}, P'_{k_i}), k_i = 1 \ldots K_i$. For example, the peaks (z'_1, P'_1), (z'_2, P'_2) and (z'_3, P'_3) are located within the interval $(-\infty, q_1)$ according to Figure A.1. The set of probability peaks $(z'_{k_i}, P'_{k_i}), k_i = 1 \ldots K_i$, is now represented by a single peak (Z_i, P_i) where $P_i = w_i - w_{i-1}$. A sensible choice for the position Z_i of this single peak is the mean value of the contributing peaks (z'_{k_i}, P'_{k_i}):

$$Z_i = \frac{1}{P_i} \sum_{k_i=1}^{K_i} P'_{k_i} z'_{k_i}. \tag{A.8}$$

The question is how to handle the peaks (z'_l, P'_l) which lie directly on an interval limit q_i. To solve this problem a peak-splitting procedure has been proposed in [211]. This procedure is explained with help of the peak (z'_3, P'_3) according Figure A.1. This peak lies on the interval limit q_1 and causes the pcf $P_{cum}(Z)$ to intersect the probabilities w_1 and

w_2. These probabilities divide the peak (z_3', P_3') into 3 partial peaks which have all the same position z_3':

$$\begin{aligned} (z_3', P_3') &\longrightarrow (z_3', P_3'(1)),\ (z_3', P_3'(2)),\ (z_3', P_3'(3)) \\ P_3' &= P_3'(1) + P_3'(2) + P_3'(3). \end{aligned} \tag{A.9}$$

The first partial peak $P_3'(1)$ is assigned to the first interval $[-\infty, q_1]$ to complement the respective peak sum:

$$P_1 = w_1 - w_0 = P_1' + P_2' + P_3'(1). \tag{A.10}$$

Consequently, the partial peak $(z_3', P_3'(1))$ is taken into account for the mean value determination of Z_1 according to equation (A.8). The second partial peak $P_3'(2)$ has the value $P_2 = w_2 - w_1$ and forms directly the new peak (Z_2, P_2) with $Z_2 = q_2$. The respective interval $[q_1, q_2]$ is degenerated to the point $q_1 = q_2$. The remainder of the peak (z_3', P_3'), the third partial peak $(z_3', P_3'(3))$, is assigned to the interval $[q_2, q_3]$ and is taken into account in the mean value determination of Z_3 of the peak (Z_3, P_3).

A.3 Derivation of Pdf Equations

In this section, the conditions for the pdfs P_α and P_β are derived. The starting point are the original conditions (eq. 6.40) for the pdfs P_{ij}, $P_u(i)$ and $P_v(j)$. Using equation (6.43) within (6.40), the first result term becomes

$$P_u(i) = \sum_j P_{ij} = \sum_j P_{\alpha(i,j)} \cdot P_{\beta(i,j)}. \tag{A.11}$$

By using $\alpha(i,j) = i - j$ and $\beta(i,j) = i + j$ (eq. 6.42) and changing notation $P_{\alpha(i,j)=i-j} \to P_\alpha(i-j)$, $P_{\beta(i,j)=i+j} \to P_\beta(i+j)$, this term becomes

$$P_u(i) = \sum_j P_\alpha(i-j) \cdot P_\beta(i+j). \tag{A.12}$$

Let $j' = j + i$ be the new summing index. Then

$$\begin{aligned} P_u(i) &= \sum_{j'} P_\alpha(2i - j') \cdot P_\beta(j') \\ &= (P_\alpha * P_\beta)(2i). \end{aligned} \tag{A.13}$$

with the symbol "$*$" for discrete convolution. Similarly, the second result term from equation set (6.40) becomes

$$\begin{aligned} P_v(j) &= \sum_i P_{ij} = \sum_i P_{\alpha(i,j)} \cdot P_{\beta(i,j)} \\ &= \sum_i P_\alpha(i - j) \cdot P_\beta(i + j) = \sum_{i'} P_\alpha(i' - 2j) \cdot P_\beta(i') \quad \text{with} \quad i' = i + j \\ &= \sum_{i'} P_{\overline{\alpha}}(2j - i') \cdot P_\beta(i') \qquad\qquad \text{with} \quad P_{\overline{\alpha}}(k) = P_\alpha(-k) \\ &= (P_{\overline{\alpha}} * P_\beta)(2j), \end{aligned} \tag{A.14}$$

The third equation of (6.40) states

$$\rho_{uv} = \frac{\sum_{i,j} P_{ij}(u_i - \overline{u})(v_j - \overline{v})}{\sigma_u \sigma_v} \tag{A.15}$$

Assuming that the discretization points of u and v are almost equidistant[1] the following interpolation is valid

$$u_i \approx \overline{u} + i\Delta u - \overline{i}\Delta u, \qquad v_j \approx \overline{v} + j\Delta v - \overline{j}\Delta v, \tag{A.16}$$

where Δu, Δv are the lengths of the discretization intervals and $\overline{i}$, $\overline{j}$ are the mean values of the indices i and j respectively. Then, equation (A.15) becomes

$$\begin{aligned} \rho_{uv} &= \frac{1}{\sigma_u \sigma_v} \sum_{i,j} P_{ij}(i\Delta u - \overline{i}\Delta u)(j\Delta v - \overline{j}\Delta v) \\ &= \frac{\Delta u^2}{\sigma_u^2} \sum_{i,j} P_{ij}(i - \overline{i})(j - \overline{j}) \end{aligned} \tag{A.17}$$

where the relation $\frac{\Delta u}{\sigma_u} = \frac{\Delta v}{\sigma_v}$ has been used. Now, the summing process is written in terms of the indices α and β. This is achieved by using $i = \frac{1}{2}\alpha + \frac{1}{2}\beta$ and $j = -\frac{1}{2}\alpha + \frac{1}{2}\beta$ (eq.

[1]The discretization (step 1 of QA, Section 6.2.1) is accomplished in such a way that the discretization intervals are almost equidistant.

6.42) and applying the relations $\frac{\sigma_u}{\Delta u} = \sigma_i$ and $\sigma_i^2 = \frac{1}{4}(\sigma_\alpha^2 + \sigma_\beta^2)$:

$$\begin{aligned}
\rho_{uv} &= \frac{1}{\sigma_i^2} \sum_{\alpha,\beta} P_\alpha P_\beta \left(\frac{\alpha+\beta}{2} - \overline{\frac{\alpha+\beta}{2}} \right) \left(\frac{\beta-\alpha}{2} - \overline{\frac{\beta-\alpha}{2}} \right) \\
&= \frac{1}{4} \frac{1}{\sigma_i^2} \sum_{\alpha,\beta} P_\alpha P_\beta \left[(\beta - \overline{\beta})^2 - (\alpha - \overline{\alpha})^2 \right] \\
&= \frac{\sigma_\beta^2 - \sigma_\alpha^2}{\sigma_\beta^2 + \sigma_\alpha^2} . \qquad \text{(A.18)}
\end{aligned}$$

A.4 Solution of Pdf Equations

In this section, the pdf equations

$$P_u(i) = (P_\alpha * P_\beta)(2i), \qquad \text{(A.19)}$$

$$P_v(j) = (P_{\overline{\alpha}} * P_\beta)(2j), \qquad \text{(A.20)}$$

$$\rho_{uv} = \frac{\sigma_\beta^2 - \sigma_\alpha^2}{\sigma_\beta^2 + \sigma_\alpha^2}, \qquad \text{(A.21)}$$

are solved to find P_α and P_β for given $P_u(i)$, $P_v(j)$ and ρ_{uv}. According Section 6.2.1, all random variables are described by the same set of probability peaks $(P_1, \ldots, P_N)$. Consequently $P_u(i)$ and $P_v(j)$ are of the form

$$P_u(i) = P_v(i) = P_i, \qquad \text{(A.22)}$$

where P_i is a discretized normal pdf according to equation (6.23). Since the convolution "$*$" of two normal pdfs results again in a normal pdf, equations (A.19) and (A.20) can be solved by choosing P_α and P_β as normal pdfs. A normal pdf is completely defined by specifying its mean value and standard deviation. In terms of $\overline{\alpha}$, $\overline{\beta}$, σ_α and σ_β, the pdf equations become

$$\bar{i} = \frac{1}{2}\overline{\alpha} + \frac{1}{2}\overline{\beta} \qquad \text{(A.23)}$$

$$\bar{j} = \frac{1}{2}\overline{\beta} - \frac{1}{2}\overline{\alpha} \qquad \text{(A.24)}$$

$$\sigma_i^2 = \frac{1}{4}(\sigma_\alpha^2 + \sigma_\beta^2) \qquad \text{(A.25)}$$

$$\sigma_j^2 = \frac{1}{4}(\sigma_\alpha^2 + \sigma_\beta^2)(= \sigma_i^2) \tag{A.26}$$

$$\rho_{uv} = \frac{\sigma_\beta^2 - \sigma_\alpha^2}{\sigma_\beta^2 + \sigma_\alpha^2}. \tag{A.27}$$

These equation are to be solved for the mean values ($\overline{\alpha}$, $\overline{\beta}$) and for the standard deviations (σ_α, σ_β). To simplify, the indices i and j will be translated in such a way that $\overline{i} = \overline{j} = 0$. Then, equations (A.23) to (A.27) are solved by

$$\overline{\alpha} = 0, \tag{A.28}$$

$$\overline{\beta} = 0, \tag{A.29}$$

$$\sigma_\alpha = \sigma_i\sqrt{2(1-\rho_{uv})}, \tag{A.30}$$

$$\sigma_\beta = \sigma_i\sqrt{2(1+\rho_{uv})}. \tag{A.31}$$

A.5 Discretization Error Considerations

In the case of $\rho_{uv} \approx 1$ the normal pdf P_α becomes very narrow since $\sigma_\alpha = \sigma_i\sqrt{2(1-\rho_{uv})}$. As the sampling points $\alpha = -N, -N+1, \ldots, N$ have a constant distance of 1 the pdf P_α cannot be sampled sufficiently accurate in such a situation. The consequence can be clarified by distinguishing between the target standard deviation of a normal pdf which is $\sigma_t = \sigma_i\sqrt{2(1-\rho_{uv})}$ in the above case, and the real standard deviation σ_r which is realized by the discretized normal pdf:

$$\sigma_r = \frac{1}{M}\sum_{i=-N}^{N} e^{-\frac{(i-\overline{i})^2}{2\sigma_t^2}}(i-\overline{i})^2 = \frac{1}{M}\sum_{i=-N}^{N} e^{-\frac{i^2}{2\sigma_t^2}} i^2 \tag{A.32}$$

$$\text{with} \quad M = \sum_{i=-N}^{N} e^{-\frac{i^2}{2\sigma_t^2}} \tag{A.33}$$

where $\overline{i} = 0$ has been used. For a narrow pdf ($\sigma_t < 1$) the real standard deviation σ_r is plotted versus σ_t in Figure A.2. Obviously, the real standard deviation differs from the target standard deviation ($\sigma_r < \sigma_t$) because of the discretization. Mathematically, the situation can be explained as follows. For $\sigma_t \ll 1$ only the contributions of the terms with

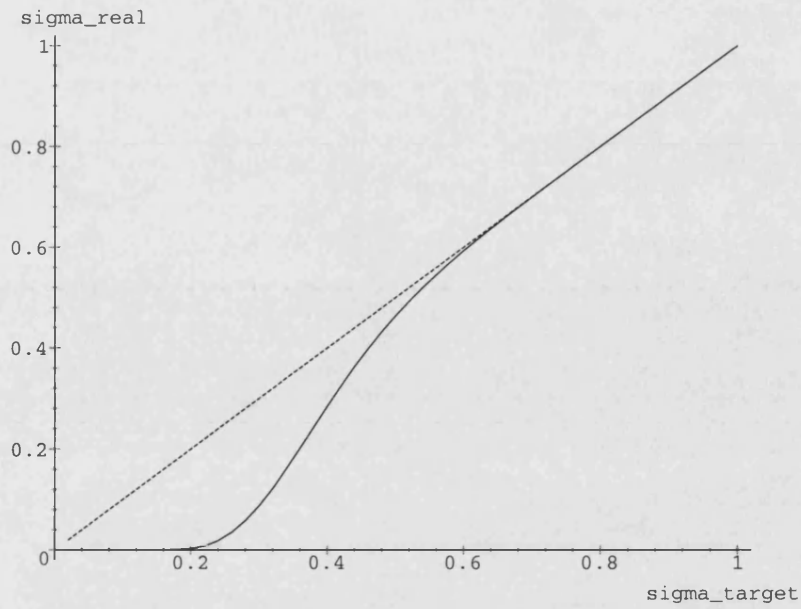


Figure A.2: Real standard deviation σ_r versus target standard deviation σ_t
solid line: σ_r calculated from discrete normal pdf (eq. A.32)
dotted line: $\sigma_r = \sigma_t$ (ideal situation)

$i = -1, 0, 1$ are significant in the summing in equations (A.32) and (A.33). All other terms can be neglected because of the strong damping of the exponential factor. Consequently, equations (A.32) and (A.33) can be approximated by

$$\sigma_r \approx \frac{1}{M} \sum_{i=-1}^{1} e^{-\frac{i^2}{2\sigma_t^2}} i^2 \qquad \text{for} \quad \sigma_t \ll 1 \tag{A.34}$$

$$\text{with} \quad M \approx \sum_{i=-1}^{1} e^{-\frac{i^2}{2\sigma_t^2}} \tag{A.35}$$

which can be simplified to

$$\sigma_r \approx \frac{2}{e^{\frac{1}{2\sigma_t^2}} + 2} \qquad \text{for } \sigma_t \ll 1 \tag{A.36}$$

The quality of the approximation is illustrated in Figure A.3. The approximation is good for $\sigma_t < 0.6$. On the other hand side, it can be seen from Figure A.2 that $\sigma_r \approx \sigma_t$ for $\sigma_t > 0.6$. Consequently, a good approximation for σ_r for the whole value range of σ_t is

$$\sigma_r = \begin{cases} \frac{2}{e^{\frac{1}{2\sigma_t^2}} + 2} & : \quad \sigma_t < 0.6 \\ \sigma_t & : \quad \sigma_t \geq 0.6 \end{cases} \tag{A.37}$$

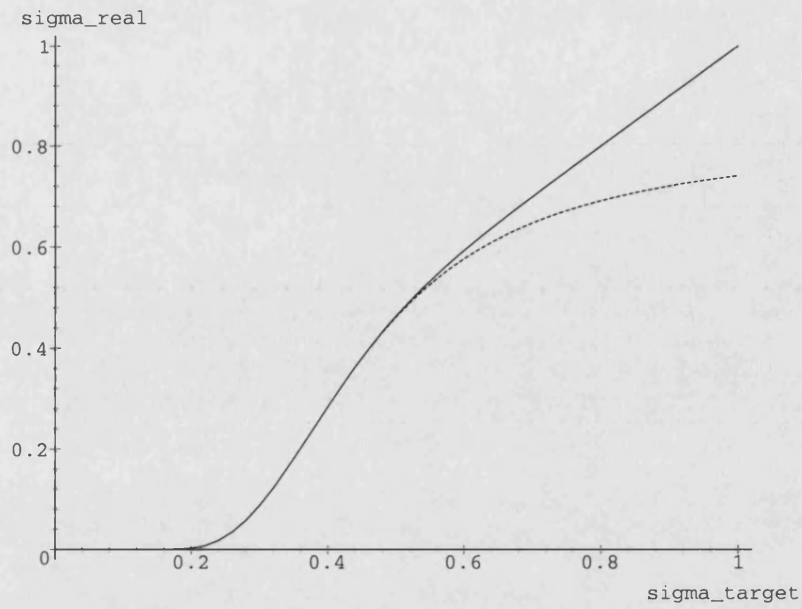


Figure A.3: Quality of approximation formula (A.36),
solid line: σ_r calculated from equation (A.32)
dotted line: σ_r calculated using approximation formula (A.36)

This approximation can now be used to apply a correction $\sigma_t \longrightarrow \sigma_t'$ such that the real standard deviation becomes identical to the wished target value $\sigma_r = \sigma_t$. Solving equation (A.37) for σ_t yields

$$\sigma_t = \begin{cases} \frac{1}{\sqrt{2\ln(\frac{2}{\sigma_r^2}-2)}} & : \quad \sigma_r < 0.6 \\ \sigma_r & : \quad \sigma_r \geq 0.6 \end{cases} \tag{A.38}$$

This formula says how to choose σ_t to achieve a required σ_r. Consequently, changing on the left hand side of equation (A.38) notation $\sigma_t \longrightarrow \sigma_t'$ and applying the requirement $\sigma_r = \sigma_t$ on the right hand side yields

$$\sigma_t' = \begin{cases} \frac{1}{\sqrt{2\ln(\frac{2}{\sigma_t^2}-2)}} & : \quad \sigma_t < 0.6 \\ \sigma_t & : \quad \sigma_t \geq 0.6 \end{cases} \tag{A.39}$$

In MQA, formula (A.39) has been implemented to correct for discretization errors in case of narrow pdfs P_α and P_β. In all tested cases, a good agreement of the σ_r and σ_t has been achieved which allows the correct treatment of highly correlated variables in MQA.

References

[1] P. Shepherd, 'Integrated Circuit Design, Fabrication and Test', Macmillan Press Ltd, 1996.

[2] K. Baker, A. Richardson and A. Dorey, 'Mixed Signal Test – Techniques, Applications and Demands', IEE Proceedings Circuits Devices Systems, Vol. 143, No. 6, 1996.

[3] 'Defect-Oriented Testing – the Future Test Perspective to Specification Test', Panel Session, 4th IEEE International Mixed-Signal Testing Workshop, The Hague, 1998.

[4] V. Agrawal and S. Seth, 'Test Generation for VLSI Chips', Computer Society Order Number 786, 1988.

[5] F. Fantini and C. Morandi, 'Failure Modes and Mechanisms for VLSI ICs – a Review', IEE Proceedings, Vol. 132, Pt. G, No. 3, pp. 74-81, 1985.

[6] J. Poage, 'Derivation of Optimum Tests to Detect Faults in Combinational Circuits', Proceedings Symposium on Mathematical Theory of Automata, Polytechnic Press, New York, pp. 483-528, 1963.

[7] R. Wadsack, 'Fault Modeling and Logic Simulation of CMOS and MOS Integrated Circuits', Bell Syst. Tech. Jour., Vol. 57, pp. 1449-1474, 1978.

[8] K. Mei, 'Bridging and Stuck-at Faults', IEEE Transactions Computers, Vol. C-23, pp. 720-727, 1974.

[9] P. Dahlgren, 'Switch-Level Bridging Fault Simulation in the Presence of Feedback', Proceedings International Test Conference, pp. 363-371, 1998.

[10] S. Thatte and J. Abraham, 'Test Generation for Microprocessors', IEEE Transactions Computers, Vol. C-29, pp. 429-441, 1980.

[11] V. Agrawal, 'When to Use Random Testing', IEEE Transactions Computers, Vol. C-27, pp. 1054-1055, 1978.

[12] J. Roth, W. Bouricius and P. Schneider, 'Programmed Algorithms to Compute Tests and to Detect and Distinguish Between Failures in Logic Circuits', IEEE Transactions Electronic Computers, Vol. EC-16, pp. 567-580, 1967.

[13] F. Sellers, M. Hsiao and L. Bearnson, 'Analyzing Errors with Boolean Difference', IEEE Transactions Computers, Vol. C-17, pp. 676-683, 1968.

[14] P. Nigh, D. Vallett, A. Patel and J. Wright, 'Failure Analysis of Timing and IDDq-only Failures from the SEMATECH Test Methods Experiment', Proceedings International Test Conference, pp. 43-52, 1998.

[15] M. Soma, 'Challenges in Analog and Mixed Signal Fault Models', IEEE Circuits and Devices Magazine (USA), Vol. 12, No. 1, pp. 16-19, 1996.

[16] K. Eckersall, G. Taylor, I. Bell and C. Toumazou, 'Current Monitoring for Test', Circuits & Systems, Tutorials, IEEE International Symposium on Circuits and Systems, London, pp. 551-568, 1994.

[17] P. Collins, S. Yu, K. Eckersall, B. Jervis, I. Bell and G. Taylor, 'Application of Kohonen and Supervised Forced Organisation Maps to Fault Diagnosis in CMOS Opamps', Electronic Letters, Vol. 30, No. 22, pp. 1846-1847, 1994.

[18] R. Tucker and L. McNamee, 'Computer-Aided Design Application to Fault Detection and Isolation Techniques', Proceedings International Symposium on Circuits and Systems, Phoenix, pp. 684-687, 1977.

[19] M. Zwolinski, C. Chalk and B. Wilkins, 'Analogue Fault Modelling and Simulation for Supply Current Monitoring', Proceedings of European Design and Test Conference, Paris, France, pp. 547-552, 1996.

[20] W. Maly, F. Freguson, J. Shen, 'Systematic Characterization of Physical Defects for Fault Analysis of MOS IC Cells', Proceedings International Test Conference, Vol. 2, No. 6, pp. 390-399, 1984.

[21] D. Walker and S. Director, 'VLASIC: A Catastrophic Fault Yield Simulator for Integrated Circuits', IEEE Transactions on Computer-Aided Design, Vol. CAD-5, No. 4, pp. 541-546, 1986.

[22] J. Khare and W. Maly, 'Inductive Contamination Analysis (ICA) with SRAM Application', Proceedings International Test Conference, Washington DC, pp. 552-560, 1995.

[23] J. Khare, C. Kellen and W. Maly, 'CODEF: A Contamination Fault Mapper', SRC-CMU Technical Report, 1995.

[24] S. Kundu and S. Chakravarty, 'Test Challenges for Nano-Meter Designs', tttc Tutorial, European Test Workshop, Konstanz, 1999.

[25] M. Ohletz, 'Realistic Faults Mapping Scheme for the Fault Simulation of Integrated Analogue CMOS Circuits', Proceedings International Test Conference, Test and Design Validity, pp. 776-785, Washington DC, USA, 1996.

[26] S. Tabatabaei and A. Ivanov, 'Schematic-Based Fault Generation', 4th IEEE International Mixed-Signal Testing Workshop, hand-out, The Hague, 1998.

[27] J. Prieto, A. Rueda, I. Grout, E. Peralias, J. Huertas and A. Richardson, 'An Approach to Realistic Fault Prediction and Layout Design for Testability in Analog Circuits', Proceedings of the DATE, 1998.

[28] N. Hamida and B. Kaminska, 'Analog Circuits Testing based on Sensitivity Computation and New Circuit Modeling', Proceedings International Test Conference, pp. 652-661, Baltimore, USA, 1993.

[29] C. Chao, H. Lin and L. Milor, 'Optimal Testing of VLSI Analog Circuits', IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 16, No. 1, 1997.

[30] W. Lindermeir, H. Graeb, K. Antreich, 'Design Based Analog Testing by Characteristic Observation Inference', Proceedings IEEE International Conference on Computer Aided Design, San Jose, pp. 620-626, 1995.

[31] S. Nassif, A. Strojwas and S. Director, 'FABRICS II: A Statistically Based IC Fabrication Process Simulator', IEEE Transactions on Computer-Aided Design, Vol. CAD-3, No. 1, pp. 40-46, 1984.

[32] J. Power, A. Mathewson and W. Lane, 'MOSFET Statistical Parameter Extraction Using Multivariate Statistics', Proceedings IEEE International Conference on Microelectronic Test Structures, pp. 209-214, 1991.

[33] B. Minch, C. Diorio, P. Hasler and C. Mead, 'The Matching of Capacitors for Analog VLSI', Proceedings IEEE, pp. 239-241, 1996.

[34] N. Nagi and J. Abraham, 'Hierarchical Fault Modeling for Linear Analog Circuits', Kluwer Journal Analog Integrated Circuits and Signal Processing, Vol. 10, pp. 89-99, 1996.

[35] C. Pan, K. Cheng and S. Gupta, 'A Comprehensive Macromodel for Opamps', Proceedings IEEE International Conference on Computer Aided Design,, pp. 344-348, 1994.

[36] C. Pan and K. Cheng, 'Fault Macromodeling for Analog/Mixed-Signal Circuits', Proceedings International Test Conference, pp. 913-922, 1997.

[37] M. Zwolinski and C. Tan, 'Characterisation of Analogue Macromodels under Fault Conditions using a Probabilistic Neural Network', Proceedings 5th IEEE International Mixed-Signal Testing Workshop, Delta Whistler, Canada, pp. 157-159, 1999.

[38] T. Berry, 'Fault Modelling for CMOS Analogue Integrated Circuits', Final Year B.Eng. Project, School of Electronic and Electrical Engineering, University of Bath, 1996.

[39] R. Harvey, A. Richardson, E. Bruls and K. Baker, 'Analog Fault Simulation based on Layout Dependent Faults', Proceedings IEEE International Test Conference, pp. 641-649, 1994.

[40] E. Liu, W. Kao, E. Felt and A. Sangiovanni-Vincentelli, 'Analog Testability Analysis and Fault Diagnosis using Behavioral Modeling', Proceedings IEEE Custom Integrated Circuits Conference, pp. 413-416, 1994.

[41] T. Koskinen and P. Cheung, 'Hierarchical Tolerance Analysis Using Statistical Behavioral Models', IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 15, No. 5, pp. 506-516, 1996.

[42] V. Liberali, F. Maloberti, M. Poletti and C. Siviero, 'A Parameter-Space Approach to Behavioral Analog Testing', Proceedings 4th IEEE International Mixed-Signal Testing Workshop, The Hague, p. 298-303, 1998.

[43] P. Duhamel and J. Rault, 'Automatic Test Generation Techniques for Analog Circuits and Systems: A Review', IEEE Transactions on Circuits and Systems, Vol. CAS-26, No. 7, pp. 411-439, 1979.

[44] J. Bandler and A. Salama, 'Fault Diagnosis of Analog Circuits', Proceedings of the IEEE, Vol. 73, No. 8, pp. 1279-1325, 1985.

[45] M. Soma, 'Automatic Test Generation Algorithms for Analogue Circuits', IEE Proceedings Circuits Devices Systems, Vol. 143, No. 6, 1996.

[46] P. Lin and Y. Elcherif, 'Analogue Circuits Fault Dictionary – New Approaches and Implementation', International Journal of Circuit Theory and Applications, Vol. 13, pp. 149-172, 1985.

[47] M. Al-Qutayri and P. Shepherd, 'On the Testing of Mixed-Mode Integrated Circuits', Journal of Semicustom ICs, Vol. 7, No. 4, pp. 32-39, Elsevier Science Punlishers Ltd., England, 1990.

[48] V. Prasad and N. Babu, 'On Minimal Set of Test Nodes for Fault Dictionary of Analog Circuit Fault Diagnosis', Journal of Electronic Testing, Kluwer, Vol. 7, pp. 255-258, 1995.

[49] K. Arabi and B. Kaminska, 'Oscillation-Test Strategy for Analog and Mixed-Signal Integrated Circuits', Proceedings 14th VLSI Test Symposium, pp. 476-482, 1996.

[50] R. Spina and S. Upadhyaya, 'Linear Circuit Fault Diagnosis Using Neuromorphic Analyzers', IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing, Vol. 44, No. 3, pp. 188-196, 1997.

[51] F. Lin, Z. Lin and T. Lin, 'A Uniform Approach to Mixed-Signal Circuit Test', International Journal of Circuit Theory and Applications, Vol. 25, pp. 81-93, 1997.

[52] N. Sen and R. Saeks, 'Fault Diagnosis for Linear Systems via Multifrequency Measurements', IEEE Transactions on Circuits and Systems, Vol. CAS-26, No. 7, pp. 457-465, 1979.

[53] N. Hamida, K. Saab, D. Marche and B. Kaminska, 'LIMSoft: Automated Tool for Design and Test Integration', Proceedings 2nd IEEE International Mixed-Signal Testing Workshop, Quebec City, pp. 56-71, 1996.

[54] J. Bandler, R. Biernacki and A. Salama, 'A Linear Programming Approach to Fault Location in Analog Circuits', Proceedings IEEE International Symposium on Circuits and Systems, pp. 256-260, 1981.

[55] L. Hu, Z. Huang, Y. Huang and R. Liu, 'A Stochastic Model for Analog Fault Diagnosis with Tolerance', Proceedings IEEE International Symposium on Circuits and Systems, pp. 680-683, 1984.

[56] Z. F. Huang, C.-S. Liu, 'Node-Fault Diagnosis and a Design of Testability', IEEE Transactions on Circuits and Systems, Vol. CAS-30, pp. 257-265, 1983.

[57] C. Wey, 'Design of Testability for Analogue Fault Diagnosis', International Journal of Circuit Theory and Applications, Vol. 15, pp. 123-142, 1987.

[58] C.-K. Ho, P. R. Shepherd, W. Tenten, R. Kainer, 'Improvements in Analogue Fault Diagnosis Techniques', Proceedings Second IEEE International Mixed-Signal Testing Workshop, Quebec City, pp. 81-97, 1996.

[59] W. Huang and C. Wey, 'Diagnosability Analysis of Analogue Circuits', International Journal of Circuit Theory and Applications, Vol. 26, pp. 439-451, 1998

[60] C. Wey and W. Huang, 'Test Point Selection Process and Diagnosability Analysis of Analogue Integrated Circuits', Proceedings IEEE International Conference on Computer Design: VLSI in Computers & Processors, Austin, USA, pp. 582-587, 1998.

[61] T. Wei, M. Wong and Y. Lee, 'Fault Diagnosis of Large Scale Analog Circuits based on Symbolic Method', Proceedings 3rd IEEE International Mixed-Signal Testing Workshop, Seattle, pp. 3-8, 1997.

[62] A. Salama and J. Starzyk, 'A Unified Decomposition Approach for Fault Location in Large Analog Circuits', IEEE Transactions on Circuits and Systems, Vol. CAS-31, pp. 609-622, 1984.

[63] V. Agrawal, 'Testing in a Mixed-Signal World', Proceedings 9th Annual IEEE International ASIC Conference and Exhibition, pp. 241-244, Rochester, NY, USA, 1996.

[64] A. Chatterjee and N. Nagi, 'Design for Testability and Built-In Self-Test of Mixed Signal Circuits: a Tutorial', Proceedings 10th International Conference on VLSI Design, pp. 388-392, 1997.

[65] K. Wagner and T. William, 'Design for Testability of Mixed Signal Integrated Circuits', Proceedings International Test Conference, pp. 823-828, 1988.

[66] C. Wey and S. Krishnan, 'Built-In Selft-Test Structures for Analogue Circuit Fault Diagnosis with Current Test Data', IEEE Transactions on Instrumentation and Measurements, Vol. 41, No. 4, pp. 535-539, 1992.

[67] A. Bratt, R. Harvey, A. Dorey and A. Richardson, 'Design-for-Test Structure to Facilitate Test Vector Application with low Performance Loss in Non-Test Mode', Electronic Letters, Vol. 29, No. 16, pp. 1438-1440, 1993.

[68] D. Vazquez, J. Huertas and A. Rueda, 'Reducing the Impact of DFT on the Performance of Analog Integrated Circuits: Improved Sw-Opamp Design', Proceedings of 14th VLSI Test Symposium, pp. 42-47, Princeton, USA, 1996.

[69] M. Soma, 'A Design for Testability Methodology for Active Analogue Filters', Proceedings IEEE International Test Conference, pp. 183-192, 1990.

[70] M. Soma and V. Kolarik, 'A Design for Test Technique for Switched Capacitor Filters', Proceedings VLSI Test Symposium, pp. 42-47, 1994.

[71] E. Teraoka, 'A Built-In Self-Test for ADC and DAC in a Single-Chip Speech CODEC', Proceedings International Test Conference, pp. 791-796, 1993.

[72] M. Toner and G. Roberts, 'A BIST SNR, Gain Tracking and Frequency Response Test of a Sigma-Delta ADC', IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing, Vol. 42, pp. 1-15, 1995.

[73] A. Lechner, A. Richardson and B. Hermes, 'Reconfiguration-Based Buit-In Self-Test for Analogue Front-End Circuits', Proceedings of International Mixed-Signal Testing Workshop, pp. 243-247, Whistler, Canada, 1999.

[74] M. Ohletz, 'Hybrid Built-In-Self-Test (HBIST) for Mixed Analogue/Digital Integrated Circuits', Proceedings European Test Conference, pp. 307-316, 1991.

[75] G. Huertas, D. Vazques, A. Rueda and J. Huertas, 'Oscillation-Based Test Experiments in Filters: A DTMF Example', Proceedings of International Mixed-Signal Testing Workshop, pp. 249-253, Whistler, Canada, 1999.

[76] A. Chatterjee, B. Kim and N. Nagi, 'DC Built-In Self-Test for Linear Analog Circuits', IEEE Design and Test of Computers, pp. 26-33, 1996.

[77] M. Renovell, F. Azais and Y. Bertrand, 'Analog Signature Analyzer for Analog Circuits: BIST implementation', Proceedings 2nd International Mixed-Signal Testing Workshop, pp. 233-238, Quebec City, Canada, 1996.

[78] 'P1149.4, Draft Standard for a Mixed-Signal Test Bus', © by the IEEE Inc., D18, DS5535, New York, USA, 1997.

[79] C.-K. Ho, 'Fault Diagnosis and Design for Testability applied to Analogue Integrated Circuits', PhD Thesis, Dept. Electronic & Electrical Engineering, University of Bath, Bath, U.K., 1998.

[80] C.-K. Ho, P. R. Shepherd, F. Eberhardt and W. Tenten, 'Hierarchical Approach to Analogue Fault Diagnosis', Proceedings 3rd IEEE International Mixed-Signal Testing Workshop, Seattle, pp. 25-30, 1997.

[81] C.-K. Ho, P. R. Shepherd, W. Tenten and F. Eberhardt, 'Improvements to Circuit Diagnosis through Hierarchical Modelling', IEE Colloquium for Testing Mixed Signal Circuits & Systems, 23rd October 1997, Savoy Place, London.

[82] W. Mayeda, 'Graph Theory', John Wiley, 1972.

[83] R. A. DeCarlo & R. Saeks, 'Interconnected Dynamical Systems', M. Dekker, New York, 1981.

[84] Y.-H. Chang and H.-T. Sheu, 'Unified Relaxation-Pseudocircuit Approach for Analogue-Digital Fault Diagnosis', IEE Proceedings Circuits Devices Systems, Vol. 142, No. 4, pp. 236-246, 1995.

[85] H.-T. Sheu and Y.-H. Chang, 'The Relaxation Pseudocircuit Method For Analogue Fault Diagnosis', International Journal of Circuit Theory and Applications, Vol. 24, pp. 201-221, 1996.

[86] C.-C. Wu, K. Nakajima, C.-L. Wey and R. Saeks, 'Analog Fault Diagnosis with Failure Bounds', IEEE Transactions on Circuits and Systems, Vol. CAS-29, No. 5, pp. 277-284, 1982.

[87] C. Wey and R. Saeks, 'On the Implementation of an Analogue TPG: the Linear Case', IEEE Transactions on Instrumentation and Measurement, Vol. 1M-34, No. 3, pp. 442-449, 1985.

[88] C. Wey, R. Saeks, 'On the Implementation of an Analogue TPG: the Nonlinear Case', IEEE Transactions on Instrumentation and Measurement, Vol. 37, No. 2, pp. 252-258, 1988.

[89] M. Hassoun and P. Lin, 'A Hierarchical Network Approach to Symbolic Analysis of Large-Scale Networks', IEEE Transactions on Circuits and Systems, Vol. 42, No. 4, pp. 201-211, April 1995.

[90] 'The Saber Simulator', Analogy Inc., Beaverton, USA.

[91] W. Current, J. Parker and W. Hardaker, 'Behavioural Modelling of Analogue and Mixed-Signal Circuits for Efficient Computer Aided Design', International Journal Electronics, Vol. 80, No. 1, pp. 63-76, 1996.

[92] R. Rosenberger and S. Huss, 'A Systems Theoretic Approach to Behavioral Modeling and Simulation of Analog Functional Blocks', Proceedings of the DATE, pp. 721-728, 1998.

[93] Y. Ju, V. Rao and R. Saleh, 'Consistency Checking and Optimization of Macromodels', IEEE Transactions on Computer-Aided Design, Vol. 10, No. 8, pp. 957-967, 1991.

[94] A. Konczykowska and M. Bon, 'Automated Design Software for Switched-Capacitor ICs with Symbolic Simulator SCYMBAL', Proceedings Design Automation Conference 1988, pp. 363-368.

[95] H. Walscharts, G. Gielen and W.Sansen, 'Symbolic Simulation of Analog Circuits in s- and z-Domain' Proceedings IEEE International Symposium on Circuits and Systems, pp. 814-817, 1989.

[96] Z. Shang and J. Sewell, 'Development of Efficient Switched Network and Mixed-Mode Simulators', IEE Proceedings Circuits Devices Systems, Vol. 145, No. 1, pp. 24-34, 1998.

[97] C.-K. Ho, 'User Manual for Version 1 of the Diagnosis Program', School of Electronic and Electrical Engineering, University of Bath, 1997.

[98] W. Chen, 'The Circuits and Filters Handbook', CRC Handbook, IEEE Press, pp. 2388.

[99] G. Gielen, Z. Wang and W. Sansen, 'Optimal Fault Detection for Analogue Circuits under Manufacturing Tolerances', Electronics Letters, Vol. 32, No. 1, pp. 33-34, 1996.

[100] P. Lin, 'Symbolic Network Analysis', Amsterdam, Elsevier Science, The Netherlands, 1991.

[101] G. Gielen and W. Sansen, 'Symbolic Analysis for Automated Design of Analog Integrated Circuits', MA: Kluwer Academic Publishers, Norwell, 1991.

[102] G. Gielen, P. Wambacq and W. Sansen, 'Symbolic Analysis Methods and Applications for Analog Circuits: A Tutorial Overview', Proceedings of the IEEE, Vol. 82, No. 2, pp. 287-304, Feb. 1994.

[103] M. M. Hassoun and L. P. Huelsman, 'Symbolic Circuit Analysis: An Overview', 38th Midwest Symposium on Circuits and Systems, Proceedings, Vol. 2, Rio de Janeiro, Brazil, 13-16 Aug., 1995.

[104] F. Fernandez and A. Rodriguez-Vazquez, 'Symbolic Analysis Tools – The State-Of-The-Art', Proceedings International Symposium Circuits and Systems, pp. 798-801, 1996.

[105] W. Sansen, G. Gielen and H. Walscharts, 'A Symbolic Simulator for Analog Circuits', in Proceedings ISSCC 1989, pp. 204-205.

[106] G. Gielen, H. Walscharts and W. Sansen, 'ISAAC: A Symbolic Simulator for Analog Integrated Circuits', IEEE Journal Solid-State Circuits, Vol. 24, No. 6, pp. 1587-1597, Dec. 1989.

[107] F. Fernandez, A. Rodriguez-Vazquez and J. Huertas, 'A Tool for Symbolic Analysis of Analog Integrated Circuits including Pole/Zero Extraction', Proceedings European Conference on Circuit Theory and Design, pp. 752-761, 1991.

[108] F. Fernandez, A. Rodriguez-Vazquez and J. Huertas, 'Interactive ac Modeling and Characterization of Analog Circuits via Symbolic Analysis', Kluwer Journal Analog Integrated Circuits and Signal Processing, Vol. 1, pp. 183-208, Nov. 1991.

[109] S. Seda, M. Degrauwe and W. Fichtner, 'A Symbolic Analysis Tool for Analog Circuit Design Automation', Proceedings International Conference Computer-Aided Design, pp. 488-491, 1988.

[110] S. Seda, M. Degrauwe and W. Fichtner, 'Lazy-Expansion Symbolic Expression Approximation in SYNAP', Proceedings International Conference Computer-Aided Design, pp. 310-317, 1992.

[111] S. Manetti, 'New Approaches to Automatic Symbolic Analysis of Electronic Circuits', Proceedings IEE, Pt. G, pp. 22-28, Feb. 1991.

[112] G. Wierzba et al., 'SSPICE – A Symbolic SPICE Program for Linear Active Circuits', in Proceedings Midwest Symposium on Cicuits and Systems, pp. 1197-1201, 1989.

[113] L. Huelsman, 'Personal Computer Symbolic Analysis Programs for Undergraduate Engineering Courses', Proceedings International Symposium on Circuits and Systems, pp. 798-801, 1989.

[114] G. Nebel, U. Kleine and H. Pfleiderer, 'Symbolic Pole/Zero Calculation Using SANTAFE', IEEE Journal of Solid-State Circuits, Vol. 30, No. 7, pp. 752-761, 1995.

[115] S. Jou, M. Perng and C. Su, 'Hierarchical Techniques for Symbolic Analysis of Electronic Circuits', IEE Proceedings Circuits Devices Systems, Vol. 144, No. 3, 1997.

[116] A. Vladimirescu, A. Newton and D. Pederson, 'SPICE Version 2G.1 User's Guide', Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Oct. 1980.

[117] M. Hassoun and F. Fernandez, 'Pleasures, Perils and Pitfalls of Symbolic Analysis', Proceedings International Symposium on Circuits and Systems, Vol. 1, Ch. 122, pp. A 451 - A 457 CAD, 1994.

[118] W. Daems, W. Verhagen, P. Wambacq, G. Gielen and W. Sansen, 'Evaluation of Error-Control Strategies for the Linear Symbolic Analysis of Analog Integrated Circuits', IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications, Vol. 46, No. 5, 1999.

[119] G. Gielen, H. Walscharts and W. Sansen, 'Analog Circuit Design Optimization based on Symbolic Simulation and Simulated Annealing', IEEE Journal Solid-State Circuits, Vol. 25, No. 3, pp. 707-713, June 1990.

[120] H. Koh, C. Squin and P. Gray, 'OPASYN: A Compiler for CMOS Operational Amplifiers', IEEE Transactions Computer-Aided Design, Vo, 9, No. 2, pp. 113-125, Feb. 1990.

[121] J. Vital, N. Horta, N. Silva and J. Franca, 'CATALYST: A Highly Flexible CAD Tool for Architecture Level Design and Analysis of Data Converters', Proceedings European Design Automation Conference, pp. 472-477, 1993.

[122] F. Medeiro, B. Perez-Verdu, A. Rodriguez-Vazquez and J. Huertas, 'A Tool for Automated Design of Sigma-Delta Modulators using Statistical Optimization', Proceedings International Symposium on Circuits and Systems, pp. 1373-1376, 1993.

[123] M. Degrauwe et al., 'The ADAM Analog Design Automation System', Proceedings International Symposium on Circuits and Systems, pp. 820-822, 1990.

[124] K. Swings and W. Sansen, 'ARIADNE: Constrained-based Approach to Computer-Aided Synthesis and Modeling of Analog Integrated Circuits', Kluwer Journal Analog Integrated Circuits and Signal Processing, Vol. 3, No. 2, pp. 37-55, May 1993.

[125] F. Leyn, W. Daems, G. Gielen and W. Sansen, 'Analog Circuit Sizing with Constraint Programming Modeling and Minimax Optimization', Proceedings International Symposium on Circuits and Systems, pp. 1500-1503, 1997.

[126] S. Jou, K. Liu and C. Su, 'Circuits Design Optimization Using Symbolic Approach', Journal of the Chinese Institute of Electrical Engineering, Vol. 4, No. 1, pp. 51-60, 1997

[127] A. Konczykowska and M. Bon, 'Analog Design Optimization using Symbolic Approach', Proceeding International Symposium on Circuits and Systems, pp. 786-789, 1991.

[128] S. Manetti and M. Piccirilli, 'Symbolic Simulators for the Fault Diagnosis of Nonlinear Analog Circuits', Kluwer Journal Analog Integrated Circuits and Signal Processing, Vol. 3, No. 1, pp. 59-72, Jan. 1993.

[129] J. Lee and R. Rohrer, 'AWEsymbolic: Compiled Analysis of Linear(ized) Circuits using Asymptotic Waveform Evaluation', Proceedings Design Automation Conference, pp. 213-218, 1992.

[130] M. Glesner and K. Haubrichs, 'A New Efficient Statistical Tolerance Analysis and Design Procedure', Proc. IEEE CADEMICS, Hull, Great Britain, 1977.

[131] C. Borchers, R. Sommer and E. Henning, 'On the Symbolic Calculation of Nonlinear Circuits', Proceedings International Symposium on Circuits and Systems, pp. 719-722, 1996.

[132] W. Van Petegem, B. Geeraerts and W. Sansen, 'Electrothermal Simulation and Design of Integrated Circuits', IEEE Journal of Solid-State Circuits, Vol. 29, No. 2. pp. 143-146, Feb. 1994.

[133] P. Wambacq, G. Gielen and W. Sansen, 'Symbolic Simulation of Harmonic Distortion in Analog Integrated Circuits with Weak Nonlinearities', Proceedings International Symposium on Circuits and Systems, pp. 536-539, 1990.

[134] A. Buonomo, C. Di Bello and P. Di Genio, 'Symbolic Analysis of Nearly Sinusoidal Oscillators', Proceedings International Symposium on Circuits and Systems, pp. 1736-1739, 1997.

[135] G. Dröge, T. Czysz and E. Horneber, 'Symbolic Pole and Zero Estimation for Circuit Design', Proceedings ICECS, pp. 93-96, 1996.

[136] J. Starzyk and A. Konczykowska, 'Flowgraph Analysis of Large Electronic Networks', IEEE Transactions Circuits and Systems, Vol. 33, No. 3, pp. 302-315, 1986.

[137] C. Shi and X. Tan, 'Symbolic Analysis of Large Analog Circuits with Determinant Decision Diagrams', Proceedings International Symposium on Circuits and Systems, pp. 366-373, 1997.

[138] X. Tan and C. Shi, 'Hierarchical Symbolic Analysis of Large Analog Circuits with Determinant Decision Diagrams', Proceedings International Symposium on Circuits and Systems, pp. VI-318 - VI-321, 1998.

[139] M. Hassoun and K. McCarville, 'Symbolic Analysis of Large-Scale Networks using a Hierarchical Signal Flowgraph Approach', Kluwer Journal Integrated Circuits and Signal Processing, Vol. 3, No. 1, pp. 31-42, Jan. 1993.

[140] C. Fridas and J. Sewell, 'Symbolic Analysis of Networks by Partitioned Polynomial Interpolation', IEEE Transactions on Circuits and Systems, Vol. CAS-21, No. 3, pp. 345-347, 1974.

[141] Z. Shang and J. Sewell, 'Accurate Semi-Symbolic Analysis of Large Non-Ideal Switched Linear Networks', Proceedings International Symposium on Circuits and Systems, pp. 1408-1411, 1995.

[142] I. Garcia-Vargas, M. Galan, F. Fernandez and Rodriguez-Vazquez, 'An Algorithm for Numerical Reference Generation in Symbolic Analysis of Large Analog Circuits', Proceedings European Design and Test Conference, pp. 395-399, 1997.

[143] P. Wambacq, F. Fernandez, G. Gielen, W. Sansen and A. Rodriguez-Vazquez, 'Efficient Symbolic Computation of Approximated Small-Signal Characteristics of Analog Integrated Circuits', IEEE Journal of Solid-State Circuits, Vol. 30, No. 3, pp. 327-330, 1995.

[144] J. Katzenelson and A. Unikovski, 'Symbolic-Numeric Circuit Analysis or Symbolic Circuit Analysis with Online Approximations', IEEE Transactions on Circuits and Systems–I: Fundamental Theory and Applications, Vol. 46, No. 1, pp. 197-207, Jan. 1999.

[145] A. Konczykowska and J. Starzyk, 'Computer Justification of Upward Topological Analysis of Signal-Flow Graphs', Proceedings European Conference on Circuit Theory and Design, The Hague, pp. 464-467, 1981.

[146] A. Konczykowska and M. Bon, 'Symbolic Simulation for Efficient Repetitive Analysis and Artificial Intelligence Techniques in C.A.D.', Proceedings International Symposium on Circuits and Systems, pp. 802-805, 1989.

[147] A. Sangiovanni-Vincentelli, L. Chen and L. Chua, 'An Efficient Heuristic Cluster Algorithm for Tearing Large-Scale Networks', IEEE Transactions on Circuits and Systems, Vol. CAS-24, No. 12, Dec. 1977.

[148] M. Hassoun and P. Lin, 'An Efficient Partitioning Algorithm for Large-Scale Circuits', Proceedings International Symposium on Circuits and Systems, New Orleans, pp. 2405-2408, 1990.

[149] F. Eberhardt, W. Tenten, C.-K. Ho and P. Shepherd, 'A Structural Approach to Hierarchical Tolerance Modelling of Analogue CMOS Integrated Circuits', Proceedings 4th IEEE International Mixed-Signal Testing Workshop, The Hague, p. 288-297, 1998.

[150] C. Ho, A. Ruehli and Brennan, 'The Modified Nodal Approach to Network Analysis', IEEE Transactions on Circuits and Systems, Vol. CAS-25, pp. 504-509, June 1975.

[151] J. Starzyk and J. Zou, 'Direct Symbolic Analysis of Large Analog Networks', Proceedings of the 39th Midwest Symposium on Circuits and Systems, pp. 425-428, 1997.

[152] M. Pierzchala and B. Rodanski, 'Efficient Generation of Symbolic Network Functions for Large-Scale Circuits', Proceedings of the 39th Midwest Symposium on Circuits and Systems, pp. 421-424, 1997.

[153] W. M. Smith, 'Worst Case Circuit Analysis - An Overview', Proceedings of the IEEE Annual Reliability and Maintainability Symposium, pp. 326-334, 1996.

[154] G. Strube, 'Robuste Verfahren zur Worst-Case- und Ausbeute-Analyse Analöger Integrierter Schaltungen', PhD dissertation, University of Munich, 1998.

[155] A. Ilumoka and R. Spence, 'A Sensitivity based Approach to Tolerance Assignment', Proc. IEE, Pt.G, 129(4), pp. 139-149, 1982.

[156] B. Fabien, 'The Determination of Feedback Gains for Uncertain Linear Systems Using Higher-Order Sensitivity Minimization', Optimal Control Applications & Methods, Vol. 14, pp. 259-269, 1993.

[157] W. Tao, M. Wong and Y. Lee, 'Fault Diagnosis of Large Scale Analog Circuits based on Symbolic Method', Chinese Journal of Electronics, Vol. 7, No. 4, pp. 395-399, 1998.

[158] R. Brayton, R. Spence, 'Sensitivity and Optimization', Elsevier, Amsterdam, 1980.

[159] T. Ellman, J. Keane, A. Banerjee and G. Armhold, 'A Transformation System for Reformulation of Design Optimization Strategies', Research in Engineering Design, Springer London, Vol. 10, pp. 30-61, 1998.

[160] K. Reinschke and P. Schwarz, 'Verfahren zur Rechnergestützten Analyse Linearer Netzwerke', Band 9, Akademie Verlag Berlin, 1976.

[161] S. W. Director and R. A. Rohrer, 'The Generalized Adjoint Network and Network Sensitivities', IEEE Transactions on Circuit Theory, Vol. CT-16, No. 3, pp. 318-323, August 1969.

[162] G. Li, 'On Pole and Zero Sensitivity of Linear Systems', IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications, Vol. 44, No. 7, pp. 583-590, 1997.

[163] L. O. Chua and P. M. Lin, 'Computer Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques', Englewood Cliffs, New Jersey: Prentice-Hall, 1975.

[164] A. Liberatore and S. Manetti, 'Network Sensitivity Analysis via Symbolic Formulation', Proceedings International Symposium on Circuits and Systems, pp. 705-708, 1989.

[165] S. Chang and G. Wierzba, 'Symbolic Sensitivity Analysis using SSPICE', Proceedings International Symposium on Circuits and Systems, pp. 1043-1046, 1992.

[166] Y. Cheng and R. Fujii, 'SAUCES: a Sensitivity Analysis Program for Analog Circuit Design', Proceedings International Symposium on Circuits and Systems, pp. 1175-1178, 1992.

[167] P. M. Lin, 'Sensitivity Analysis of Large Linear Networks using Symbolic Programs', Proceedings International Symposium on Circuits and Systems, San Diego, pp. 1145-1148, May 1992.

[168] J. Echtenkamp, M. Hassoun, G. Prabhu and C. Wright, 'Hierarchical Sensitivity Analysis for Sequence of Expressions Method', Proceedings of the European Conference on Circuit Theory and Design, pp. 75-78, 1995.

[169] J. Echtenkamp and M. H. Hassoun, 'Implementation Issues for Symbolic Sensitivity Analysis', Proceedings of the 39th Midwest Symposium on Circuits and Systems, Vols. I-III, Ch. 319, pp. 429-432, 1997.

[170] W.-K. Chen, 'Applied Graph Theory', North-Holland Publishing Company, Amsterdam - London, 1971.

[171] F. Eberhardt, W. Tenten and P. R. Shepherd, 'Improvements in Hierarchical Symbolic Tolerance & Sensitivity Analysis', Electronics Letters, Vol. 35, No. 4, pp. 261-263, Feb. 1999.

[172] 'Maple V Release 3, Computerprogramm für Symbolische Mathematik',©1981-1994 by Waterloo Maple Software and the University of Waterloo.

[173] F. Eberhardt, W. Tenten and P. R. Shepherd, 'Symbolic Parametric Fault Simulation & Diagnosis of Large Scale Linear Analogue Circuits', Proceedings 5th IEEE International Mixed-Signal Testing Workshop, Vancouver, Canada, pp. 221-228, 1999.

[174] R. Voorakaranam, S. Chakrabarti, J. Hou, A. Gomes, S. Cherubal and A. Chatterjee, 'Hierarchical Specification-Driven Analog Fault Modeling for Efficient Fault Simulation and Diagnosis', Proceedings International Test Conference, pp. 903-912, 1997.

[175] N. Ben-Hamida, K. Saab, D. Marche and B. Kaminska, 'FaultMaxx: A Perturbation Based Fault Modeling and Simulation for Mixed-Signal Circuits', Proceedings Asian Test Symposium, pp. 182-187, 1997.

[176] H. Yoon, P. Variyam, A. Chatterjee and N. Nagi, 'Hierarchical Statistical Inference Model for Specification Based Testing of Analog Circuits', Proceedings of the 16th IEEE VLSI Test Symposium, pp. 145-150, 1998.

[177] G. Devarayanadurg, P. Goteti and M. Soma, 'Hierarchy based Statistical Fault Simulation of Mixed-Signal ICs', Proceedings International Test Conference, pp. 521-527, 1996.

[178] A. Balivada, H. Zheng, N. Nagi, A. Chatterjee and J. Abraham, 'A Unified Approach for Fault Simulation of Linear Mixed-Signal Circuits', Journal of Electronic Testing: Theory and Applications, Vol. 9, pp. 29-41, 1996.

[179] M. Zwolinski, A. Brown and C. Chalk, 'Concurrent Analogue Fault Simulation', Proceedings 3rd IEEE International Mixed Signal Testing Workshop, Seattle, pp. 42-47, 1997.

[180] J. Hou and A. Chatterjee, 'CONCERT: A Concurrent Fault Simulator for Analog Circuits', Proceedings 4th IEEE International Mixed Signal Testing Workshop, The Hague, pp. 3-8, 1998.

[181] J. Soares Augusto and C. Beltran Almeida, 'Fast Fault Simulation in Linear and Nonlinear Circuits with Fault Rubber Stamps', Proceedings 4th IEEE International Mixed Signal Testing Workshop, The Hague, pp. 28-37, 1998.

[182] F. Eberhardt, W. Tenten and P. R. Shepherd, 'Parallel Symbolic Sensitivity Analysis

of Large-Scale Analogue Circuits', submitted to IEEE Transactions on Circuits and Systems.

[183] M. Glesner, K. Haubrichs and H. Linsler, 'Statistische Toleranzuntersuchungen an elektrischen Netzwerken mit Hilfe der verallgemeinerten Quantilarithmetik', Archiv für Elektronik und Übertragungstechnik (Electronics and Communication), Band 32, Heft 1, pp. 1-7, Jan. 1978.

[184] F. Eberhardt, W. Tenten and P. R. Shepherd, 'Symbolic Tolerance Analysis of Large Scale Analogue Circuits', submitted to IEE Proceedings Circuits Devices Systems.

[185] R. Spence and R. Soin, 'Tolerance Design of Electronic Circuits', Addison-Wesley Publishing Company, Wokingham, England, 1988.

[186] S. Director, P. Feldmann and K. Krishna, 'Optimization of Parametric Yield: a Tutorial', International Journal of High Speed Electronics, Vol. 3, No. 1, pp. 95-136, 1992

[187] R. Wilton, 'Yield Analysis Tools for Analogue Circuit Design', Electronic Engineering, pp. 56-59, 1997.

[188] W. Tian, X. Ling and R. Liu, 'Novel Methods for Circuit Worst-Case Tolerance Analysis', IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications, Vol. 43, No. 4, pp. 272-278, 1996.

[189] J. Bandler, P. Liu and J. Chen, 'Worst Case Network Tolerance Optimization', IEEE Transactions, MTT, Vol. 23, No. 8, pp. 630-641, 1975.

[190] M. Lulu, 'Designing Quality (Yield) and Reliability into Circuits', Quality Engineering, Vol. 8, No. 3, pp. 383-393, 1996.

[191] K. Antreich, H. Gräb and C. Wieser, 'Circuit Analysis and Optimization driven by Worst-Case Distances', IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems CAD, Vol. 13, pp. 57-71, 1994.

[192] K. Krishna and S. Director, 'The Linearized Performance Penalty (LPP) Method for Optimization of Parametric Yield and Its Reliability', IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 14, No. 12, 1995.

[193] J. P. Spoto, W. J. Coston and C. P. Hernandez, 'Satistical Integrated Circuit Design and Characterization', IEEE Transactions Computer-Aided Design, CAD-5(1), pp. 90-103, 1986.

[194] A. Ponnambalam and J. Vlach, 'A Unified Approach to Statistical Design Centering of Integrated Circuits with Correlated Parameters', IEEE Transactions on Circuits and Systems–I, Vol. 46, No. 1, pp. 190-196, 1999.

[195] J. Hammersly and D. Handscombe, 'Monte Carlo Methods', Methuen, London, 1964.

[196] W. Huang, J. Resh and C. Wey, 'On Synthesis of Manufacturable and Testable Analogue Integrated Circuits', Proceedings of 1998 Midwest Symposium on Circuits and Systems, pp. 340-343, 1998.

[197] D. Papakostas, G. Papouktsoglou and A. Hatzopoulos, 'Estimation of Statistical Variables for Analog Circuit Parameter Evaluation', Proceedings ICECS, Cairo, Egypt, pp. 740-743, 1997.

[198] M. Styblinski, 'Tolerance Analysis and Optimization of Electronic Circuits', Proceedings European Confonference on Circuit Theory and Design, pp. 97-114, Warsaw, Poland, 1980.

[199] J. Chen and M. Styblinski, 'A Systematic Approach of Statistical Modeling and Its Application to CMOS Circuits', Proceedings International Symposium on Circuits and Systems, pp. 1805-1808, 1993.

[200] C. Kurker, J. Paulos, R. Gyurcsik and J. Lu. 'Hierarchical Yield Estimation of Large Analog Integrated Circuits', IEEE Journal of Solid-State Circuits, Vol. 28, No. 3, pp. 203-209, 1993.

[201] J. Chen and A. Yang, 'STYLE: A Statistical Design Approach Based on Nonparametric Performance Macromodeling', IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 14, No. 7, pp. 794-802, 1995.

[202] M. Lie and L. Milor, 'Computing Parametric Yield Adaptively Using Local Linear Models', Proceedings 33rd Design Automation Conference, pp. 831-835, 1995.

[203] J. Swidzinski, D. Alexander, M. Qu and M. Styblinski, 'A Systematic Approach to Statistical Simulation of Complex Analog Integrated Circuits', Proceedings International Workshop on Statistiacal Metrology, pp. 86-89, 1997.

[204] D. Enright and R. Mack, 'A Hierarchical Technique to Model Parametric Device Variation in Non-Linear Analogue CMOS Architectures', IEE Colloquium (Digest), No. 331, pp. 1/1-1/7, 1997.

[205] T. Scott and T. Walker 'Regionalization: A Method for Generating Joint Density Estimates', IEEE Transactions on Circuit and Systems, Vol. 23(4), pp. 229-234, 1976.

[206] D. Gibson, R. Poddar, G. May and M. Brooke, 'Statistically Based Parametric Yield Prediction for Integrated Circuits', IEEE Transactions on Semiconductor Manufacturing, Vol. 10, No. 4, pp. 445-458, 1997.

[207] S. Director and G. Hachtel, 'The Simplicial Approximation Approach to Design Centering', IEEE Transactions on Circuits and Systems, Vol. 24, No. 7, pp. 363-372, 1977.

[208] H. Abdel-Malek and A. Hassan, 'The Ellipsoidal Technique for Design Centering and Region Approximation', IEEE Transactions on Computer-Aided Design, Vol. 10, No. 8, pp. 1006-1013, 1991.

[209] M. Dempster, 'Distributions in Intervals and Linear Programming, in: Topics in Interval Analysis', Clarendon Press, Oxford, 1969.

[210] L. Starzyk and E. Sliwa, 'Tolerances in Symbolic Network Analysis', Proceedings International Symposium on Circuits and Systems, pp. 810-813, 1989.

[211] K. Haubrichs, 'Entwicklung eines schnellen Toleranzanalyseverfahrens als Alternative zur Monte Carlo Technik bei der Untersuchung elektrischer Netzwerke', PhD thesis, University of Saarbrücken, 1977.

[212] V. Petrov, 'Limit Theorems of Probability Theory', Oxford University Press, Oxford, 1995.

[213] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, 'Numerical Recipes in C', Cambridge University Press, New York, USA, 1992.

[214] K. Diem and C. Lentner, 'Wissenschaftliche Tabellen', Documenta Geigy, 7th edition, Ciba-Geigy AG, Basel, 1973.

[215] P. Gill, W. Murray and M. Wright, 'Numerical Linear Algebra and Optimization', Vol. 1, §4.9.2, Redwood City, CA: Addison-Wesley, 1991.

[216] P. Peebles, 'Probability, Random Variables, and Random Signal Principles', McGraw-Hill, third Edition, New York, 1993.

[217] German Government Funded Project, SSE/P13, 'VIRTUS: Virtuelle, Analoge und Mixed-Signal Testentwicklung und Verifikation'.

[218] W. Huang and C. Wey, 'Development of HDL-A Modeled Test Programs for Fault Diagnosis of Analog/Mixed-Signal Circuits', Proceedings 3rd IEEE International Mixed-Signal Testing Workshop, Seattle, USA, pp. 9-14, 1997.

[219] C.-L. Wey, Michigan State University, USA, Private Communications, May 1999.