TESTING AND CHARACTERIZATION OF ANALOG SYSTEMS USING BEHAVIORAL MODELS AND OPTIMAL EXPERIMENTAL DESIGN

by

Eric James Felt

Memorandum No. UCB/ERL M96/68

15 November 1996

COVER PAGE

126

TESTING AND CHARACTERIZATION OF ANALOG SYSTEMS USING BEHAVIORAL MODELS AND OPTIMAL EXPERIMENTAL DESIGN

by

Eric James Felt

Memorandum No. UCB/ERL M96/68

15 November 1996

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Abstract

Testing and Characterization of Analog Systems Using Behavioral Models and Optimal Experimental Design

by

Eric James Felt

Doctor of Philosophy in Engineering—Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Alberto L. Sangiovanni-Vincentelli, Chair

The presence of analog components in today's complex mixed-signal systems complicates their testing and statistical characterization significantly. Analog circuits, in general, are much more difficult to test than digital circuits because performance specifications must be considered and because few design technology tools are available to aid in the design of the test vectors or the analysis of the testing results. Analog testing is currently performed on a relatively ad-hoc basis, with test suites frequently defaulting to the complete set of circuit specifications. This approach is becoming increasingly expensive in both test development and test execution times.

This dissertation presents a methodology for solving analog testing and characterization problems more *systematically* by using *statistical techniques*, especially linear models and optimal design of experiments. These statistical techniques are applied to the problems of automatic test pattern generation, optimal test structure design, and hierarchical statistical performance characterization. This systematic methodology allows analog systems to be characterized more accurately and more efficiently. Since testing currently accounts for approximately 30-50% of total product cost, these algorithms are expected to make a significant impact on the overall cost of designing and manufacturing analog systems.

Professor Alberto Sangiovanni-Vincentelli
Dissertation Committee Chair

Contents

List of Figures v

List of Tables vii

1 Introduction 1
- 1.1 Design Methodology for Analog Systems . . . 2
- 1.2 Testing of Analog Systems . . . 2
- 1.3 Overview . . . 5
  - 1.3.1 Analog ATPG . . . 5
  - 1.3.2 Optimal Test Structures . . . 8
  - 1.3.3 Statistical Characterization . . . 8
- 1.4 Main Contributions . . . 9
- 1.5 Organization . . . 10

2 Previous Work 11
- 2.1 Practical Approaches . . . 12
- 2.2 Fault Diagnosis . . . 13
  - 2.2.1 Linear Network Theory . . . 13
  - 2.2.2 Fault Dictionaries . . . 14
  - 2.2.3 Artificial Intelligence . . . 15
  - 2.2.4 Summary . . . 16
- 2.3 Fault-Based Testing . . . 16
  - 2.3.1 Fault Modeling and Simulation . . . 17
  - 2.3.2 Topological Approaches . . . 17
  - 2.3.3 Catastrophic Fault Coverage . . . 17
  - 2.3.4 Parametric Fault Coverage . . . 19
  - 2.3.5 Ordering of Tests . . . 20
  - 2.3.6 Types of Input Stimuli . . . 21
  - 2.3.7 Power Supply Current Monitoring . . . 21
  - 2.3.8 Summary . . . 23
- 2.4 Performance-Based Testing . . . 24
- 2.5 Design for Testability (DFT) . . . 25
  - 2.5.1 Analog Scan . . . 26

2.5.2 Built-In Self-Test (BIST) 27
2.6 Conclusion 28

3 Automatic Test Pattern Generation for Linear Circuits 29
3.1 Introduction 29
3.2 Motivation 30
3.3 Algorithm 32
3.3.1 Optimality Criteria 35
3.3.2 Optimization 36
3.4 Model Derivation 39
3.4.1 Eliminating Ambiguity Groups 40
3.4.2 Calculating Confidence Intervals 42
3.5 Results 43
3.6 Conclusions 46

4 Automatic Test Pattern Generation for Nonlinear Analog Circuits 49
4.1 Algorithm 50
4.1.1 Nonlinear regression 51
4.1.2 Robust QR-decomposition 53
4.2 Results 54
4.2.1 Bandpass Filter 55
4.2.2 MOS Transistor 58
4.3 Conclusions 59

5 Designing Optimal Analog Test Structures 63
5.1 Introduction 63
5.2 Methodology 64
5.2.1 Calculating D-Values 65
5.2.2 Generating D-Optimal Test Sets 67
5.3 Results 67
5.4 Conclusions 74

6 Statistical Characterization of Analog Circuits 75
6.1 Introduction 75
6.2 Parameter Mismatch and Correlation 78
6.3 Analytic Statistical Calculations 79
6.4 Correlated Parameters at the Behavioral Level 84
6.5 Results 85
6.5.1 Folded Cascode Operational Amplifier 86
6.5.2 Phase-Locked Loop 87
6.6 Conclusions 98

7 Conclusions 100

Bibliography 101

List of Figures

1.1 Overview of testing within typical product design cycle. 6

3.1 Six switchable current sources connected in parallel. 31
3.2 (a) Confidence intervals from applying Test Set 1. (b) Confidence intervals from applying Test Set 2. 33
3.3 Proposed testing algorithm. 34
3.4 6-bit binary-weighted current source D/A converter. 43
3.5 Upper and lower bounds on INL error from seven test vectors. 47

4.1 Pseudo-code for modified robust Gram-Schmidt orthonormalization. 53
4.2 Bandpass filter with center frequency at 24.5 kHz. 55
4.3 Nominal frequency response of bandpass filter. 56
4.4 Estimated output and 99% confidence intervals for bandpass filter. 57
4.5 SPICE model for MOS transistor. 58
4.6 (a) Sensitivity w.r.t. μ. (b) Sensitivity w.r.t. V_{T0}. (c) Sensitivity w.r.t. C_{ox}. (d) Sensitivity w.r.t. γ. (e) Sensitivity w.r.t. t_{ox}. (f) Sensitivity w.r.t. θ. . . 60
4.7 Estimated response and confidence intervals for MOS transistor from 6 test points. 61
4.8 Standard error of estimated response for MOS transistor. 62

5.1 Pseudo-code of optimal test structure design algorithm. 65
5.2 (a) Proposed test structure for Circuit 1. (b) Proposed test structure for Circuit 2. (c) Proposed test structure for Circuit 3. 68
5.3 From top to bottom, GOSSET programs for Circuits 1, 2, and 3, respectively. 70
5.4 D-value vs. percentage of transistors turned on. 72
5.5 D-value vs. number of tests for each circuit. 73

6.1 Hierarchical characterization. 77
6.2 Example variance-covariance matrix for low-level parameters. 79
6.3 Pseudo-code for computing the Cholesky decomposition of Σ. 85
6.4 Transistor-level schematic of operational amplifier circuit. 86
6.5 Block diagram of PLL. 88
6.6 Extraction using UTMOST. 89
6.7 Histogram of principal component 7 of MOS models. 91

6.8 Histogram of gain. 94
6.9 Histogram of $f_{0.8}$. 95
6.10 Histogram of I_{up}. 96
6.11 Histogram of I_{dn}. 97
6.12 Matrix of correlation coefficients of intermediate-level parameters. 98
6.13 Distribution of lock time. 99

List of Tables

1.1 Summary of relationship between parameters, circuit, and test vectors. . . . 5

3.1 Two sets of test vectors for testing current sources. 31
3.2 Test vectors chosen for D/A converter. 44

4.1 Test frequencies chosen for bandpass filter. 56
4.2 Test points chosen for MOS transistor. 59

5.1 Normalized D-values for minimum-size test sets. 69
5.2 From top to bottom, D-optimal test sets for Circuits 1, 2, and 3, respectively. 71
5.3 Measured mismatch for 10 sets of measurements on 6 "optimal" test structures. 74

6.1 Expected values and standard deviations of intermediate-level parameters. . 85
6.2 Distributions of principal components of MOS models. 90
6.3 Comparison of linear and quadratic models for intermediate-level parameters. 92
6.4 CPU times for building linear and quadratic models of intermediate-level parameters, on a Sun Sparc 20. 92
6.5 Expected values and standard deviations of intermediate-level parameters. . 93
6.6 Comparison of linear and quadratic models for high-level performance. . . . 93
6.7 Expected values and standard deviations of high-level performance. 93

Acknowledgements

I would like to thank my advisor, Professor Alberto Sangiovanni-Vincentelli, for his guidance and support during the past 5 1/2 years. He is an inspiring and visionary leader who can assimilate and refocus months of research in a single five-minute meeting. His foresight guided me to the topic of analog testing in 1992, a topic with wide-open research opportunities and tremendous industrial interest.

I would like to express my gratitude to Professor Paul Gray for serving on my dissertation committee and for his general involvement with our analog CAD research group. His healthy skepticism insured that our research made sense to circuit designers as well as to the design technology community, and he pushed us to prove our research ideas with fabricated silicon whenever possible.

My gratitude also extends to statistics Professor Terry Speed for serving as the outside member of my committee. He taught me that statistical formulas must never substitute for common sense and insight. He was an excellent professor whose class inspired me to ground my research firmly in statistical techniques.

I would also like to thank the other professors who are part of our design technology research group, Robert Brayton and Richard Newton. Together with Professor Sangiovanni-Vincentelli they have created a dynamic and productive research group of which I have been proud to be a part. Working with so many brilliant minds and friendly, cooperative colleagues insured that the grueling graduate school life was always intellectually stimulating.

I would like to thank Carlo Guardiani and Stefano Zanella from SGS-Thomson Microelectronics in Italy for the productive joint research which resulted in Chapter 6 of this dissertation.

For their financial support, I am grateful to the companies and organizations that provided my fellowships, namely the California MICRO program, the Armed Forces Communications and Electronics Association, and Lockheed. I would also like to thank the various donors who supported my research by contributing to our design technology research group and my advisor's research funds.

I would like to extend a special thank you to my wonderful wife, Sandy, for her continuous love and support, coping so well with the hardships of graduate student life, and diligently proofreading my papers and this dissertation. Her zest for life has added tremendous fun and excitement to these years in Berkeley. I would also like to thank my

parents, grandparents, and siblings for their love, visits, support, and encouragement.

It has been a pleasure to work with the other members of our analog CAD research group: Henry Chang, Edoardo Charbon, Alper Demir, Edward Liu, Enrico Malavasi, and Iasson Vassiliou. Their motivation and intelligence will take them far in life. I have also enjoyed the camaraderie of my classmates and friends Jagesh Sanghavi, Gitanjali Swamy, and Lisa Guerra, who entered Berkeley with me in 1991, studied for the preliminary exam with me in the summer of 1992, and are now also graduating with their Doctorates. I would also like to thank all of the other friends who have been a positive part of my life in Berkeley. In particular, I have enjoyed many spontaneous and occasionally heated lunchtime discussions with Henry Chang, Desmond Kirkpatrick, Chris Lennard, Ken Nishimura, Mark Noworolski, and Jeff Rothman. I would especially like to thank Desmond Kirkpatrick for his tireless help with Linux and other computer issues.

The University of California, Berkeley is truly a unique institution, and I am thankful to all of the current students, alumni, faculty, and staff for creating and maintaining such a magnificent academic environment.

Chapter 1

Introduction

The complexity of electronic systems being designed today is increasing in many dimensions: on one hand, the number of components is growing constantly; on the other, several radically different functions must be integrated. For example, in the exploding personal communications market, a device is the combination of wireless transmission, analog and digital signal processing, and digital computing. In this device, antennas, radio-frequency components, and analog and digital subsystems have to be designed in a unified way to meet the performance, power, and size requirements of the application.

The intermingling of analog and digital signals in these systems is likely to increase as more functionality is integrated onto a single chip. Analog components are generally used in these systems for two reasons [59]:

1. To interface digital processing with applications-specific environments. Many applications include interactions among electronics and various sensors and actuators; such interactions imply analog signals, since "the real world is analog."

2. To accelerate processing in high-performance systems. Processing analog signals and images will be important to information delivery technologies in the 21st century, and digital processing and transmission limit performance to an extent that is unacceptable for high-performance systems.

Furthermore, decreasing the size of integrated circuits leads to higher frequencies in general, and hence to more analog behavior of even digital signals.

Because of these requirements and trends, analog components are here to stay. Today's design and test methodologies must deal with these components systematically,

accurately, and efficiently.

1.1 Design Methodology for Analog Systems

Our design technology research group at the University of California, Berkeley has developed a new top-down, constraint-driven design methodology for analog and mixed-signal system design [18, 17, 16, 15]. The methodology has two basic goals: (1) making the design cycle robust by use of hierarchical partitioning, behavioral modeling, and specification propagation; (2) drastically reducing the number of design iterations by use of accurate performance evaluation and early error diagnosis. The key points of the methodology are:

- Top-down hierarchical process starting from the behavioral level based on early verification and constraint propagation;
- Bottom-up accurate extraction and verification;
- Automatic and interactive synthesis of components with constraint-driven layout design tools;
- Maximum support for automatic synthesis tools to accommodate users of different levels of expertise, but not the enforcement of these tools upon the user; this is not an automatic synthesis process; and
- Consideration for testability at all stages of the design.

It is this final bullet which is the subject of this dissertation.

1.2 Testing of Analog Systems

According to the Semiconductor Industry Association, in about 10 years design and test methodology and tools must cost-effectively handle 1-GHz microprocessors with 300 million transistors and 16 Gbyte, 0.1-micron DRAMS. Testers that cost under $2M (versus the $50M we would extrapolate from today's situation) must thoroughly test 200-million-gate ASICs with 3,000 I/O pins [87]. With typical testing targets currently being better than 40 ppm defect levels [102], testing these huge systems poses many formidable challenges.

The analog portions of these huge systems complicate their testing and characterization significantly. While less than 20% of a typical mixed-signal ASIC is used for analog circuitry, testing the analog portion is a major problem and one which will cause production bottlenecks as devices integrate higher proportions of analog functions onto mixed-signal chips [29].

Analog circuits, in general, require much longer testing times than digital circuits because second-order effects must be considered and because few design automation tools are available to aid in the design of the test vectors. Analog testing and characterization is currently performed on a relatively ad-hoc basis; a design or test engineer relies primarily upon intuition about a circuit's internal functionality to derive the circuit's test suite. This test suite frequently defaults to the complete set of circuit specifications. This approach is becoming increasingly expensive in both test development and test execution times. The specifications of mixed analog-digital circuits are usually very large (e.g. see [9]), which not only results in long manual test development, but also in prohibitive testing times on very expensive automated test equipment with mixed-signal capabilities; in aggregate, test-related costs for today's electronic products typically range from 30% to 50% of total product cost [13]. Furthermore, the use of sophisticated design automation tools continues to reduce the design cycle time so that the influence of testing on time-to-market and final cost of the circuit is becoming increasingly significant. For these reasons, analog testing is considered to be one of the most important problems in analog and mixed-signal design.

The main factors that make analog circuit testing difficult can be summarized as follows [119]:

- Analog systems are frequently nonlinear, include noise, and have parameter values that vary widely. Thus, deterministic methods are often inefficient for modeling these systems.

- Relations between input and output signals in analog circuits are sometimes complicated compared to those of digital systems. These relations in analog circuits are more difficult to model than digital circuit representations, which are based on classical truth tables and thus are precise and easy to model.

- The statistical distribution of faults in analog systems is generally not known with enough precision. For this reason, probabilistic methods are often ineffective.

- The complexity of today's analog circuits and their many parameters, as well as the limited accessibility to their internal components, restricts the use of conventional automatic test equipment. Such equipment does not have enough storage and lacks the capability of computation during actual testing.

Standard practice tends toward *specification testing* of analog circuits: testing some or all of the response parameters for conformity. For linear circuits (such as filters) the tested parameters could include DC specifications, like input bias currents and impedances; AC specifications, like the gain bandwidth and total harmonic distortion; and transient specifications, like the step response settling time [4].

Verifying the entire set of specifications would provide complete confidence in the tested part. However, the time and cost overheads of such a procedure are high. Moreover, the high redundancy in specifications causes overtesting of the part. Therefore, in practice, manufacturers test only a few specifications over a limited input space. The drawback is that such compromise can lessen the quality of shipped parts.

This dissertation offers alternatives to specification-based testing which will provide benefits such as [13]:

- Shorter time to market,
- Lower manufacturing cost,
- Reduced capital for test equipment,
- Reduced development cost,
- Improved out-of-the-box quality,
- Reduced field-installation time,
- Increased product up-time, and
- Reduced field-maintenance cost.

These new methods use recent advances in *behavioral models* and *optimal design of experiments* to perform analog automatic test pattern generation (ATPG), to design for test (DFT), to build optimal test structures, and to statistically characterize large circuits. Test vectors are chosen to be "maximally orthogonal" so that circuit performance will be

	parameters	circuit	test vectors
Analog ATPG	fixed	fixed	optimize
Optimal test structures	fixed	optimize	optimize
Statistical characterization	fixed	fixed	fixed

Table 1.1: Summary of relationship between parameters, circuit, and test vectors.

characterized as accurately as possible in the presence of measurement noise and model inaccuracies. These techniques allow analog systems to be characterized more accurately and more efficiently than previously possible, thereby significantly reducing system test cost.

1.3 Overview

Figure 1.1 shows an overview of the typical product design cycle and the role of testing and characterization within this process [13]. Our research in analog testing and characterization has impact throughout this design cycle, particularly in the "Design Evaluation" and "Test" stages. There are three major thrusts to our research. The first is analog automatic test pattern generation (ATPG), both for circuits which can be modeled in a linear function space and for highly nonlinear circuits. The second is optimal test structure design, in which design circuits are optimized for measuring a specific set of parameters as accurately as possible. The third is the statistical characterization of analog circuits during the design process.

The relationship between the parameters, circuit, and test vectors for each of these three situations is summarized in Table 1.1.

1.3.1 Analog ATPG

Our approach to automatic test pattern generation involves combining recent advances in *behavioral modeling* of analog integrated circuits with statistical algorithms for *optimal design of experiments.*

The algorithms are based upon the statistical theory of optimal experimental design, in which test vectors are chosen to be maximally orthogonal so that the system performance will be characterized as accurately as possible in the presence of measurement noise and model inaccuracies. More specifically, for an I-optimal design we wish to choose

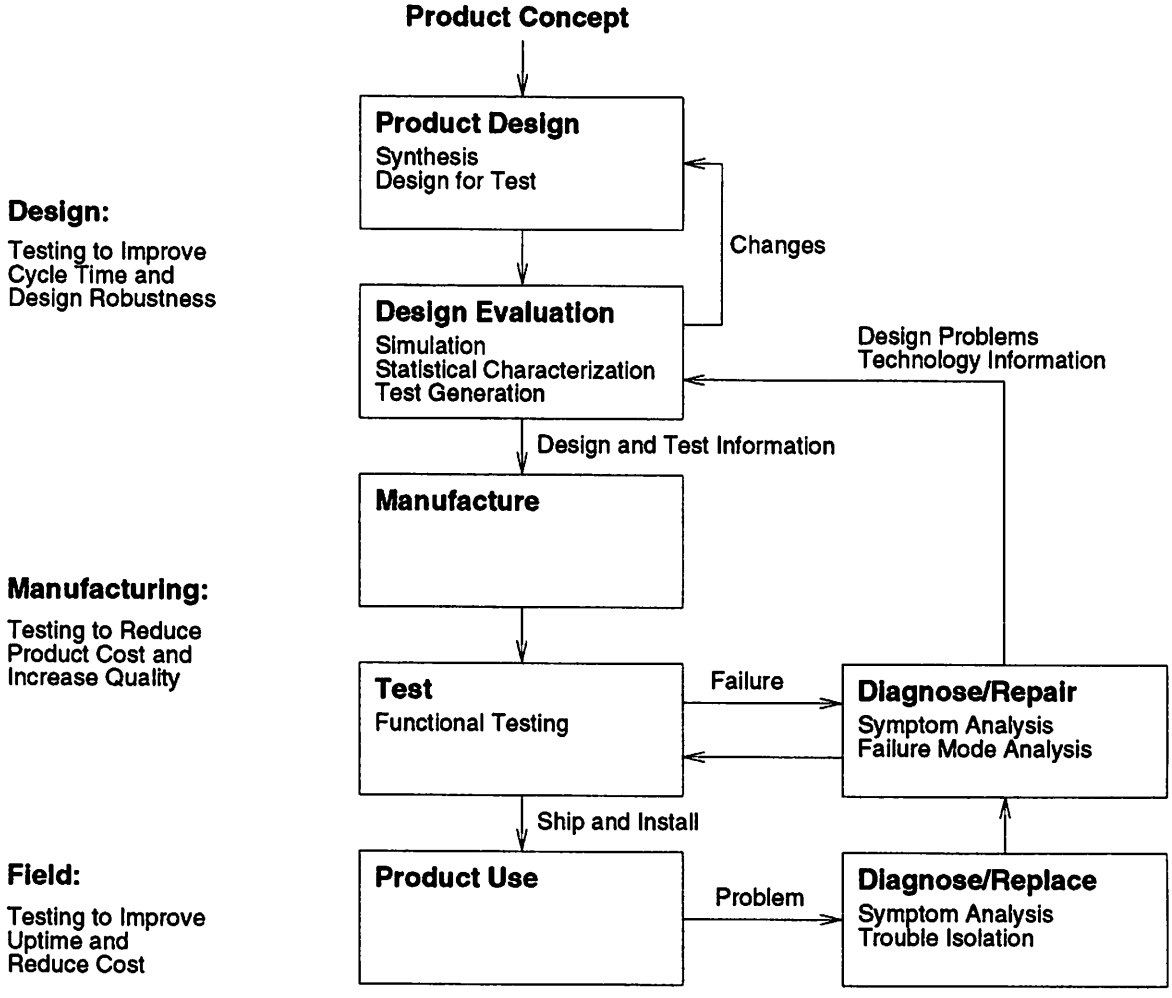


Figure 1.1: Overview of testing within typical product design cycle.

the test vectors to minimize the average standard error of the predicted output, thereby maximizing the likelihood that we will be able to conclusively verify that the performance specifications have or have not been met after applying a minimum number of test vectors. If the minimum number of test vectors is not sufficient to conclusively verify the performance specifications, then additional test vectors are selected and applied, one at a time, until the standard error of the predicted output is low enough to verify the performance specifications. Linear regression is used to analyze the results of the tests and compute the required standard errors.

Finding an exactly I-optimal design is believed to be NP-complete [23] and hence only feasible for very small problems. For larger problems, several heuristic algorithms have been successfully used to find "good" solutions to this and other related problems in the area of optimal experimental design. These heuristic algorithms include simulated annealing [23], greedy swap techniques [89], and gradient descent techniques. For this research we used the gradient descent techniques implemented in the software package GOSSET, which was recently developed by Hardin and Sloane at AT&T Bell Laboratories [48].

The analog testing algorithm we propose is:

1. Linearize about the nominal parameter values and use GOSSET to generate an initial set of n test vectors from this linear model, where n is the dimensionality of the space to be characterized.

2. Apply the initial set of n test vectors.

3. Use iterative techniques (Newton-Raphson) to solve for the actual model parameters.

4. Re-linearize about the actual model parameters and construct confidence intervals for the response function from this model.

5. If the confidence intervals for the response function fall definitely within the system performance specifications at all points, then accept the chip.

6. If the confidence intervals for the response function fall definitely outside the system performance specifications at any point, then reject the chip.

7. If the confidence intervals are too wide to make a conclusive decision, then use GOSSET to generate one additional test vector. Apply the additional test vector and go to Step 3.

For linear circuits the process simplifies because linear regression can be used instead of an iterative technique and because there is no need to re-linearize in Step 4.

1.3.2 Optimal Test Structures

Fast ATPG algorithms also enable the construction of *optimal test structures*, which are circuits optimized to measure a specified set of parameters as efficiently as possible in the presence of measurement noise and other potential errors.

We have developed a figure of merit which can be used to evaluate the efficiency of various candidate test structures. Comparing the efficiencies of test structures will permit a test engineer to determine which structures are optimal for measuring a certain set of interesting parameters. This optimality information, when considered with area, shape, and other factors, can be used to determine which test structures should actually be fabricated.

The figure of merit which we propose is the *normalized D-values of the minimum-size D-optimal test set* for each circuit. The D-value is the average variance of the estimated model coefficients, so a circuit with a lower D-value is a more efficient test structure than a circuit with a higher D-value. The inputs to the algorithm are a set of candidate circuits and a list of parameters of interest. The output is a list of the normalized D-values corresponding to the D-optimal test set for each circuit. Generating the candidate circuits is an open problem. Once some candidate circuits have been generated, though, this algorithm can be used to determine which structure is optimal.

One interesting observation that we have made is that extracting parameters from relatively complex structures is often more efficient than measuring single devices. The reason for this phenomenon is that in the complex circuit each parameter is, in effect, sampled multiple times and hence the error due to measurement noise and model inaccuracies is reduced. This result is similar to measuring a single device multiple times and averaging the measurements, but fewer tests are required when the complex circuit is used.

1.3.3 Statistical Characterization

The same behavioral models that are used for ATPG can be used to statistically characterize analog circuits during the design process. We have developed a methodology for hierarchical statistical circuit characterization which does not rely upon circuit-level Monte Carlo simulation. The methodology uses principal component analysis, response surface

methodology, and statistics to directly calculate the statistical distributions of higher-level parameters from the distributions of lower-level parameters. This methodology permits the statistical characterization of large analog and mixed-signal systems, many of which are extremely time-consuming or impossible to characterize using existing methods.

In this hierarchical design methodology, two statistical characterizations are performed. First, the statistical distributions of the intermediate-level parameters are calculated from those of the low-level parameters. Second, the statistical distributions of the high-level circuit performance is calculated from the intermediate-level parameters. The first characterization can be quickly performed with the non-Monte Carlo techniques described in this dissertation. The second characterization can be performed either in the same way or using Monte Carlo analysis. Monte Carlo analysis is generally acceptable for the second characterization if the behavioral model being used is fast and involves only a relatively small number of statistical parameters, which is often the case.

At each level of the statistical characterization it is essential to consider the *correlations* between parameters, as independent parameters are uncommon. Computing and using a variance-covariance matrix of the parameters at each level of the hierarchy can properly account for parameter correlations and, furthermore, provides an excellent conduit for incorporating parameter mismatch information into circuit models. These variance-covariance matrices are one of the most important cornerstones of our methodology.

1.4 Main Contributions

The field of analog testing and characterization is still relatively new and there are many theoretical and practical issues which have not yet been explored. The previously published research in these areas is reviewed in Chapter 2. We believe that our main contributions to the field are as follows:

1. *Use of statistical confidence intervals for analog testing.* This technique allows the errors due to noise and model inaccuracies to be quantified and properly considered during testing.

2. *Analog ATPG for linear circuits using optimal design.* Techniques based on optimal design of experiments are more theoretically sound and yield better results than previous techniques.

3. *Analog ATPG for nonlinear circuits using optimal design.* Combining Newton-Raphson iterative techniques with optimal design enables the advantages discovered for linear circuits to be extended to nonlinear circuits.

4. *Design for test by incorporating ATPG into analog design cycle.* Minimizing test time is almost never currently considered in the analog design process; our analog ATPG techniques allow test time to be estimated during design.

5. *Optimal test structures.* We have defined the concept of optimal test structures and proposed methods for designing them.

6. *Direct statistical circuit characterization.* Expected values, variances, and correlations of higher-level parameters are calculated from those of lower-level parameters, with proper accounting for parameter correlation and mismatch.

1.5 Organization

This dissertation is organized as follows. Chapter 2 describes previous work in the area of analog testing and characterization and the relationship between the previous work and the content of this dissertation. Chapters 3 and 4 describe the testing and ATPG algorithms, first for circuits which are linear in their statistical parameters and then for nonlinear circuits. Chapter 5 discusses techniques for optimal test structure design. Chapter 6 presents our algorithms for statistical characterization. Chapter 7 concludes.

Chapter 2

Previous Work

This chapter reviews previously published work in the area of analog testing. Most of the previous work in testing of electrical systems has been directed at *digital* circuits, and efficient techniques have been developed for testing both combinational and sequential digital systems [112, 41]. These digital testing techniques are based on the single stuck-at-0/stuck-at-1 fault model and the controllability and observability of each fault.

Unfortunately most of the digital testing ideas cannot be directly applied to analog systems. One of the major differences is that analog systems are, in general, much more difficult to model because second- and higher-order effects can significantly impact system performance. The binary nature of digital circuits is much simpler to model.

Another major difference is that analog circuits are susceptible to both catastrophic and parametric faults, while digital circuits are usually susceptible only to catastrophic faults. Catastrophic faults consist of stuck-at faults, open faults, and bridging faults. The parametric fault class consists of faults that result in performance outside accepted limits and are usually associated with variations in design parameters (e.g. passive component values, device sizes) [75, 78, 124].

The previous research in analog testing can be classified into five main categories: practical approaches, fault diagnosis, fault-based testing, performance-based testing, and design for testability. Each of these categories will be considered, in turn, followed by comments on the relationship between this previous work and the testing methodology presented in this dissertation.

2.1 Practical Approaches

Many of the most frequently discussed analog testing problems concern very practical issues relating to the automatic test equipment (ATE) which is used to test analog and mixed-signal circuits. The mixed-signal tester market is growing at a faster rate than the logic tester market because the percentage of ASICs containing analog components has increased dramatically over the past decade; 40% of all VLSI ASIC devices contain some analog components. LTX and Teradyne dominate the market, with 45% and 30% of dollar volumes, respectively. A typical mixed-signal test system costs $1.5 million [74].

These expensive test systems require input files that specify which inputs are to be applied and which outputs are to be measured. These input files are referred to as the *test program*. In the majority of cases, these test programs are manually created, with an accuracy ranging from 100% to as little as 10% [97]. A surprising amount of research has been directed toward the languages for specifying the test input/output vectors for specific test equipment [12] and, in particular, toward automating the conversion of tests from the designers' environments to the specific languages used by the testers [29, 46].

Tektronix sells a Waveform Analysis and Verification Environment which translates test vectors into the format needed for common ATE equipment [91]. LTX offers a similar product called enVision, which they describe as a "visual test programming language" [94].

Cadence's DANTES (Design and Test Engineering System) is an integrated design and test environment and a tool set for developing tests for analog and mixed-signal ICs. Test tools are tightly integrated with the design tools to let engineers consider test parameters, tester specifications, and testability issues during the design cycle, rather than at its end. DANTES then produces a test program for a specific tester [92].

These commercial analog testing products aid in the writing of test programs in the specific languages used by various mixed-signal testers, but they have generally been quite disappointing because they focus exclusively on practical issues like language and number of inputs and outputs. Determining which tests to apply is still a human decision. In 1989 Runyon predicted that we were still 10 years away from any commercial product which would achieve automatic test program generation [105]; unfortunately, his prediction has held true.

2.2 Fault Diagnosis

Fault diagnosis is the process of locating faults in a system, if any exist, by observing the system outputs under various test conditions. Fault diagnosis is useful because it permits the designer to pinpoint faulty components so that they can be repaired, replaced, or redesigned. It can also be useful in a production test environment, where circuits with no faults are passed and circuits with one or more faults are rejected.

As integrated circuits have increased in size and functionality, fault diagnosis has decreased in importance because defective ICs are generally discarded rather than repaired. In production testing it is usually not necessary to know why a particular IC has failed, merely that it has failed. The exception to this is during system prototyping and debugging, during which fault diagnosis can provide important information about how to modify the design to make it more robust.

Analog fault diagnosis received a thorough theoretical treatment in the 1970s. Duhamel and Rault presented an excellent review of the topic [30]. Here we summarize some of the more recent work.

2.2.1 Linear Network Theory

For systems in which the outputs vary linearly with respect to the possible faults, linear network theory has been used by many researchers to perform fault diagnosis. The underlying concept is that a system of linear equations is derived which relates the system outputs to the possible faults; solving this system of equations diagnoses any faults present in the system.

The testability of a system generally refers to the solvability of this system of diagnostic equations. Saeks developed a quantitative evaluation of this testability in [108]. Temes gave a unique measure of how readily elements can be diagnosed from test terminal measurements, computed from a set of test points on the circuit under test [136]. And Sen linked the measure of solvability of the system to the fault diagnosis equations [115].

Hemink extended this approach to *nonlinear* systems [53, 52]. He combined a rank-test algorithm with statistical methods to find sets of dependent parameters and determine whether it is possible to calculate a certain parameter with sufficient accuracy. Saeks, Visvanathan, and Sangiovanni-Vincentelli analyzed the testability of *dynamic* systems using a similar approach based on sensitivity analysis and the solvability of the resultant equations

[109, 145].

When the system of diagnostic equations is rank deficient and hence not solvable, one can either add more observable test nodes or ignore some of the parameters. An interesting method for selecting additional test nodes using Boolean algebra is presented in [100], and a refinement is presented in [98, 99] which improves the computational complexity from exponential to $O(fp^2)$, where p is the number of nodes and f is the number of faults.

To ignore some of the parameters, one can analyze a circuit's *ambiguity groups*, which are the sets of linearly dependent parameters which cause the diagnostic equations to be rank deficient. Finding ambiguity groups is a computationally intense process which has been studied by many researchers over the years [7, 115, 5, 133, 65]. Once the ambiguity groups are found, some of the parameters can be assumed to be fault free (set to their nominal values) to make the diagnostic equations solvable [67].

Several researchers have applied these linear network theory ideas to the decomposition of large networks into subnetworks [110, 62, 131]. These approaches facilitate testing by localizing the effects of faults to specific subnetworks.

2.2.2 Fault Dictionaries

The fault dictionary is one of the oldest approaches to analog fault diagnosis, and it is still frequently used in industrial environments. Its popularity is due to:

1. The method is easily understood by test engineers,
2. The fault dictionary is easily upgraded,
3. The on-line computation time is small,
4. It can test both linear and nonlinear circuits, and
5. It measures only a small number of test nodes.

To create a fault dictionary, one simulates the fault-free circuit and each of the possible faults, recording the output from each simulation. When a circuit is tested, the measured output is compared to each simulated output and the best match is determined. Large fault dictionaries require very large storage requirements and long search times; to limit the number of dictionary entries, these methods always consider only one fault at a

time and almost always consider only catastrophic faults, since the number of parametric faults is unlimited for real-valued parameters.

Augusto presented a typical implementation in [3], in which he carefully considers manufacturing variations among all parameters. First one good and f faulty circuits are generated, where f is the number of faults. Then a Monte Carlo analysis is performed on each of the $f + 1$ circuits, with statistical variation of the circuit parameters. The node voltages are collected and their means and standard deviations are calculated. Finally, these statistical parameters are used to construct the ambiguity sets for each node (using a heuristic algorithm) and to select the test nodes in order to maximize the fault coverage.

To determine which faults need to be included in the fault dictionary, most implementations use inductive fault analysis (IFA) [116]. This technique, which is also used extensively on digital circuits, involves randomly placing circular "defects" of various radii onto the layout and recording the shorts and opens created by each simulated defect. Sachdev uses IFA directly in his fault dictionary approach [107] to generate realistic fault dictionaries.

2.2.3 Artificial Intelligence

The most popular commercial fault diagnosis systems use artificial intelligence (AI) approaches. In the first AI approach, measurement effects were propagated backwards though the circuit model until a fault was found [73]. The DC voltage was measured at different nodes. The AI engine deduced the values of parameters within the circuits by propagating the effect of measurement through the model. Faults were inferred from the detection of inconsistencies and located by suspending constraints within the model.

ELECTRA is a commercial product based on a different AI approach. ELECTRA uses behavioral models and a special troubleshooting decision tree to diagnose errors, similar to a binary search [117, 101].

AUTOTEST is another AI-based commercial product which performs testability analysis, generating both quantitative and qualitative descriptions of the testability of a given design. The quantitative measures of testability are:

1. Fraction of faults detected,

2. Fault isolation resolution, and

3. Average ambiguity group size.

The qualitative descriptions include identifying feedback loops and unique circuit configurations which are difficult to test. The qualitative analysis consists of a set of design rules that are applied to the circuit [61].

2.2.4 Summary

The problem of determining precisely what is wrong with a circuit has been becoming less and less important for the past three decades as more and more functionality is integrated on-chip. Since defective chips are discarded rather than repaired, the more important questions for most production testing are simply whether or not a circuit is faulty and whether or not a circuit meets its specifications. Thus fault diagnosis is overkill, providing more information than is required. To the extent that this overkill requires more testing time or cost than would be necessary to answer the aforementioned simpler questions, fault diagnosis is wasteful of precious testing resources.

Many researchers claim to be improving testability by increasing the rank of the fault diagnosis matrix by adding additional circuitry to improve the control and/or observability of the circuit, e.g. additional pins or an analog scan architecture [102]. These "testability improvements" may actually increase testing cost, so an extremely careful analysis should be conducted before blindly adopting these proposed improvements; when the goal of testing is to verify circuit functionality and specifications, rather than to diagnose faults, then these testability improvements are rarely of value.

2.3 Fault-Based Testing

Analog circuits are traditionally tested by directly measuring their specifications. This approach has been increasingly questioned over the past decade due to high implementation costs, the difficulties associated with quantifying the effectiveness of the tests, and difficulties in accessing embedded analog sub-blocks [102]. Many researchers have promoted fault-based testing as an alternative to specification-based testing. The idea comes from the testing of digital circuits: since circuits fail because they contain faults, we assume that a circuit meets its specifications if we can verify that it does not contain any faults. In the testing of digital circuits it is generally assumed that at most one fault is present in any given circuit, and most of the fault-based approaches discussed in this section make that same *single-fault assumption.*

2.3.1 Fault Modeling and Simulation

In order to apply fault-based testing techniques one must determine which faults can occur and which faults are to be considered during testing. Daugherty presented the basic concepts of analog fault simulation in [26]. Early work was based on experimental manufacturing defect statistics and showed that open faults and bridging faults are the most frequent catastrophic fault types [70, 148, 157].

Soma and Meixner developed analog fault models by performing Monte-Carlo defect simulations [124, 75]. Meixner models faulty analog behavior as modifications to the nominal macromodel. Nagi published some results on fault modeling for both catastrophic and parametric AC and DC faults in passive and active components [84]. Faults in active components can be modeled at the behavioral level, which allows the method to be applied to larger circuits than methods which only consider the circuit level.

2.3.2 Topological Approaches

Several testing methods have been developed which are based directly on these fault models and the topology of the circuit under test. Wey and Saeks represent a circuit as a set of decoupled state machines together with algebraic connection equations. Using this model they simulate all possible single and double catastrophic faults, and have developed an automatic test pattern generation method for circuits with both linear [152] and nonlinear [153] input/output transfer functions.

Marlett developed a path sensitization method which can be used for DC test generation using a resistive shunt model [72]. Naiknaware published a similar idea which can be used hierarchically. A test model is stored with each generic block in a cell library. The test model is represented by a sequence of tests to be performed on the block. To generate the test plan for the chip, Naiknaware finds the chip input values that need to be applied to produce the desired block inputs and the chip output values that need to be measured to detect the appropriate block outputs [86].

2.3.3 Catastrophic Fault Coverage

Catastrophic faults are generally considered to be random defects that cause failures in various components. They may be structural deformations like short and open circuits, or cause large variations in design parameters (e.g., a change in a single transis-

tor's length-to-width ratio caused by a dust particle on a photolithographic mask) [78]. Several researchers have developed automatic test pattern generation techniques for detecting catastrophic faults in analog systems using DC tests. Given that the designer can identify the critical parameters in the design and supply a model of process fluctuations, Milor described an efficient algorithm in [78]. Her algorithm is based on fault signatures similar to those used for constructing fault dictionaries, but seeks only to distinguish between faulty and fault-free circuits, rather than to fully diagnose causes of failure. She concluded that observing the primary outputs of an op amp and low-pass filter during DC tests detects only 81% and 40% of all catastrophic faults, respectively. Soma reached a similar conclusion in his study of catastrophic fault coverage of DC parametric tests on amplifiers, in which he reported coverage of less than 80% [125].

Another DC test generation technique for detecting catastrophic failures was presented by Devarayanadurg in [27]. The algorithm first finds those values of the process parameters which will cause the faulty and good circuits to behave as close to each other as possible, and then finds the corresponding input vector which will detect the fault for this worst case. Thus the test generation problem is formulated as a minimax optimization problem and solved iteratively as successive linear programming problems. An analytical fault modeling technique based on manufacturing defect statistics is used to derive the fault list for the test generation. Devarayanadurg extended the method to AC tests in [28], in which he determines the time points of a transient analysis at which circuits should be compared to maximize difference between the faulty and non-faulty circuits.

Bernier presented a comparative analysis of neural networks, simulated annealing, and genetic algorithms in the determination of input patterns for testing analog circuits for catastrophic faults. In his formulation the objective is to determine a test signal that maximizes the quadratic difference between the nominal response and the faulty one due to a defect in the circuit [8].

While techniques based on catastrophic faults dominate research in the testing of digital circuits, they are of only limited use when testing analog circuits. Analog circuits must be tested for parametric faults, and the tests for parametric faults will almost always detect any catastrophic faults as well.

2.3.4 Parametric Fault Coverage

Parametric faults are caused by statistical fluctuations in the manufacturing environment. Since these statistical fluctuations can cause violations of the circuit specifications, parametric faults are just as important as catastrophic faults; they are, however, much more difficult to detect. Milor presented algorithms for detecting parametric faults in [76]. Her approach is based on setting upper and lower bounds on the permissible value of each parameter in the design and then testing the circuit to determine that no parameter falls outside of its acceptable range. In [19] Chao and Milor generalized this approach to include behavioral models and both catastrophic and parametric faults.

For circuits with a linear input/output relationship, several researchers have published interesting approaches. Tsai formulated the problem of detecting parametric faults as a quadratic programming problem [142]. Nagi developed DRAFTS (DiscRetized Analog circuit FaulT Simulator), which is an efficient AC fault simulator for linear analog circuits. Her approach maps good and faulty circuits to the discrete Z-domain, and then uses a search technique in the frequency domain to determine test frequencies for a given set of faults. At every chosen input frequency, simulations are performed to determine whether that frequency could be used as a test for a fault, until all faults have been covered [85]. Nagi uses inductive fault analysis (IFA) [116] to generate the faults to be considered, and only specific discretized values of parametric faults are considered. Balivada published a similar approach based on time-domain measurements and pole-zero analyses. To determine a test, he chooses the input which maximizes the error for each fault. After determining a test, he performs fault simulations on the remaining set of faults and eliminates the detected faults from the fault list. Balivada claims that his approach does not suffer from the error introduced by the Laplace to Z-domain transformation of the Nagi method [4]. The main shortcomings of these approaches seem to be the single-fault model assumption and the failure to address the issue of manufacturing tolerances of the non-faulty parameters.

Slamani proposed a fault-based testing approach for parametric faults which is very similar to the sensitivity-based fault diagnosis algorithms. His approach uses sensitivity analysis to solve for the values of the internal parameters [119, 120]. In the case of linearly dependent parameters, one must add additional test points to make the sensitivity matrix full rank. In [121] Slamani used sensitivities to study fault masking, fault dominance, fault equivalence, and non-observable faults. For each fault he picks the measurement which

maximizes sensitivity of the output node with respect to the candidate fault.

Hamida presented a similar sensitivity-based approach. He uses sensitivity analysis to formulate a flow problem which is solved with linear programming to deduce which parameters should be measured, for single faults [43] and for multiple faults [44, 45]. The approach can be viewed as fault diagnosis for only a limited number of faults. The algorithm finds adequate tests for detecting catastrophic and parametric faults, but performs no optimization to find the best tests. Furthermore, the method is complicated and exponential in CPU time complexity for the general case of testing for an arbitrary number of simultaneous faults.

An algebraic approach to test generation for linear analog circuits was presented in [71]. The method is based on frequency domain analysis and expressing the input/output transfer function in a sum-of-products form. The faults considered are single abnormal value changes of elements, e.g. resistors, capacitors, and inductors. The effects of manufacturing tolerances are considered in test generation, and a procedure is proposed to determine the output ranges for acceptance or rejection. The method indicates which elements in the circuit are hard to test. Another fault-based multifrequency test generation and fault diagnosis procedure for linear circuits was proposed by Mir in [80]. This procedure selects a minimal set of test measures and generates the minimal set of frequency tests which guarantee maximum fault coverage and maximal fault diagnosis. Mir chooses several self-testable linear analog circuits as examples.

2.3.5 Ordering of Tests

In the area of test ordering, Milor described an algorithm for minimizing average test time by ordering the tests in such a way that those which are most likely to detect faults are performed first [77]. Given a statistical description of the fabrication process, the algorithm minimizes the testing time required to verify all of the circuit specifications. This problem reduces to finding the best choice and order of the specification tests such that all of the faults in the model are tested.

The only weakness in this approach is the requirement for a statistical description of the parameters of the design, which may be difficult to obtain when the parameters of the design are behavioral model parameters rather than SPICE-level parameters.

2.3.6 Types of Input Stimuli

Circuits are tested using various types of input stimuli, the most common being simple DC voltages for detecting DC faults and sinusoids for detecting AC faults.

At least three researchers have studied time-domain techniques. Chin analyzed the transient response to step inputs with multivariable discriminant analysis [21]. Dai and Souders described a time-domain approach based on sensitivity analysis [25]. Taylor developed testing techniques using transient response analysis for linear sub-systems embedded within mixed-signal ICs [135].

Borrowing an effective technique from digital testing, Sloan proposed the use of random waveforms in [122]. Russell expanded on this idea in [106], presenting two new types of input stimuli:

1. Residual multiple frequency testing. This technique is derived from concurrent error detection methods employing 'information redundancy' techniques used for testing digital circuits. Two information-redundant pilot signals, whose frequencies lie just outside the operational bandwidth of the analog circuit under test, are continuously applied to the circuit, and their output values are monitored. Fluctuations in the output level of these pilot signals, which are generated on chip, indicate a fault in the circuit.

2. M-sequences. These sequences are similar to the pseudo-random binary signals used to test digital circuits. The output signature is used to determine whether or not the circuit is fault free. M-sequences are DC level, not bit sequences. A periodic pseudo-noise signal is applied to the circuit under test, and the Weiner-Hopf equation is used to estimate the impulse response of the circuit.

Finally, Schreiber and Corsi apply a sequence of pulses with varying amplitude [111, 22].

2.3.7 Power Supply Current Monitoring

Several researchers have noted that monitoring the power supply current of the circuit being tested can detect many catastrophic faults. For digital circuits this technique is known as I_{DDQ} testing [90, 40], and is based on the observation that short-circuit and open-circuit faults often dramatically impact the quiescent power supply current, especially

for CMOS logic circuitry with most transistors connected to either the power supply or the ground. Applying similar techniques to the testing of analog and mixed-signal circuits was first proposed in 1991 [6]; DC faults are detected by monitoring the quiescent power supply current. This technique was further developed in [14], in which the authors reported that for a typical circuit, 80% of the catastrophic faults produce a change in the quiescent power supply current of at least 25%. Robson demonstrated how M-sequence and current monitoring can be combined to produce a system level technique for testing mixed-signal circuits, with much of the test hardware being obtained from reconfigured digital system hardware [103].

In [95, 96] the authors proposed a method for identifying AC faults by using the spectrum of the power supply current to construct a fault dictionary. [42, 149] presented a similar approach, in which time-domain testing followed by spectral analysis of the power-supply current is used to detect both DC and AC faults.

In [82] Miura tested an A/D converter for catastrophic faults by measuring the integral of the power supply current during one clock period in which a test vector was applied.

This method is not sensitive enough to detect many parametric faults; most of the previously published works in this area claim to detect only catastrophic faults.

A much more serious problem with some of these methods stems from neglecting the fact that the power supply nodes are often designed to have a very high capacitance to ground. This capacitance severely limits the practical usability of any technique which relies upon power supply signatures rather than merely quiescent power dissipation, since the large capacitance smooths out the error signatures. The published papers which examine power supply signatures are based only upon simulation, and examining their simulation results reveals that realistic power supply capacitance was not considered.

Note that the power supply current should be considered an output of the circuit, just like any other output. Therefore all of the techniques described in this dissertation are directly applicable to monitoring power supply current. A recent study [24] measured the cross-correlation between power supply current and voltage output for a low-pass Sallen-Key filter. They concluded that power supply current is more sensitive for detecting faults in MOS transistors and output voltage is better at detecting faults in passive components. [2] concluded that the power supply current and output voltage are complementary in terms of achieving a high percentage of fault coverage with a high degree of confidence. The best

approach, therefore, is to simply consider the power supply current as an additional circuit output rather than as a special kind of analog test.

A related testing technique is to consider the power supply voltage as a controllable circuit input [1]. By varying the power supply voltages it is possible to expose faults within the circuits which are difficult to detect by conventional input voltage stimulation.

2.3.8 Summary

While considerable academic research over the past two decades has been devoted to fault-based testing methods for analog systems, these methods are not generally being used in production environments. Some of the reasons for this lack of acceptance are:

1. The methods are perceived as too complicated,

2. Some of the methods may require information that is generally not readily available, such as statistical distributions and correlations among behavioral model parameters,

3. The single-fault assumption is not considered realistic for parametric faults, and

4. Handling multiple faults by enumerating all possible fault combinations is prohibitive in memory usage and CPU time.

One of the most important problems with fault-based methods, however, is that testing for all of the faults does not generally guarantee that the performance specifications will be met. This shortcoming makes it necessary to test directly for performance anyway, so the fault-based testing becomes irrelevant or, at best, of only minor value. The reason for this problem is that a fault-free circuit is defined to be any circuit whose parameters fall within their specified bounds, and designing a circuit which meets specifications for all possible permutations of non-faulty parameter values is extremely difficult. When this is possible, the resulting design will be overly conservative and hence probably not competitive in performance. In addition, there is the problem that some circuits with parameters outside of their fault limits are perfectly functional and meet their specifications; these circuits will be wasted if discarded because of the results from a fault-based testing method.

2.4 Performance-Based Testing

The major alternative to fault-based testing is performance-based testing. Performance-based testing refers to testing techniques that verify circuit performance specifications. The most direct form of performance-based testing is the direct measurement of all circuit specifications, which is the way in which almost all analog circuits are presently tested [107]. But there are several other approaches which have been developed, including the one described in this dissertation, which are more efficient *and at least as effective* at distinguishing circuits which meet specifications from those which do not.

Souders and Stenbakken developed automatic test pattern generation algorithms for verifying performance specifications using linear models based on the sensitivity of the output with respect to the internal process/model parameters. These linear models can be derived either from simulation [132] or from manufacturing data [129] using QR-decomposition. Souders and Stenbakken select test points by using a greedy method, iteratively picking the test point to which the circuit is the most sensitive, given that the previous tests have been applied. The chosen test points are reasonable, but in no sense are they optimal. Souders and Stenbakken did not address the issue of circuits which are nonlinear with respect to their process/model parameters and they never developed a satisfactory method for dealing with ambiguity groups formed by linearly dependent parameters.

Leenaerts calculates the mathematical relation between DC parameters and performances, then uses DC measurements to verify performance intervals [64]. His approach looks promising, but the algorithm is only illustrated with one parameter, measurement noise and modeling errors are not considered, and he does not propose a method for automatic test pattern generation.

Fares proposed an analog testing method which uses fuzzy optimization models to determine whether a circuit is good or bad [32]. For example, measurements which are close to their specifications are treated with more skepticism than measurements which clearly exceed their specifications. Fares raises the issue of "error" or uncertainty in testing, which is an important issue that few other researchers have addressed; in this dissertation "error" is dealt with as a statistical phenomenon.

Lindermeir recently proposed an interesting approach based on characteristic observation inference (COI) [66]. In many situations it is prohibitive to directly verify the circuit specifications due to the test equipment costs. This approach considers a set of

reasonable input stimuli and measurements that can be performed with lower-cost test equipment. From this set a minimal number of measurements is automatically selected that represent a set of observations characterizing the state of the circuit under test with respect to parametric faults. For each given circuit specification, a corresponding test inference criterion is computed, based on logistic discrimination analysis. By applying these criteria, the satisfaction or violation of the given circuit specifications can be inferred from the observations of the circuit under test. They applied the COI method to a complex operational amplifier and found encouraging simulated results with respect to parametric faults as well as to catastrophic faults. Their main contribution, the idea of verifying performance specifications by doing simple measurements instead of direct measurements, is a valuable insight.

2.5 Design for Testability (DFT)

Design for testability encompasses a wide range of techniques which involve modifying circuits to reduce their testing cost. A useful survey of the most common design for testability techniques for analog and mixed-signal circuits can be found in [156].

Rijsinge proposed a statistical approach; he evaluates the number of test vectors required to measure a given parameter with a specified accuracy [144]. The technique is geared more toward fault diagnosis than production testing, but the testing applications are evident.

Most researchers in this area have attempted to develop circuitry which permits the various analog sub-blocks to be individually controlled and observed in isolation, e.g. [56]. This can be accomplished by signal multiplexing [147] or the use of MOS switches to isolate filter stages [123].

For op-amp-based modules, Renovell proposed some specific circuit modifications that can be used to bring controllability and observability to the frontier of each embedded module by creating transparent paths between external and internal I/Os. The key point of this transformation is to permit each analog stage to have a test mode for which it is converted into a follower stage. Adaptive solutions are proposed depending on the availability of on-chip digital resources. The testability cost is shown to be very low in terms of additional circuitry, number of extra pins, analog response penalty, and test management. Bratt proposed a similar architecture in which control voltages are injected into an operational

amplifier with a configurable internal architecture. Bandwidth performance loss is minimal and area overhead is approximately 5% for each modified operational amplifier [11].

In general, DFT schemes based on some kind of a structural division of the circuit have been largely unsuccessful because of their impact on the circuit performance [107]. Analog scan and analog built-in-self-test are efforts to make analog circuits more testable without sacrificing performance.

The main use for the techniques described in this dissertation in design for testability lies in estimating the number of tests which must be applied to fully verify circuit functionality. Given several candidate circuit architectures, it is straightforward to run the automatic test pattern generation algorithms described in Chapters 3 and 4 on each candidate architecture and compare the number of test vectors generated for each.

2.5.1 Analog Scan

The idea behind analog scan is that a large number of signal storage cells can be chained together and used to shift in all of the input signals and shift out all of the output signals for each internal analog block, thereby providing full DC control and observability over the internal analog modules while using only a few extra pins. Scan techniques are used extensively in the testing of digital circuits.

Fasang proposed a partial scan architecture for mixed-signal circuits in 1988 [34, 33]. He uses scan methods for the digital sections and a special arrangement of multiplexors and additional test points for the analog blocks.

Signal storage cells for a fully analog scan technique have been presented by Wey, both for voltages [150] and for currents [151]. These cells allow voltages and currents to be shifted into and out from internal nodes, but use a large area.

Soma developed analog scan cells for several specific circuit architectures. In [123] he presented an analog scan technique for active analog filters. In [127] he presented a technique for switched-capacitor filters. And in [126] he presented a general current-based analog scan cell. The general cell uses current to represent the analog signal to be scanned and otherwise functions the same as a digital scan cell.

A new mixed-mode boundary scan architecture was presented by Lee in [63]. The digital part of this architecture complies with the IEEE Standard 1149.1. For the analog part, Lee proposed a new boundary scan cell design and defined four analog test instructions.

2.5.2 Built-In Self-Test (BIST)

Although the difficulty of testing microelectronics products has increased, the cost of embedding testability enhancements has decreased [13]. This trend explains the increasing interest in built-in self-test. For BIST techniques, a chip contains extra circuitry which enables it to test itself.

As an example, consider the fault dictionary approach to testing. Hatzopoulos published a method for performing on-chip fault diagnosis using a "healthy signature dictionary" which is prestored in an EEPROM chip [50]. The nodes of excitation, the test points, and the sequence of voltage or current measurements are predefined. A "self-test" of the circuit "passes" when the measurements agree with the corresponding prestored signatures, within certain tolerance bounds.

Most research on analog BIST has been directed at specific circuits. Toner published extensively on using BIST for A/D converter testing [137, 138, 139, 140, 141]. He developed on-chip methods to automatically verify frequency response, signal-to-noise ratio, gain tracking, inter-modulation distortion, and harmonic distortion using 8.6 mm^2 of silicon on a BiCMOS 0.8 μm process.

Najad presented comprehensive approaches for on-chip measurements of passive components [88]. The method relies upon precision reference components, which must be located off-chip. Olbrich designed a switched-current memory cell with BIST which achieves 95% coverage for shorts and 60% coverage for open circuits [93]. Chatterjee proposed a low-cost BIST technique for linear analog circuits using DC checksum codes [20].

Mir et al. published BIST techniques for fully differential circuits in [79] and a review of general analog BIST techniques in [81]. Finally, Lopresti wrote about some experimental general analog BIST techniques in [69].

With the continuously decreasing cost of silicon and the increasing cost of mixed-signal testers, BIST is certain to become an important way of testing analog circuits. The performance impact and area impact of the BIST circuitry must be minimal, however, for the technique to be acceptable to designers. Furthermore, most BIST techniques developed to date are very circuit-specific and require extensive designer involvement.

Note that BIST does not eliminate the need for production testing; at the very least, a circuit must be powered on, the self-test sequence must be initiated, and an output signal must be observed to determine whether or not the self-test was successful.

Additionally, note that the analog testing techniques described in this dissertation do not depend upon whether the testing circuitry is on-chip or off-chip, so the techniques can be used either with or without BIST.

2.6 Conclusion

The testing methods proposed in the subsequent chapters of this dissertation have important advantages over the previous work described in this chapter. The methods described in this dissertation are performance-based, so circuit specifications are conclusively verified. In addition, ours is one of the first *statistical* approaches to the analog testing problem. The statistical methods allow us to:

1. Consider all possible parametric and catastrophic faults in all possible combinations,
2. Handle large circuits by using behavioral models, without knowing the statistical distributions of the behavioral model parameters,
3. Account for manufacturing tolerances,
4. Account for measurement noise and modeling errors,
5. Automatically generate optimal test vectors,
6. Perform statistical hypothesis testing, and
7. Handle circuits that are nonlinear in their input/output relationship and nonlinear with respect to their parameters.

All of the above can be accomplished with very reasonable memory and CPU time requirements.

Chapter 3

Automatic Test Pattern Generation for Linear Circuits

This chapter describes a new algorithm for automatic test pattern generation (ATPG) for a general class of analog systems, namely those circuits which can be efficiently modeled as a *linear* combination of statistical parameters. The algorithm is based on the statistical technique of I-optimal experimental design, in which test vectors are chosen to be maximally independent so that circuit performance will be characterized as accurately as possible in the presence of measurement noise and model inaccuracies. This technique allows analog systems to be characterized more accurately and more efficiently, thereby significantly reducing system test time and hence total manufacturing cost. Since testing currently accounts for approximately 30% of total manufacturing cost, these algorithms are expected to make a significant impact on the overall cost of designing and manufacturing analog systems.

3.1 Introduction

In this chapter we present an algorithm for deriving a minimal set of test vectors for fully testing the performance specifications of a general class of analog systems. The class of systems to which the algorithm can be applied are those systems which can be modeled in a linear function space, i.e. the system output function must be a *linear* combination of

the relevant statistical parameters. Mathematically, this model is formulated as

$$Y = \beta_0 g_0(x_0, x_1, \ldots) + \beta_1 g_1(x_0, x_1, \ldots) + \ldots + \beta_{p-1} g_{p-1}(x_0, x_1, \ldots) + \epsilon \qquad (3.1)$$

where Y is the system output vector, $\{g_i\}$ is a set of arbitrary user-specified basis vectors that are functions of the system inputs $\{x_0, x_1, \ldots\}$, β_i is the coefficient of the i^{th} basis vector, and ϵ is an error term. Many analog systems can be accurately modeled in this fashion. Note that the $\{g_i\}$ basis functions themselves do not have to be linear.

The algorithm is based upon the statistical theory of optimal experimental design, in which test vectors are chosen to be maximally independent so that the system performance Y will be characterized as accurately as possible in the presence of measurement noise and model inaccuracies. More specifically, we wish to choose the test vectors to minimize the average standard error of the predicted output, thereby maximizing the likelihood that we will be able to conclusively verify that the performance specifications have or have not been met after a minimum number of test vectors. If the minimum number of test vectors is not sufficient to conclusively verify the performance specifications, then additional test vectors are selected and applied, one at a time, until the standard error of the predicted output is low enough to verify the performance specifications. Linear regression is used to analyze the results of the tests and compute the required standard errors.

This chapter is organized as follows. Section 3.2 presents a simple example to illustrate some of the relevant basic concepts and motivate this research. Section 3.3 presents our algorithm for selecting the optimal test set and analyzing the test results. Section 3.4 discusses system modeling issues and, in particular, the selection of the $\{g_i\}$ basis vectors. Section 3.5 presents the results of applying our algorithm to some actual linear analog systems and Section 3.6 concludes.

3.2 Motivation

Consider a simple system of six current sources connected in parallel, as shown in Figure 3.1. Each of the current sources can be turned on or off by the controlling inputs $x_5, x_4, \ldots, x_0$. The sources were designed to each output one unit of current when on, but may actually output slightly more or slightly less than that amount due to manufacturing nonidealities.

Suppose one wishes to test this system to insure that the amount of current each

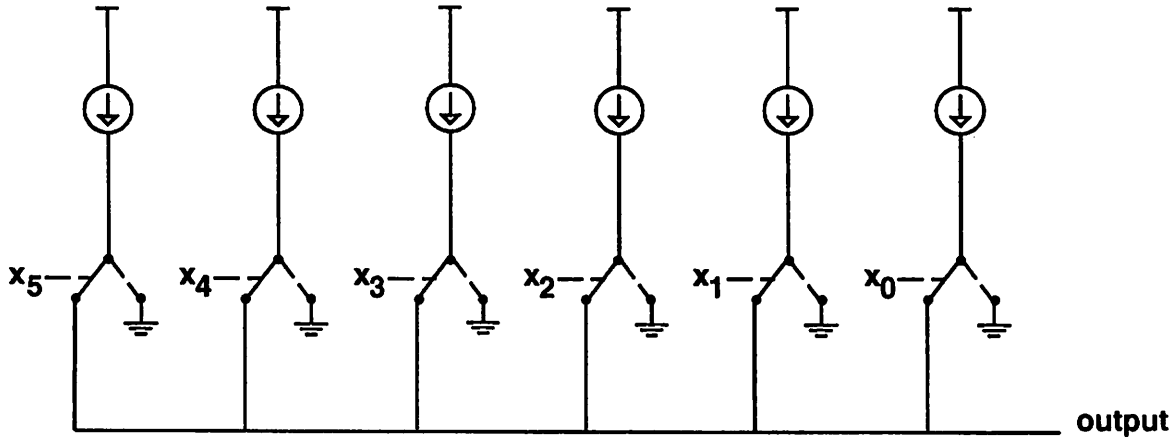


Figure 3.1: Six switchable current sources connected in parallel.

Test Set 1						Test Set 2					
x_5	x_4	x_3	x_2	x_1	x_0	x_5	x_4	x_3	x_2	x_1	x_0
1	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	1	0	0	1	0
0	0	1	0	0	0	0	1	1	0	0	1
0	0	0	1	0	0	0	1	1	1	0	0
0	0	0	0	1	0	1	1	0	0	0	0
0	0	0	0	0	1	1	1	1	0	1	0

Table 3.1: Two sets of test vectors for testing current sources.

source outputs is within 1% of its nominal value. There are six independent current sources which must be measured, so at least six test vectors must be applied. Furthermore, suppose that the ammeter used to measure the current at the output is known to be accurate to within 0.5% of the nominal current.

The simplest set of test vectors that can be imagined is probably Test Set 1 in Table 3.1, which tests each current element in turn by setting one of the x_i's to 1 while leaving the others at 0.

An alternative set of possible test vectors, which also happens to be a provably optimal set of test vectors for this circuit, is shown as Test Set 2 in Table 3.1. Both sets of test vectors can be used to estimate the actual current that each source outputs. The two test sets differ, however, in the accuracy with which they can make this measurement.

Figure 3.2 shows the 99% confidence intervals which could be constructed after applying each set of test vectors to a randomly-generated deterministic system. For Test Set 1 it can be concluded (with > 99% confidence) that the x_4 and x_3 current sources fall within their specifications, but it is not possible to draw any definite conclusions about the other current sources. For Test Set 2, on the other hand, the confidence intervals are much tighter and it can be concluded (with > 99% confidence) that the x_5 and x_0 current sources fall outside their specifications and that the x_4, x_3, and x_2 current sources fall within their specifications. Thus the simple test set leaves four estimates uncertain while the best test set leaves only one estimate uncertain.

A common technique for tightening the confidence intervals is to repeat each test vector several times and then average the results. Each test vector would have to be applied five times, however, for a total of 30 tests, to obtain the same confidence intervals that can be found from one application of the six vectors in the best test set.

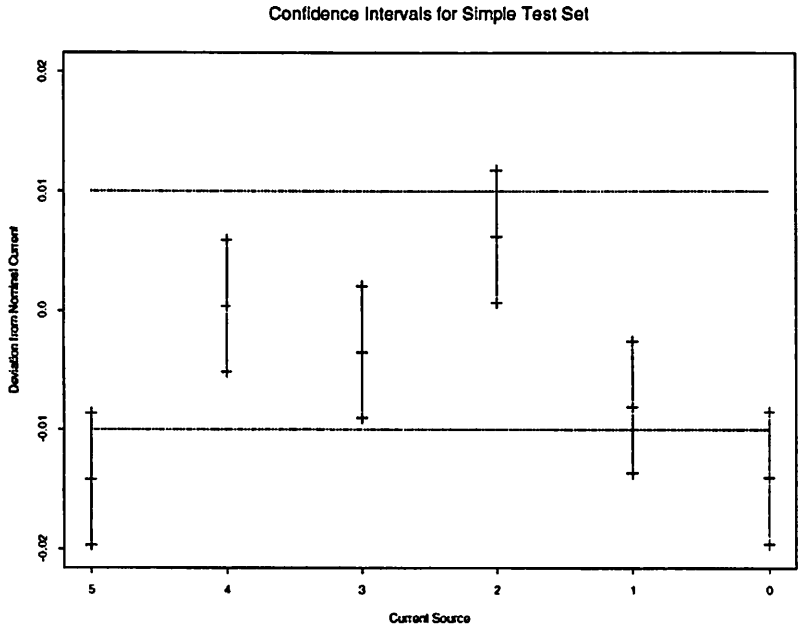
From this simple example it is clear that choosing a "good" set of test vectors is desirable because it will lead to more accurate characterizations of system output, and hence possibly smaller test sets. Good test sets are not intuitively obvious, however, even for very simple systems. In the remainder of the chapter we discuss some new ATPG techniques which have been developed and implemented for automatically finding these good test sets for linear systems such as these current sources.

In this chapter we assume that a suitable linear behavioral model for the system exists (or can be easily derived from sensitivity analysis), and we focus on a new optimization algorithm for selecting the best set of test vectors.

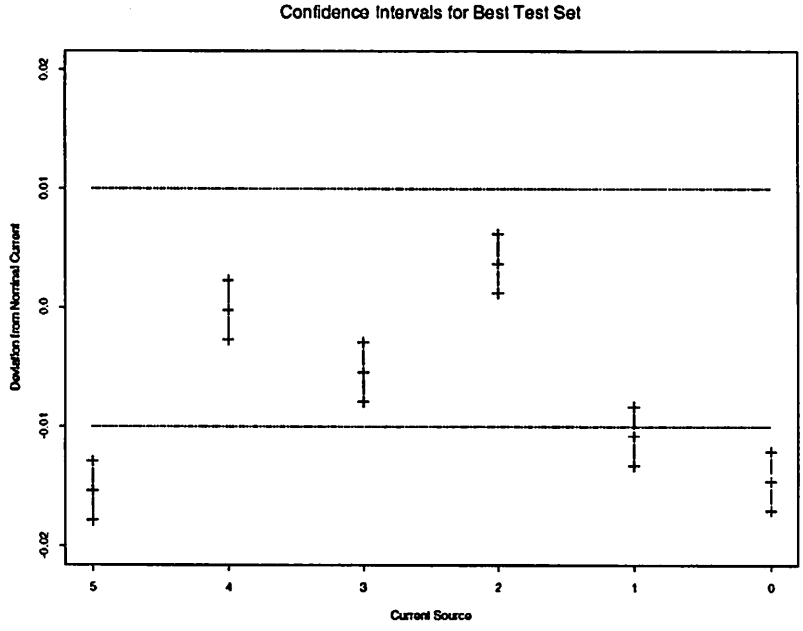
Because of measurement noise and modeling inaccuracies, the system output predicted from a finite number of measurements will never match the actual system output precisely. Thus we propose the use of statistical *confidence intervals* to verify system performance in the presence of measurement noise and model inaccuracies, with which we can guarantee (with 99% probability) that the actual system output falls within the confidence interval. The test points are chosen to make these confidence intervals as tight as possible.

3.3 Algorithm

The goal of testing is to determine whether a circuit meets its specifications. Given the behavioral model shown in Equation 3.1, we can determine whether the specifications



(a)



(b)

Figure 3.2: (a) Confidence intervals from applying Test Set 1. (b) Confidence intervals from applying Test Set 2.

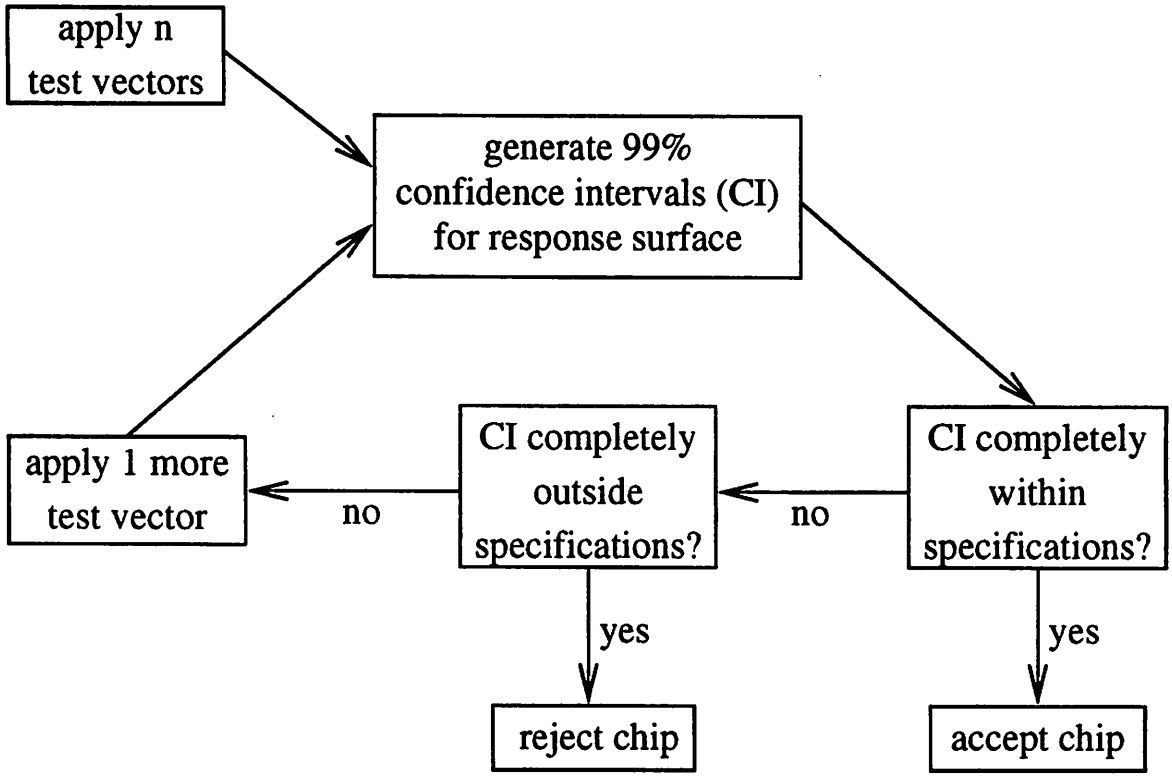


Figure 3.3: Proposed testing algorithm.

are met by estimating the system response Y over all inputs x. To minimize cost, we wish to use as few test vectors as possible to estimate Y. Because of inevitable measurement noise and modeling inaccuracies, there will always be uncertainty associated with our estimation of Y; we can reduce this uncertainty by choosing "good" test vectors and/or by applying more test vectors. Note, however, that if n is the number of independent behavioral model parameters $\{\beta_i\}$, then at least n test vectors must be applied in order to fully characterize Y. If fewer than n input vectors are applied, then at least one dimension of the linear function space remains unexplored and hence the output function is unconstrained in that dimension.

With these factors in mind, the testing algorithm that we propose is shown in Figure 3.3.

The choice of test vectors is a difficult optimization problem. The objective is to minimize the standard error of the estimated response function, which is a function of the choice of test vectors. Intuitively, the orthogonality of the test vectors is measured by the

degree to which each test vector maximizes the contribution of one basis function while minimizing the contribution of the others.

The algorithm used to derive the maximally orthogonal test vectors is:

1. Eliminate any redundant basis vectors.
2. Run the I-optimality algorithm to select the best n tests, where n is the dimensionality of the function space after eliminating redundant basis vectors.
3. Run the I-optimality algorithm to select the best additional vectors, one at a time, for use if the prior tests are not conclusive.

3.3.1 Optimality Criteria

There are several different optimality criteria (A-, D-, E-, G-, and I-), the relative merits of which have been debated extensively in the relevant literature [60, 10]. D-optimality, which is generally considered to be the simplest type of optimality, minimizes the average prediction variance of the model coefficients. This type of optimality would be very suitable for fault diagnosis, in which we wish to estimate the actual values of each circuit component as accurately as possible. D-optimality claims nothing about the average prediction variance of the system output, however, so it is not the best choice for verifying that the system output meets its specifications.

The two types of optimality which do consider the prediction variance of the system output are G- and I-optimality. G-optimality minimizes the *maximum* prediction variance over the response surface of interest. It would probably be the most suitable for verifying that a circuit meets its specifications, since specifications are frequently stated as worst-case bounds. G-optimality is difficult to optimize upon, however, because it is not continuously differentiable. Hence I-optimality, which is continuously differentiable, was chosen for this research. I-optimality minimizes the *average* prediction variance over the response surface of interest. To formulate these ideas mathematically, let

$$y = \beta_0 g_0 + \beta_1 g_1 + \beta_2 g_2 + \beta_3 g_3 + \ldots + \beta_{p-1} g_{p-1} + \epsilon \tag{3.2}$$

where y is the response variable, $\{g_i\}$ are the p independent basis vectors, β_i is the coefficient of the i^{th} basis vector, and ϵ represents the measurement and modeling errors, which are assumed to be independent with mean 0 and variance σ^2.

Let X be the design matrix, which contains one row for each of the n test vectors.

$$X = \begin{bmatrix} g_0(x_0) & g_1(x_0) & g_2(x_0) & \cdots & g_{p-1}(x_0) \\ g_0(x_1) & g_1(x_1) & g_2(x_1) & \cdots & g_{p-1}(x_1) \\ & & \vdots & & \\ g_0(x_{n-1}) & g_1(x_{n-1}) & g_2(x_{n-1}) & \cdots & g_{p-1}(x_{n-1}) \end{bmatrix} \tag{3.3}$$

The design moment matrix M_X can be calculated as

$$M_X = \frac{1}{n} X'X \tag{3.4}$$

and the prediction variance at an arbitrary point x on the response surface is

$$var\ \hat{y}(x) = \frac{\sigma^2}{n} f(x) M_X^{-1} f(x)' \tag{3.5}$$

where

$$f(x) = \begin{bmatrix} g_0(x) & g_1(x) & g_2(x) & \cdots & g_{p-1}(x) \end{bmatrix} \tag{3.6}$$

for each point x on the response surface R. An I-optimal design is one which minimizes the normalized average of $var\ \hat{y}(x)$ over R,

$$I = \frac{n}{\sigma^2} \int_R var\ \hat{y}(x) d\mu(x). \tag{3.7}$$

This integral simplifies [10] to give

$$I = trace\left\{ M M_x^{-1} \right\} \tag{3.8}$$

where M is the moment matrix of R,

$$M = \int_R f(x)' f(x) d\mu(x). \tag{3.9}$$

3.3.2 Optimization

Finding an exactly I-optimal design is believed to be NP-complete [23] and hence only feasible for very small problems. For larger problems, several heuristic algorithms have been successfully used to find "good" solutions to this and other related problems in the area of optimal experimental design. These heuristic algorithms include simulated annealing [23], greedy swap techniques [89], and gradient descent techniques. For this research we used the gradient descent techniques implemented in the software package

GOSSET, which was recently developed by Hardin and Sloane at AT&T Bell Laboratories [48]. The primary focus of GOSSET is low-order polynomial models, which are of only limited use in characterizing typical analog circuits. For this research, therefore, GOSSET was extended to utilize arbitrary Lipschitz continuous functions, such as the piecewise linear output of common behavioral simulators [68] and SPICE [146].

In applying each of the algorithms to the analog testing problem, we found that the usefulness of the greedy swap and simulated annealing techniques seems to be restricted to low dimensions and small numbers of design points (maximum of 20-30), making those algorithms unsuitable for the general analog testing problem. The gradient descent techniques, on the other hand, work reasonably well on all sizes of problems. The primary limitation of the gradient-based techniques is that the basis vectors must be differentiable; this requirement is usually satisfied in the analog testing problem, so a gradient descent method was selected.

The problem of finding the best set of test vectors can be understood intuitively as follows. We want the test vectors to be as orthogonal as possible, in the sense that they are widely separated from each other in the space defined by the coefficients of the basis vectors. After choosing an initial set of random test vectors, the direction in which each test point should be moved to be further away from the other test points can be calculated. Each test point can be perturbed in the direction of this gradient and the design will have been improved.

GOSSET uses an optimization algorithm known as Hooke and Jeeves pattern search [54], which is based on the idea of finding a "valley" and following it downward until reaching the lowest point on the response surface, similar to the manner in which a stream flows down a mountain. The optimization begins by selecting a random point on the response surface, calculating the gradient at that point, and proposing a set of small perturbations in the direction of the gradient. If this set of perturbations causes the objective function to improve, then this "move" is accepted and the step size is increased by a constant factor. Otherwise the set of perturbations is rejected and a smaller move is attempted.

The initial point in the search space, $x^{(0)}$, is chosen randomly. The initial velocity vector $v^{(0)}$ is set to 0, where the velocity v_i of input i is defined as being the amount by which that input is perturbed in a given move. The step size s is set to a small value. The search then proceeds as

$$x^{(i+1)} = x^{(i)} + v^{(i+1)} \tag{3.10}$$

$$v^{(i+1)} = v^{(i)} + sg(x^{(i)}) \tag{3.11}$$

where $g(x^{(i)})$ is the gradient evaluated at the point $x^{(i)}$. If

$$F(x^{(i+1)}) < F(x^{(i)}), \tag{3.12}$$

where F is the objective function, then the value for $x^{(i+1)}$ is accepted, s is multiplied by 1.04, and the iteration is repeated. If $F(x^{(i+1)}) \not< F(x^{(i)})$ then $v^{(i)}$ is set to 0 and (3.10) and (3.11) are tried again. If there is still no reduction in F, then s is divided by 2 and (3.10) and (3.11) are tried again. The algorithm terminates when the step size is less than some small accuracy limit. Then, if desired, a new random starting point can be chosen and the entire minimization algorithm repeated, successively, until a specified number of random starts have been investigated. At that point the algorithm terminates, returning the best design found.

If $x^{(i)}$ moves outside the feasibility region, which is defined by the limited range of values that each input can assume, then it is moved to the closest feasible point.

Note that the optimization assumes that all of the inputs to the system are continuous. If the inputs are discrete, as frequently occurs when analog systems are connected to digital systems, then a post-processing step is performed which is similar to integer programming. Each of the test vectors is sequentially considered, and discrete inputs with illegal values are converted to whichever of the two closest discrete values gives the smallest value of F. The technique is essentially *greedy* integer programming, since the order in which the inputs are considered could cause the algorithm to become stuck at a local minimum. We have empirically observed that the algorithm works well because:

1. The optimization pushes many variables to their boundaries, which are usually legal discrete values, and

2. The [usually slight] non-optimality introduced by the rounding off of one test vector can frequently be partially compensated for by the rounding off of a similar test vector in the opposite direction.

Hooke and Jeeves found empirically, in a curve-fitting problem involving a neutron reactor, that the computation time for their pattern search algorithm increased only *linearly* with the number of variables, which makes it especially suitable for the analog testing problem because analog systems may require large numbers of parameters to accurately characterize them.

3.4 Model Derivation

The statistical design and analysis techniques which we use for system testing require a *homoskedastic, linear* function space. The linearity requirement means that any system output can be expressed as an additive combination of a set of basis vectors, as shown in Equation 3.1. Homoskedastic refers to a requirement that the measurement error ϵ, which is a combination of model inaccuracies and noise, is not a function of the input; this assumption is reasonable for many typical analog testing situations.

There are several simple methods which can be used to choose the $\{g_i\}$ basis vectors. For extremely simple systems the basis functions may be obvious from a simple description of the expected output. Consider, for example, the current sources discussed in Section 3.2; the output is modeled as

$$Y = \beta_5 x_5 + \beta_4 x_4 + \beta_3 x_3 + \beta_2 x_2 + \beta_1 x_1 + \beta_0 x_0 + \epsilon \tag{3.13}$$

where the β coefficients are the unknown model parameters we wish to characterize. The basis functions for this system are simply $\{x_5, x_4, x_3, x_2, x_1, x_0\}$, the set of contributions from each current source, which are summed together to form the output.

For more complicated systems, the Taylor expansion can be used to derive a very useful additive model

$$f(a+x) = f(a) + xf'(a) + \frac{x^2 f''(a)}{2!} + \ldots + \frac{x^{n-1} f^{(n-1)}(a)}{(n-1)!} \tag{3.14}$$

where a represents the nominal value of a model parameter, $f(a)$ represents the value of the output when that model parameter is at its nominal value, and x represents the amount by which that model parameter deviates from its nominal value because of manufacturing nonidealities. We wish to estimate $f(a+x)$.

A first-order Taylor series approximation is a reasonably accurate model for many common analog systems with parameters that do not deviate significantly from their nominal values. This is the model used by Stenbakken and Souders [134], and our discussion of it here will be brief. Dropping the higher-order terms and generalizing to multiple dimensions, the expansion becomes

$$f(a+x) = f(a) + \nabla f(a) x \tag{3.15}$$

$$= f(a) + \frac{\partial f}{\partial a_1} x_1 + \frac{\partial f}{\partial a_2} x_2 + \ldots \tag{3.16}$$

where a_i is the nominal value of the i^{th} model parameter and x_i is the deviation in that parameter. The basis functions for this system are thus $\{f(a), \frac{\partial f}{\partial a_1}, \frac{\partial f}{\partial a_2}, \ldots\}$. $f(a)$ is the nominal system performance, and each of the partial derivatives represents an error signature for a particular type of manufacturing defect which can occur. The error signatures are computed by finding the sensitivity of the output to the parameters of interest at each point on the response surface. Note that these error signatures could represent either catastrophic faults, such as shorts and disconnections, or parametric faults, such as small deviations in capacitance values or process parameters.

3.4.1 Eliminating Ambiguity Groups

Once the basis vectors $\{g_1, g_2, \ldots, g_n\}$ are identified, we compute their null space to verify that they are all independent. The parameters associated with error signatures that are linearly dependent are said to belong to the same *ambiguity group*, since variations in those parameters are indistinguishable at the system output. Ambiguity groups reduce the number of basis vectors needed to model the response surface and hence the number of test vectors which must be applied to fully characterize a system.

Let U be the matrix formed from these basis vectors, where g_i is the i_{th} column of U. Suppose U has dependent columns, then its null space is non-empty such that

$$UN = 0 \tag{3.17}$$

where $N \in R^{m \times r}$ is a matrix with r independent column vectors that spans the null space of U. Non-zero entries in N indicate that the corresponding components are in ambiguity groups. A component i belongs to an ambiguity group if and only if row i of N has a non-zero entry. Furthermore, we have the following theorem.

Theorem 3.4.1 *Components i and j are in the same ambiguity group if rows i and j of N are non-zero and not orthogonal to each other [67].*

Proof. Suppose components i and j are not in the same ambiguity group and not orthogonal. Then, because they are not in the same group, there exists an orthonormal matrix $M \in R^{m \times m}$ such that if any entry in N_iM is non-zero, then the corresponding entry in N_jM is zero, or vice-versa, where $N_i \in R^{1 \times m}$ and $N_j \in R^{1 \times m}$ are rows i and j of N. It follows that

$$(N_iM)(N_jM)' = N_iMM'N_j' = 0 \tag{3.18}$$

which implies that

$$N_i N_j' = 0 \tag{3.19}$$

since $MM' = I$. Thus, N_i and N_j are orthogonal, which results in contradiction and completes the proof. From Theorem 3.4.1, it follows that the components fall into the same group if their corresponding row vectors of N are non-zero and not orthogonal. In other words, we have the following corollary:

Corollary 3.4.1 *The number of ambiguity groups is equal to the number of orthogonal subspaces spanned by the rows of N.*

The null space of U can be computed using singular value decomposition (SVD) or Gaussian elimination. In the case of SVD, we first compute $U'U$, followed by SVD

$$U'U = X_1 X_2 N' \tag{3.20}$$

where N spans the null space of $U'U$. Since $U'UN = 0$, $UN = 0$, so N is the null space of U also. The reason for computing $U'U$ in (3.20) is that U often has many more rows than columns, so computing N for a smaller matrix $U'U$ is more efficient. Furthermore, note that computing N and checking the rows of N for pairwise orthogonality can be performed in polynomial time.

In summary, the approach for finding ambiguity groups is:

1. Given a sensitivity matrix U.
2. Find N, the null space of U, using singular value decomposition (3.20) or Gaussian elimination. Let ambiguity group number $g = 1$.
3. Remove the first non-zero row of N and assign to group number g.
4. Check if any remaining rows are orthogonal. If not, assign them to group g and remove.
5. Increment g and repeat Step 3 until all rows are removed.

To find a set of independent basis vectors for the system, we repeatedly:

1. Remove one vector from each ambiguity group.
2. Recompute the ambiguity groups.

We continue until no ambiguity groups remain. Once this set of independent basis vectors is formed, the I-optimality routines, as described in Section 3.3, are executed to find a good set of test vectors.

3.4.2 Calculating Confidence Intervals

Once the test vectors have been applied, the measured responses are used to estimate $\hat{\beta}$, the vector of coefficients for each of the basis vectors. For the special case when the number of test points is equal to the number of basis vectors, $\hat{\beta}$ is found by solving

$$X\hat{\beta} = Y \tag{3.21}$$

for $\hat{\beta}$, where X is the design matrix as output by the I-optimality routine and Y is the vector of measured responses. When the number of test points is greater than the number of basis functions, $\hat{\beta}$ is found by using linear regression, solving

$$X'X\hat{\beta} = X'Y \tag{3.22}$$

for $\hat{\beta}$,

$$\hat{\beta} = (X'X)^{-1} X'Y. \tag{3.23}$$

The variance-covariance matrix of $\hat{\beta}$, $\mathcal{D}\left[\hat{\beta}\right]$, is given by

$$\mathcal{D}\left[\hat{\beta}\right] = s^2 (X'X)^{-1} \tag{3.24}$$

where s^2, an estimator of σ^2, is given by

$$s^2 = \frac{\sum_{i=1}^{n}[Y_i - \hat{y}(x_i)]^2}{(n-p)} \tag{3.25}$$

where Y_i is the i^{th} observation, $\hat{y}(x_i)$ is the predicted value of Y_i, based on the model, n is the number of measurements, and p is the dimensionality of the model (the number of independent basis vectors). Given these values we can calculate the exact confidence interval for the entire response surface using Scheffé's method for simultaneous interval estimation [114, Chapter 5]. The Scheffé confidence interval is given by

$$CI\left(\hat{y}(x)\right) = f(x)\hat{\beta} \pm \sqrt{pF^{\alpha}_{p,n-p} f(x) \left(\mathcal{D}\left[\hat{\beta}\right]\right) f(x)'} \tag{3.26}$$

where $1-\alpha$ is the exact overall probability that the actual system response lies completely within the confidence interval. Typically, $\alpha = 1\%$, which results in 99% confidence intervals.

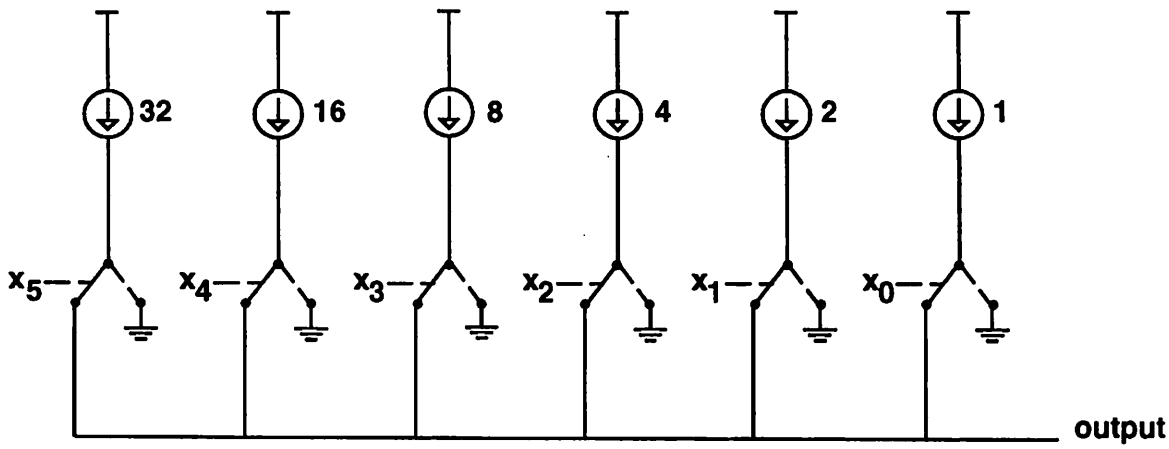


Figure 3.4: 6-bit binary-weighted current source D/A converter.

3.5 Results

In this section we describe two Nyquist-rate D/A converters on which our linear ATPG algorithms have been run. The first is a 6-bit binary converter and the second is a 10-bit interpolative D/A converter.

A 6-bit Nyquist-rate D/A converter based on binary-weighted current sources is shown in Figure 3.4, which is similar to the simple example presented in Section 3.2 except that in this case the current sources are binary-weighted instead of unit-weighted. The basis vectors for the system are chosen to be $\{1, x_5, x_4, x_3, x_2, x_1, x_0\}$, where the constant function 1 is used to model the converter offset. Since there are seven independent basis functions in the model, at least seven tests must be performed to fully characterize the system. The I-optimal design is shown in Table 3.2, along with the next seven extra points which would be chosen, in succession, to tighten the confidence intervals on the estimated performance.

Application of the seven initial test vectors to a simulated D/A converter [68]

Code	Inputs						I-Value
	x_5	x_4	x_3	x_2	x_1	x_0	
8	0	0	1	0	0	0	
15	0	0	1	1	1	1	
21	0	1	0	1	0	1	
22	0	1	0	1	1	0	
35	1	0	0	0	1	1	
44	1	0	1	1	0	0	
59	1	1	1	0	1	1	1.27778
61	1	1	1	1	0	1	1.12500
38	1	0	0	1	1	0	0.97619
1	0	0	0	0	0	1	0.83333
48	1	1	0	0	0	0	0.70000
48	1	1	0	0	0	0	0.58333
26	0	1	1	0	1	0	0.55263
10	0	0	1	0	1	0	0.52222
All 64 codes							0.10938

Table 3.2: Test vectors chosen for D/A converter.

produced the following measurements:

$$Y = \begin{bmatrix} 6.9303 \\ 13.8717 \\ 19.8016 \\ 20.8288 \\ 34.0298 \\ 43.0256 \\ 57.9037 \end{bmatrix}. \tag{3.27}$$

The design matrix X is

$$X = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}. \tag{3.28}$$

Solving for $\hat{\beta}$,

$$\hat{\beta} = (X'X)^{-1} X'Y = \begin{bmatrix} -1.0449 \\ 32.1143 \\ 15.8988 \\ 7.9751 \\ 3.9811 \\ 1.9938 \\ 0.9666 \end{bmatrix}. \tag{3.29}$$

We assume that s^2, an estimate of σ^2, has already been found by measuring all 64 codes on a previously fabricated part. $\mathcal{D}\left[\hat{\beta}\right]$ is calculated to be

$$\mathcal{D}\left[\hat{\beta}\right] = s^2 (X'X)^{-1} \tag{3.30}$$

$$= 0.00104^2 \begin{bmatrix} 1.52 & -0.37 & -0.37 & -0.63 & -0.63 & -0.30 & -0.30 \\ -0.37 & 0.74 & 0.07 & -0.07 & 0.26 & -0.07 & -0.07 \\ -0.37 & 0.07 & 0.74 & 0.26 & -0.07 & -0.07 & -0.07 \\ -0.63 & -0.07 & 0.26 & 0.74 & 0.07 & 0.07 & 0.07 \\ -0.63 & 0.26 & -0.07 & 0.07 & 0.74 & 0.07 & 0.07 \\ -0.30 & -0.07 & -0.07 & 0.07 & 0.07 & 0.74 & -0.26 \\ -0.30 & -0.07 & -0.07 & 0.07 & 0.07 & -0.26 & 0.74 \end{bmatrix} \tag{3.31}$$

The response surface R for this converter is the set of all possible input codes,

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ & & & \vdots & & & \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{3.32}$$

The 99% confidence interval is therefore

$$CI\left(\hat{y}(x)\right) = f(x)\hat{\beta} \pm \sqrt{pF^{\alpha}_{p,n-p} f(x) \left(\mathcal{D}\left[\hat{\beta}\right]\right) f(x)'} \tag{3.33}$$

$$= \begin{bmatrix} -1.0449 \\ -0.0783 \\ 0.9489 \\ 1.9155 \\ \vdots \\ 61.8848 \end{bmatrix} \pm \sqrt{(7)(2.9706)(0.00104^2) \begin{bmatrix} 1.5185 \\ 1.6667 \\ 1.6667 \\ 1.2963 \\ \vdots \\ 1.2963 \end{bmatrix}} \tag{3.34}$$

$$= \begin{bmatrix} -1.0507, & -1.0390 \\ -0.0844, & -0.0721 \\ 0.9428, & 0.9550 \\ 1.9101, & 1.9209 \\ \vdots & \\ 61.8794, & 61.8902 \end{bmatrix}, \tag{3.35}$$

where $f(x)$ is each row of R, in turn.

The INL error of the converter is the difference between the actual output and the expected output, after correcting for gain and offset errors. Applying these gain and offset corrections to the above confidence intervals, we obtain the INL confidence intervals shown in Figure 3.5.

This example illustrates how the entire performance of the D/A converter can be modeled quite accurately after the application of only seven well-chosen test vectors. Furthermore, we may be able to draw some conclusions regarding the acceptability of this D/A converter, depending upon the INL specification. If the INL specification is greater than 0.2 LSB, then the converter should be accepted with no further tests. If the INL specification is less than 0.1 LSB, then the converter should be rejected with no further tests. If the INL specification falls between these bounds, then additional test vectors must be applied to tighten the confidence intervals.

3.6 Conclusions

We have presented a new ATPG algorithm which automatically generates a minimal set of test vectors for characterizing a general class of analog circuits, namely those circuits which can be efficiently modeled as a linear combination of user-defined basis functions. The algorithm chooses the set of test vectors so as to minimize the average prediction

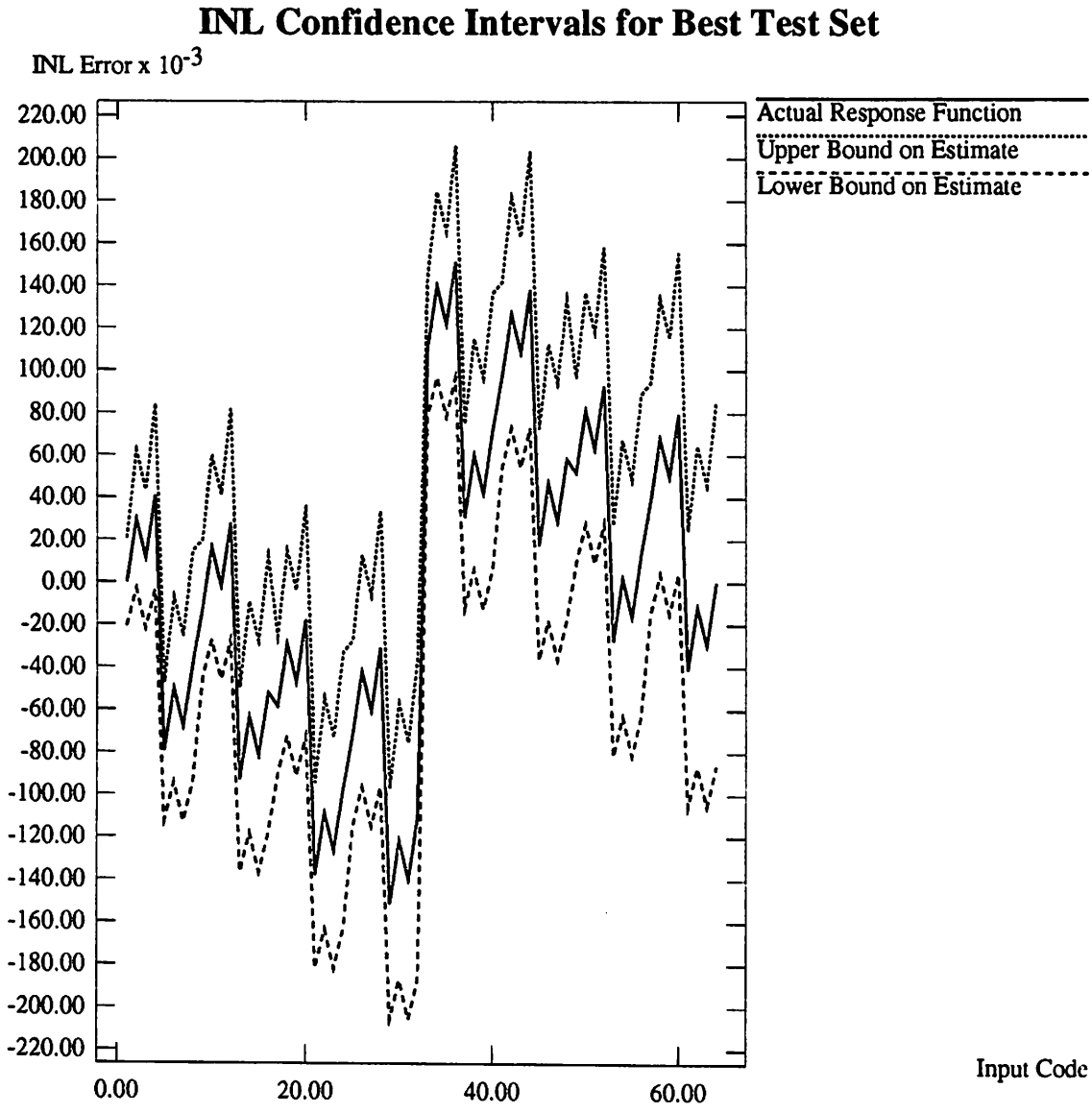


Figure 3.5: Upper and lower bounds on INL error from seven test vectors.

variance of the model. Applying the minimal set of test vectors to a circuit produces an estimate of the circuit's performance for all possible input vectors and, more importantly, confidence intervals on those estimates which can be used to determine whether the component should be passed or failed, or whether additional test vectors should be applied to tighten the confidence intervals.

Because these techniques generate the tightest possible confidence intervals after the minimum number of test vectors, they represent the most efficient way of fully characterizing system performance. Tight confidence intervals will lead to reduced testing time for analog systems because more components will be fully verifiable, to a desired confidence level, with the minimum number of test vectors. We have applied the algorithm to several analog systems and shown it to be efficient and effective.

Chapter 4

Automatic Test Pattern Generation for Nonlinear Analog Circuits

In this chapter we present a new algorithm for performing automatic test pattern generation for *nonlinear* analog systems. As in the linear case, the algorithm is based upon behavioral modeling and the statistical technique of I-optimal experimental design, in which test vectors are chosen to be maximally independent so that circuit performance will be characterized as efficiently as possible in the presence of measurement noise and model inaccuracies. This technique allows nonlinear analog systems to be characterized more accurately and more efficiently, thereby significantly reducing system test time and hence total manufacturing cost.

This algorithm can be applied to those systems that can be modeled by a behavioral model of the form

$$Y = f\left(\vec{x}, \vec{\theta}\right) + \epsilon \tag{4.1}$$

where Y is the system output(s) and f is a behavioral model which is a function of $\vec{x}$, the circuit inputs which can be controlled during testing, and $\vec{\theta}$, a set of process parameters characterizing the behavioral model, e.g. W, L, μ, V_{T0}, t_{ox}, etc. of each transistor. ϵ is an error term representing measurement noise and model inaccuracies. Many time-invariant analog systems can be accurately modeled in this fashion. Note that the linear model presented in Equation 3.1 is a special case of this more general formulation, with $\vec{\theta} = \beta$.

As for the linear case, our nonlinear analog ATPG algorithm is based upon the statistical theory of optimal experimental design, in which test vectors are chosen to be maximally independent to characterize the system performance Y as efficiently as possible in the presence of measurement noise and model inaccuracies, where we define efficiency to be the ratio of test accuracy to test cost. We choose the test vectors to minimize the average standard error of the predicted output, thereby maximizing the likelihood that we will be able to conclusively verify whether the performance specifications have been met after a minimum number of test vectors. If the minimum number of test vectors is not sufficient to conclusively verify the performance specifications, then additional test vectors are selected and applied, one at a time, until the standard error of the predicted output is low enough to verify the performance specifications. Nonlinear regression based on the modified Gauss-Newton method is used to analyze the results of the tests and compute the required standard errors.

Section 4.1 presents our algorithm for selecting the optimal test set and analyzing the test results. Section 4.2 presents the results of applying our algorithm to two nonlinear analog systems.

4.1 Algorithm

The goal of testing is to determine whether a circuit meets its specifications. Given the behavioral model shown in Equation 4.1, we can determine whether the specifications are met by estimating the system response Y over all inputs x. To minimize cost, we wish to use as few test vectors as possible to estimate Y. Because of inevitable measurement noise and modeling inaccuracies, there will always be uncertainty associated with our estimation of Y; we can reduce this uncertainty by choosing "good" test vectors and/or by applying more test vectors. Note, however, that if n is the number of independent behavioral model parameters, then at least n test vectors must be applied in order to fully characterize Y. If fewer than n test vectors are applied, then at least one dimension of the space remains unexplored and hence the output function is unconstrained in that dimension.

With these factors in mind, the general testing algorithm that we proposed in Chapter 3, Figure 3.3, can also be used for testing nonlinear circuits. As with linear circuits, the choice of test vectors is a difficult optimization problem. The objective is to minimize the standard error of the estimated response function Y, which is a function of the choice

of test vectors and the circuit parameters $\vec{\theta}$. *The test vectors are selected by linearizing f w.r.t. $\vec{\theta}$ and applying optimal experimental design algorithms to the resultant linear system.* The linearization is a Taylor series expansion based on sensitivities,

$$f\left(\vec{x},\vec{\theta}+\vec{\delta}\right) \cong f\left(\vec{x},\vec{\theta}\right)+\nabla f\left(\vec{x},\vec{\theta}\right)\vec{\delta} \tag{4.2}$$

$$\cong f\left(\vec{x},\vec{\theta}\right)+\left.\frac{\partial f\left(\vec{x},\vec{\theta}\right)}{\partial\theta_1}\right|_{\vec{\theta}}\delta_1 + \left.\frac{\partial f\left(\vec{x},\vec{\theta}\right)}{\partial\theta_2}\right|_{\vec{\theta}}\delta_2+\ldots \tag{4.3}$$

Since the values of the circuit parameters $\vec{\theta}$ are unknown prior to testing, we generate the initial n test vectors by linearizing about the *nominal* parameter values, $\vec{\theta}=\vec{\theta}_{nom}$. When additional test vectors are required for a particular circuit, they are generated by linearizing about the current estimate of the parameter values for that circuit. Note, therefore, that the initial n test vectors need only be generated once, prior to testing any circuits, but that the additional test vectors must be generated on-line for each individual circuit.

GOSSET is used in the same manner as for linear circuits, selecting the test vectors by minimizing the I-value.

4.1.1 Nonlinear regression

Given a behavioral model of the form

$$Y=f\left(\vec{x},\vec{\theta}\right)+\epsilon, \tag{4.4}$$

nonlinear regression is an optimization problem which involves choosing $\vec{\theta}$ to minimize a least squares objective function H,

$$H\left(\vec{\theta}\right)=\frac{1}{2}\sum_{k=1}^{n}\left(y_k-f\left(x_k,\vec{\theta}\right)\right)^2. \tag{4.5}$$

To perform the optimization we use a modified Gauss-Newton method with step halving [58]. The basic iteration is

$$\vec{\theta}_{j+1}=\vec{\theta}_j+\alpha\vec{\delta}^* \tag{4.6}$$

where α represents the step length, which is initially 1, and $\vec{\delta}^*$ represents the adjustments to be made to an independent set of the behavioral model parameters $\vec{\theta}$. $\vec{\theta}_0$ is set to $\vec{\theta}_{nom}$,

the nominal parameter values. $\vec{\delta}^*$ is computed by solving

$$J\left(\vec{\theta}_j\right)' J\left(\vec{\theta}_j\right) \vec{\delta}^* = J\left(\vec{\theta}_j\right)' \left(\vec{y} - f\left(\vec{x}, \vec{\theta}_j\right)\right) \tag{4.7}$$

where J represents the Jacobian of f,

$$J\left(\vec{\theta}\right)_{kj} = \frac{\partial f\left(\vec{x_k}, \vec{\theta}\right)}{\partial \theta_j}, \tag{4.8}$$

evaluated at each of the selected test points.

At each iteration, if $H_{j+1} > H_j$ then α is repeatedly halved until $H_{j+1} \leq H_j$ or α becomes less than $\frac{1}{16}$. During the course of the computations, additional test vectors are generated and applied as necessary, whenever the rank of the current Jacobian exceeds the number of test vectors which have been applied.

Equation 4.7 is solved using linear regression by robust QR-decomposition, as described in Section 4.1.2. The iteration stops when $\vec{\theta}$ converges, which is defined as

$$|\theta_{i,j} - \theta_{i-1,j}| \leq \epsilon_1 \left(|\theta_{i,j}| + \epsilon_2\right) \forall j \in \{1, \ldots, p\} \tag{4.9}$$

where $\epsilon_1 = \sqrt{\epsilon_0}$, $\epsilon_2 = 10\epsilon_1$, and ϵ_0 is related to the precision of the behavioral models being used to evaluate f and J.

Under fairly general conditions, $\hat{\theta}$ which minimizes $H\left(\vec{\theta}\right)$ will be distributed $\mathcal{N}\left(\vec{\theta}, \Sigma\right)$, where

$$\Sigma = \sigma^2 \left[J\left(\vec{\theta}\right)' J\left(\vec{\theta}\right)\right]^{-1}. \tag{4.10}$$

Since we have already computed the QR-decomposition of J, Σ can be easily computed by noting that

$$\begin{aligned} \Sigma &= \sigma^2 \left[(QR)'(QR)\right]^{-1} && (4.11)\\ &= \sigma^2 \left[R'Q'QR\right]^{-1} && (4.12)\\ &= \sigma^2 \left[R'R\right]^{-1} && (4.13)\\ &= \sigma^2 R^{-1} \left(R^{-1}\right)' && (4.14) \end{aligned}$$

and that R^{-1} is easily computed by back substitution because R is upper triangular.

The confidence intervals over the entire response surface are calculated in the same manner as for linear circuits, as described in Section 3.4.2, using the final Jacobian from the nonlinear regression.

```
Q = J_(n×p)
for j = 1 to p
    while || q_j || = 0
        delete j^th column from J and R
        decrement p
        if j > p then return
    r_jj = || q_j ||
    for i = 1 to n
        q_ij = q_ij / r_jj
    for k = j + 1 to p
        r_jk = Σ_{i=1}^{n} q_ij q_ik
        for i = 1 to n
            q_ik = q_ik − q_ij r_jk
```

Figure 4.1: Pseudo-code for modified robust Gram-Schmidt orthonormalization.

4.1.2 Robust QR-decomposition

One of the distinguishing features of our algorithm is the automatic detection and correction of dependencies among the behavioral model parameters. Detecting and correcting these dependencies is essential because it:

1. Reduces the number of test vectors needed to test the system, and

2. Prevents the nonlinear regression algorithms from aborting due to rank-deficient Jacobian matrices.

Expecting the user to detect or correct these dependencies is undesirable because the rank of the Jacobian can change with each iteration of the nonlinear regression algorithm. Our fully automatic algorithm is embedded within the nonlinear regression loop and requires no additional CPU time above that already required for performing the regression. Parameter dependencies result in ambiguity groups, which are groups of parameters that are not independent. This algorithm represents a significant computational improvement over previously published algorithms for finding these groups [133, 67]. The improvement is possible because we correct the ambiguity group problem without explicitly identifying the groups; explicit identification of the groups is necessary for fault diagnosis, but not for production testing.

Each iteration of the nonlinear regression loop involves solving a linear regression problem of the form

$$J'J\vec{\delta} = J'\vec{h}. \tag{4.15}$$

We use the modified Gram-Schmidt orthonormalization (QR-decomposition) routine shown in Figure 4.1, which, as part of the matrix decomposition algorithm, sequentially considers each column of J and automatically discards those columns found to be linearly dependent upon previously considered columns.

Upon exit, the columns of Q will represent an independent subset $\vec{\delta}^*$ of the parameters $\vec{\delta}$, and the number of columns of Q will equal the rank of J. To solve for $\vec{\delta}^*$, we note that

$$\begin{aligned}
(QR)'(QR)\vec{\delta}^* &= (QR)'h && (4.16)\\
R'Q'QR\vec{\delta}^* &= R'Q'h && (4.17)\\
R'R\vec{\delta}^* &= R'Q'h && (4.18)\\
(R')^{-1}R'R\vec{\delta}^* &= (R')^{-1}R'Q'h && (4.19)\\
R\vec{\delta}^* &= Q'h && (4.20)
\end{aligned}$$

Equation 4.20 is easily solved using backward substitution, since R is upper triangular.

To calculate n, the minimum number of test vectors required to characterize the nominal system, we perform a robust QR-decomposition of the nominal Jacobian. The result is a set of independent parameters $\vec{\theta}^*$ that is a subset of all parameters $\vec{\theta}$; n is the number of parameters in $\vec{\theta}^*$. We need the QR-decomposition of the Jacobian to generate the confidence intervals, so finding $\vec{\theta}^*$ requires no additional computational effort.

4.2 Results

In this section we describe some practical examples of analog systems on which our ATPG algorithms have been run. The first is a bandpass filter with center frequency of 24.5 kHz [128], which was analyzed using SPICE sensitivity analysis. The second is a single MOS transistor, which was analyzed in SPICE with a level 3 transistor model.

We are currently applying the techniques described in this chapter to the testing of two complex fabricated circuits, namely a $\Sigma-\Delta$ A/D converter and a phase-locked loop.

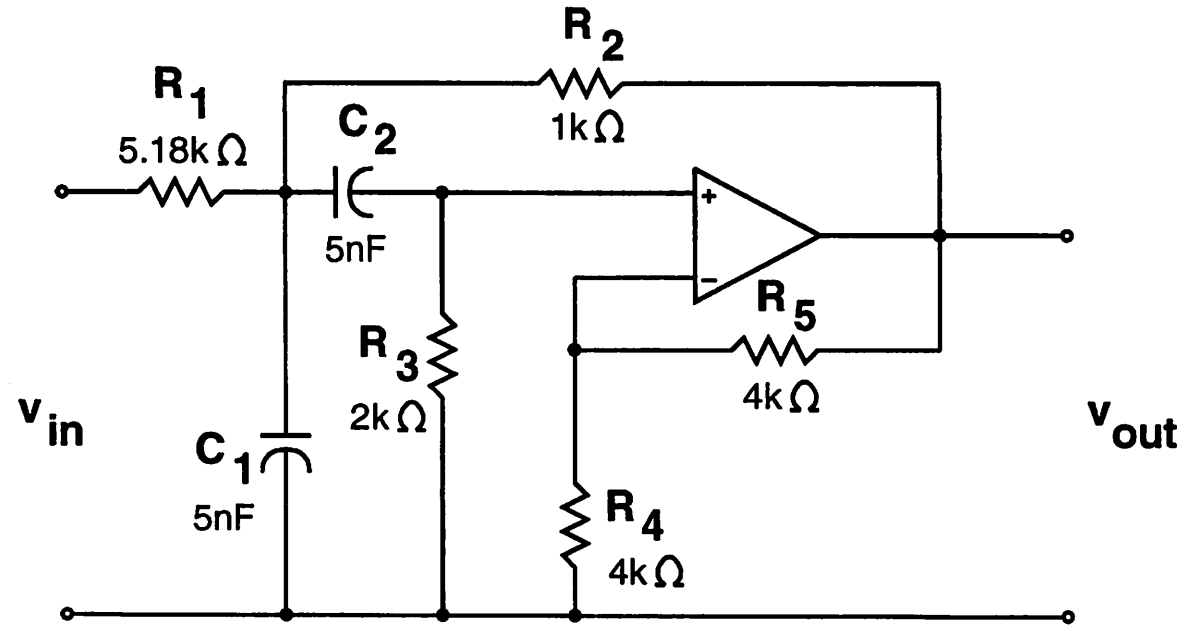


Figure 4.2: Bandpass filter with center frequency at 24.5 kHz.

4.2.1 Bandpass Filter

Figure 4.2 shows a linear model for a bandpass filter. The nominal frequency response is shown in Figure 4.3. The parameters $\vec{\theta}$ which characterize the filter are R_1, C_1, R_2, C_2, R_3, R_4, and R_5.

Performing a QR-decomposition on the nominal Jacobian reveals that its rank is 5, so at least 5 test frequencies will be needed to estimate the system response. We impose a constraint that the test frequencies lie between 15kHz and 40kHz, since that is the region of the response in which we are interested, and run the I-optimality algorithm. The 5 test frequencies which the algorithm selects are shown in Table 4.1. Note that the fifth test point is pushed to the user-imposed limit of 40kHz, while the remaining test points sample the response at intervals of approximately 3 kHz near the nominal center frequency.

Applying the five selected test frequencies to a simulated circuit produces the estimated output and 99% confidence intervals shown in Figure 4.4. According to the testing algorithm outlined in Section 4.1, these confidence intervals would be compared against the filter specifications to determine whether the component should be accepted or rejected, or whether additional test vectors should be applied to tighten the confidence intervals.

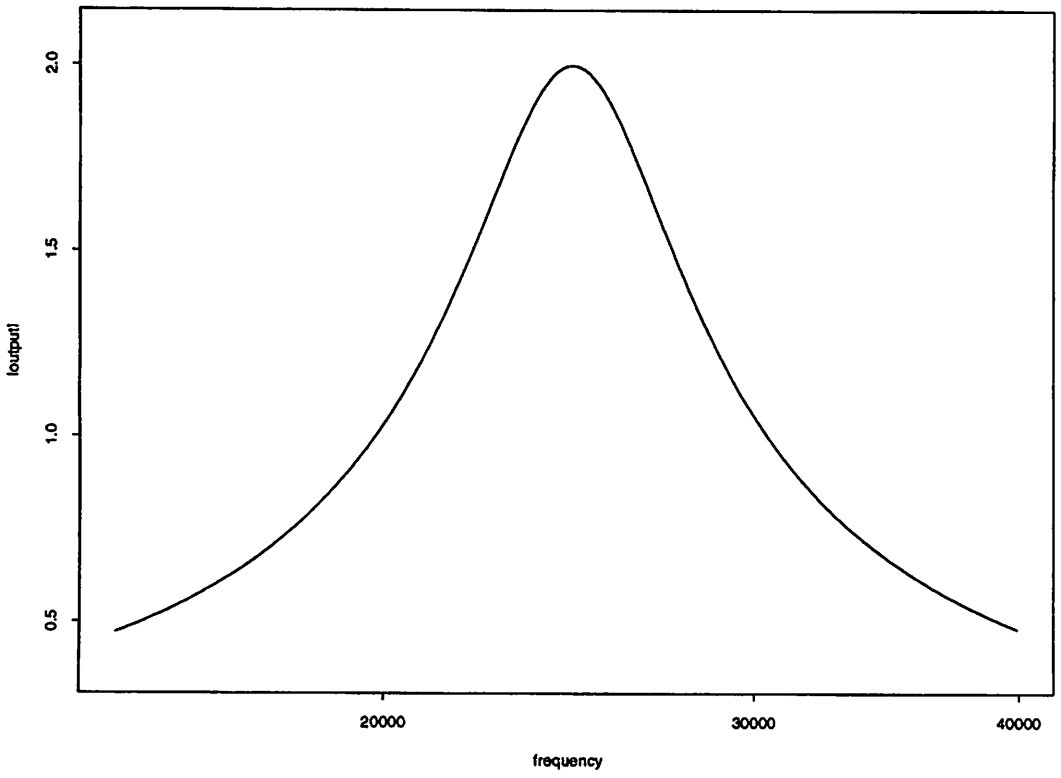


Figure 4.3: Nominal frequency response of bandpass filter.

Frequency	| Output |
19.32 kHz	0.914
22.57 kHz	1.65
24.89 kHz	1.99
28.42 kHz	1.30
40.00 kHz	0.482

Table 4.1: Test frequencies chosen for bandpass filter.

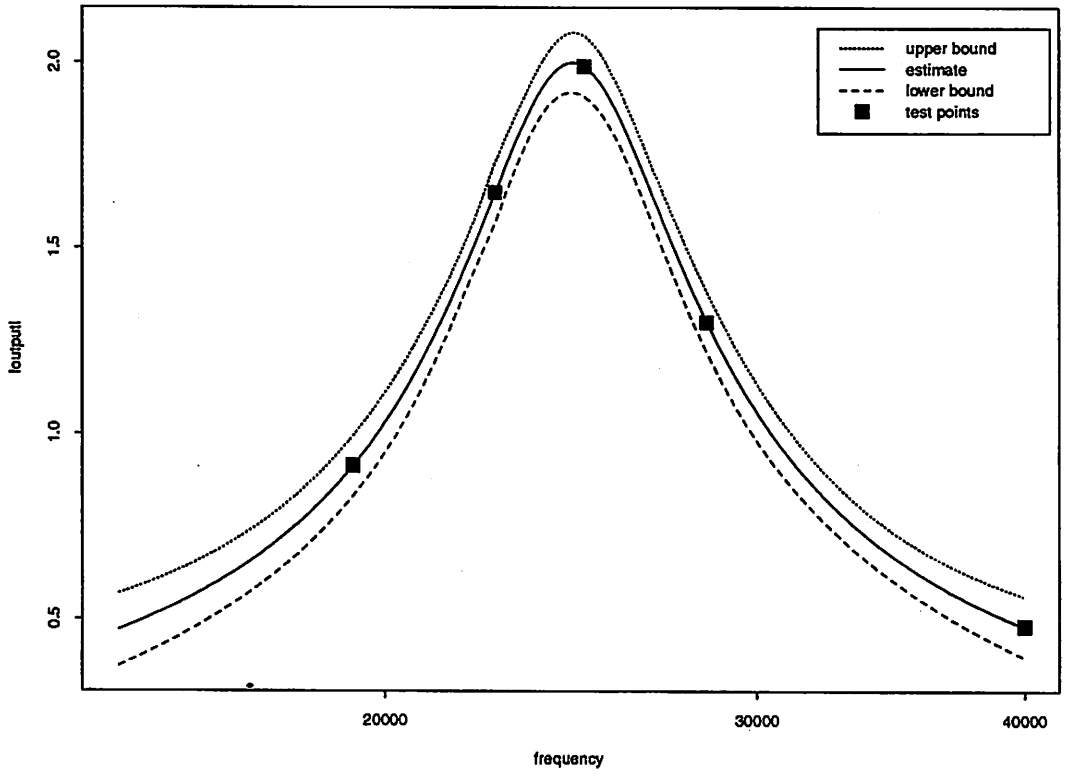


Figure 4.4: Estimated output and 99% confidence intervals for bandpass filter.

```
.MODEL nom NMOS LEVEL=3 PHI=0.600000 TOX=2.0300E-08 XJ=0.150000U
+ TPG=1 VTO=0.7333 DELTA=9.4450E-01 LD=1.0000E-09 KP=1.2964E-04
+ UO=762.1 THETA=5.2460E-02 RSH=2.3650E+00 GAMMA=0.4481
+ NSUB=1.7500E+16 NFS=2.3560E+12 VMAX=1.4870E+05 ETA=1.4850E-01
+ KAPPA=9.5100E-02 CGDO=2.5516E-12 CGSO=2.5516E-12
+ CGBO=3.0108E-10 CJ=1.1962E-04 MJ=0.4398 CJSW=4.6935E-10
+ MJSW=0.123994 PB=0.800000
```

Figure 4.5: SPICE model for MOS transistor.

4.2.2 MOS Transistor

Suppose we wish to test an MOS transistor to verify that its drain current I_{DS} falls within certain bounds over all values of V_{GS} and V_{DS}. The manufacturer has provided the level 3 SPICE model shown in Figure 4.5 for the device, with statistical parameters μ, V_{TO}, C_{ox}, γ, t_{ox}, and θ. The normalized sensitivities of I_{DS} to each of these parameters are shown graphically for three values of V_{GS} in Figure 4.6. Although only three values of V_{GS} are shown, both V_{DS} and V_{GS} are treated as continuous variables, so the response surface is 2-dimensional and continuous.

Performing a QR-decomposition of the nominal Jacobian matrix, we find that the C_{ox} parameter is not independent. If we include a constant "offset" parameter in our model, then there are 6 independent parameters, so $n = 6$ and we will need at least 6 test points to characterize the device.

To prevent the I-optimality algorithm from selecting unreasonable test points, we impose constraints on the inputs V_{GS} and V_{DS} such that $0.1V \leq V_{DS} \leq 10.0V$ and $2.0V \leq V_{GS} \leq 5.0V$. We then run the I-optimality algorithm; it selects the test points shown in Table 4.2.

Figure 4.7 shows the estimated response curves for three values of V_{GS} after applying the indicated 6 test vectors to a device, along with the 99% confidence intervals for those estimates. The confidence intervals are based upon a measurement accuracy of 0.1%. The expected value of the model error for three values of V_{GS} is shown in Figure 4.8, from which we conclude that our estimates are least accurate near $V_{GS} = V_T$. This result is not surprising, since that region of transistor operation is difficult to model.

V_{GS}	V_{DS}	I_{DS}
5.0	1.0	19.04910 x10^{-3}
3.2	0.9	10.00540 x10^{-3}
5.0	0.1	2.66950 x10^{-3}
5.0	10.0	33.62560 x10^{-3}
2.0	10.0	6.24432 x10^{-3}
5.0	2.2	27.45550 x10^{-3}

Table 4.2: Test points chosen for MOS transistor.

4.3 Conclusions

In this chapter we have presented a new ATPG algorithm which automatically generates test vectors for nonlinear analog systems. The algorithm chooses the set of test vectors so as to minimize the average prediction variance of the model. Applying the minimal set of test vectors to a circuit produces an estimate of the circuit's performance for all possible input vectors and, more importantly, confidence intervals on those estimates which can be used to determine whether the component should be passed or failed, or whether additional test vectors should be applied to tighten the confidence intervals.

Because these techniques generate the tightest possible confidence intervals after a minimum number of test vectors, they represent the most efficient way of fully characterizing system performance. Tight confidence intervals will lead to reduced testing time for analog systems because more components will be fully verifiable to a desired confidence level with the minimum number of test vectors. We have applied the algorithm to several analog systems and have shown it to be efficient and effective.

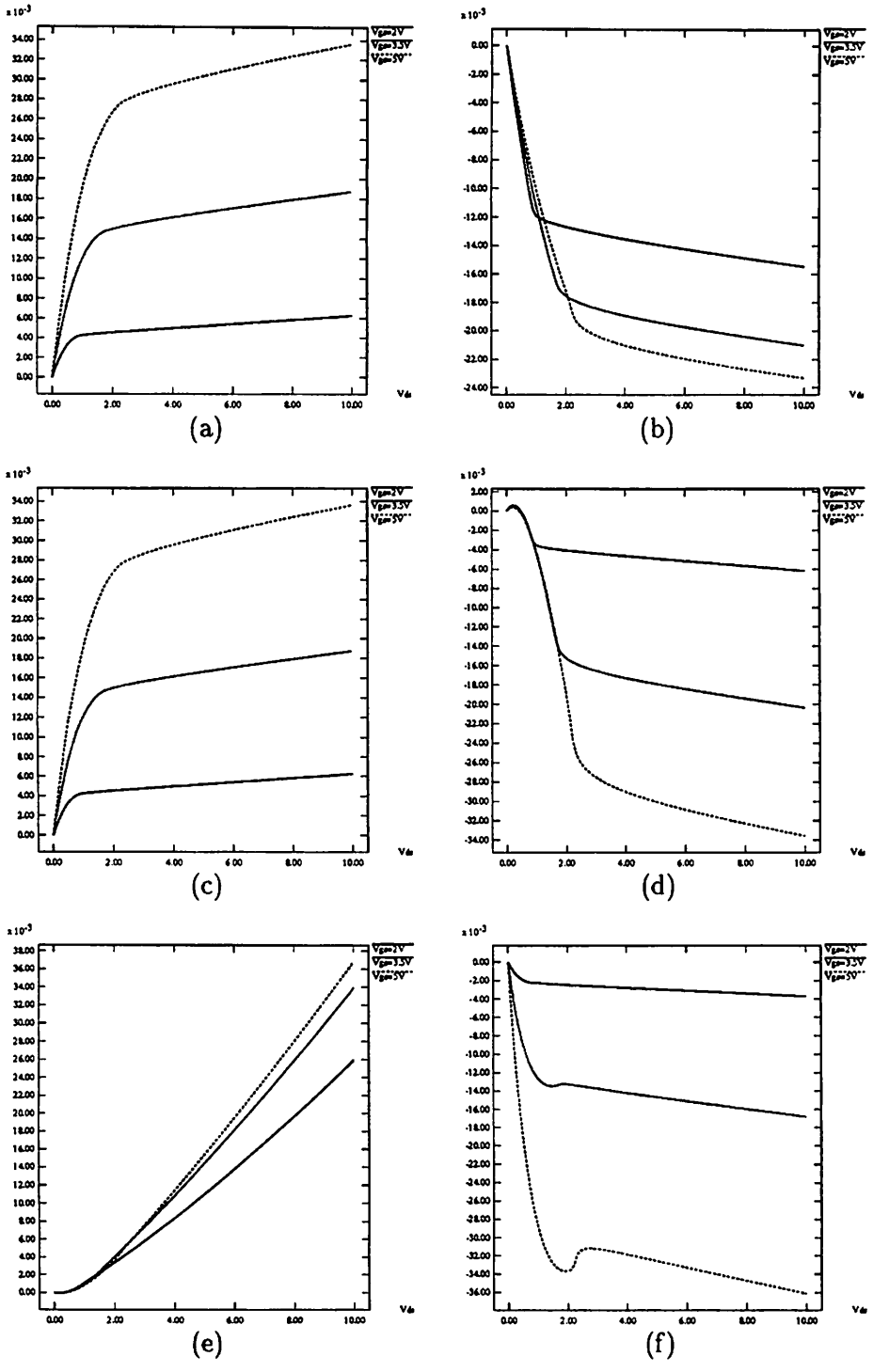


Figure 4.6: (a) Sensitivity w.r.t. μ. (b) Sensitivity w.r.t. V_{T0}. (c) Sensitivity w.r.t. C_{ox}. (d) Sensitivity w.r.t. γ. (e) Sensitivity w.r.t. t_{ox}. (f) Sensitivity w.r.t. θ.

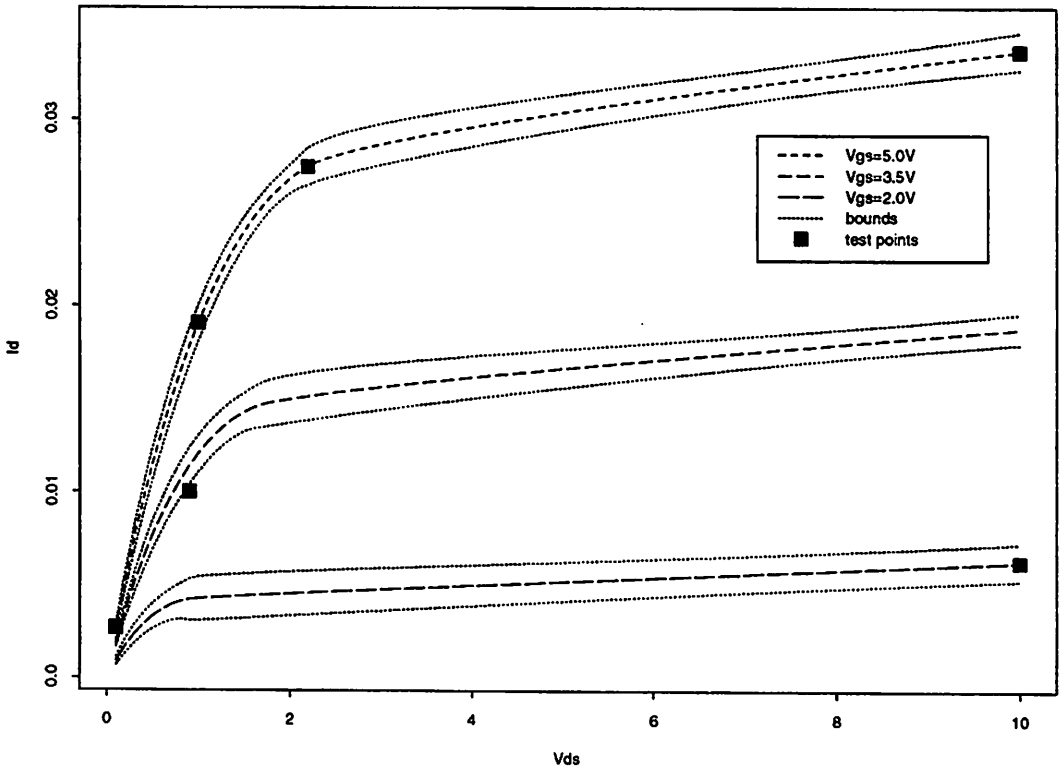


Figure 4.7: Estimated response and confidence intervals for MOS transistor from 6 test points.

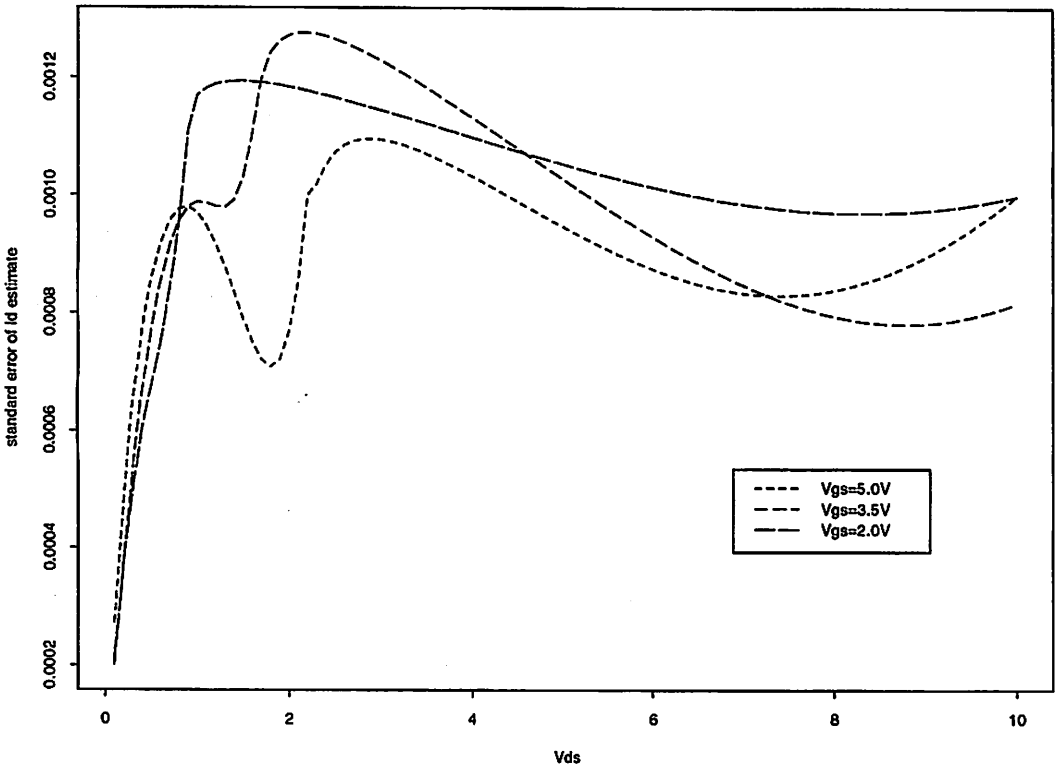


Figure 4.8: Standard error of estimated response for MOS transistor.

Chapter 5

Designing Optimal Analog Test Structures

In this chapter we present a methodology for designing optimal analog integrated circuit test structures. An optimal test structure is a circuit which allows one to characterize a specified set of circuit parameters as accurately as possible in the presence of measurement noise and other potential errors. The methodology is based upon recently developed statistical techniques for optimal design of experiments; these techniques allow analog systems to be characterized as accurately and efficiently as possible, thereby reducing cost and/or increasing accuracy. The usefulness of the methodology is illustrated with a fabricated circuit. The most interesting result is that relatively complex circuits are frequently more efficient than commonly used simple circuits.

5.1 Introduction

The design of test structures is currently performed on a relatively ad-hoc basis; a design or test engineer relies primarily upon intuition about the parameters of interest to create test structures which will permit those parameters to be measured. Emphasis is frequently placed on simplicity in either design or analysis. For example, one might create a test structure consisting of a single transistor, which is simple to design, or one might create a test structure consisting of an operational amplifier that is very sensitive to one parameter and insensitive to other parameters, which is simple to analyze.

In this chapter we attempt to make test structure design more systematic by

presenting a figure of merit which can be used to evaluate the relative *efficiency* of various candidate test structures, where we define efficiency to be the ratio of test accuracy to test effort. Comparing the efficiency of test structures will permit a test engineer to determine which structures are optimal for measuring a certain set of interesting parameters. This optimality information, when considered with area, shape, and other factors, can be used to determine which test structures should actually be fabricated.

The proposed methodology for optimal test structure design is presented in Section 5.2. In Section 5.3, the methodology is applied to three example test structures for measuring MOS transistor current mismatch.

5.2 Methodology

To compare several candidate test structures, we must calculate their relative efficiencies at estimating the parameters of interest in a small number of measurements. Under certain reasonable assumptions, accuracy is a monotonically increasing function of number of tests, so there is a tradeoff between accuracy and number of tests. Two interesting questions arise, as follows:

1. After a fixed number of well-chosen tests, how accurate are the predictions from each of the circuits?

2. To reach a fixed accuracy, how many well-chosen tests must be performed with each of the circuits?

Quantity 1 is easier to evaluate, and hence has been chosen as the primary figure of merit for our research. Based on the limited number of circuits we have examined, we conjecture that the relative ranking of candidate circuits obtained from Quantity 1 will almost always be the same as the relative ranking obtained from Quantity 2.

For a given circuit, the determination of "well-chosen tests" is nontrivial. Our methodology is based upon the statistical theory of optimal experimental design, in which test vectors are chosen to be maximally independent so that the model parameters of interest will be characterized as accurately as possible in the presence of measurement noise and model inaccuracies. More specifically, we wish to choose the test vectors to minimize the average standard error of the predicted parameters, which is a function of the choice of test vectors. Intuitively, the test vectors should be as orthogonal to each other as possible, where

```
for each candidate circuit {
   find minimum-size D-optimal set of tests
   calculate and output D-value
}
```

Figure 5.1: Pseudo-code of optimal test structure design algorithm.

the orthogonality of the test vectors is measured by the degree to which each test vector maximizes the contribution of the basis function corresponding to one parameter while minimizing the contribution of the basis functions corresponding to the other parameters.

Given a circuit which is characterized by n independent parameters, at least n test vectors must be applied to the system in order to fully characterize those parameters. If fewer than n input vectors are applied, then at least one dimension of the circuit response space remains unexplored and hence at least one parameter cannot be estimated. Furthermore, because of inevitable measurement noise, n test vectors may not be sufficient to characterize the circuit parameters to the desired accuracy. Using additional test vectors will lower the standard error of the estimates; in practice, if the variance of the predicted parameters is too large after the minimum number of tests, then additional test vectors can be selected and applied to reduce the standard error of the predicted parameters until the desired accuracy is obtained.

To compare the relative efficiency of two proposed test structures, we compare the *normalized D-values of the minimum-size D-optimal test set* for each circuit. The D-value is the average variance of the estimated model coefficients, so a circuit with a lower D-value is a more efficient test structure than a circuit with a higher D-value. The inputs to the algorithm are a set of candidate circuits and a list of parameters of interest. The output is a list of the normalized D-values corresponding to the D-optimal test set for each circuit.

Pseudo-code is shown in Figure 5.1. Calculating the D-value for a given set of test vectors is described in Section 5.2.1, and finding a D-optimal set of test vectors for a given circuit is described in Section 5.2.2.

5.2.1 Calculating D-Values

Given a circuit model and a set of test vectors, the D-value can be easily computed using the method of least squares. Consider an arbitrary circuit with output characterized

by a linear combination of independent basis functions, as in Chapter 3,

$$Y = \beta_0 g_0 + \beta_1 g_1 + \beta_2 g_2 + \beta_3 g_3 + \ldots + \beta_{p-1} g_{p-1} + \epsilon \tag{5.1}$$

where Y is the system output, $\{g_i\}$ is a set of arbitrary basis vectors, β_i is the coefficient of the i^{th} basis vector, and ϵ represents the measurement and modeling errors, which are assumed to be independent with mean 0 and constant variance σ^2. Many analog systems can be accurately modeled in this fashion by using sensitivity analysis [132] or QR decomposition [129]. ϵ may be either specified by the designer or estimated from previous tests. Note that Equation 5.1 is linear in the unknowns $\{\beta_i\}$, but the basis functions $\{g_i\}$ can be nonlinear.

Let X represent the $n \times p$ design matrix, which contains one row for each of the n test vectors.

$$X = \begin{bmatrix} g_1(x_1) & g_2(x_1) & g_3(x_1) & \ldots & g_p(x_1) \\ g_1(x_2) & g_2(x_2) & g_3(x_2) & \ldots & g_p(x_2) \\ & & \vdots & & \\ g_1(x_n) & g_2(x_n) & g_3(x_n) & \ldots & g_p(x_n) \end{bmatrix} \tag{5.2}$$

Using the method of least squares, the best estimate of β, denoted by $\hat{\beta}$, is given by

$$\hat{\beta} = \left(X^T X\right)^{-1} X^T Y \tag{5.3}$$

where X is the design matrix described above and Y is a vector of the circuit output from each of the n test vectors. The variance-covariance matrix of these estimated parameters is

$$\mathcal{D}\left[\hat{\beta}\right] = \sigma^2 \left(X^T X\right)^{-1} \tag{5.4}$$

where $\mathcal{D}\left[\hat{\beta}\right]$ is the variance-covariance matrix of $\hat{\beta}$ and σ^2 is the variance of ϵ, the error term in Equation 5.1.

The diagonal entries of D correspond to the variances of the model parameters, and the D-value is given by

$$D = \left|\left(X^T X\right)^{-1}\right|. \tag{5.5}$$

A design which minimizes D is said to be D-optimal. D-optimality minimizes the average prediction variance of the model coefficients, which is the most appropriate figure of merit for evaluating competing test structures.

5.2.2 Generating D-Optimal Test Sets

Generating a set of test vectors for a given circuit model which minimizes D, as defined in Equation 5.5, is believed to be an NP-complete optimization problem [23]. An exact solution is only feasible for very small problems. For larger problems, several heuristic algorithms have been successfully used to find "good" solutions to this and other related problems in the area of optimal experimental design. For this research we used the gradient descent techniques implemented in GOSSET.

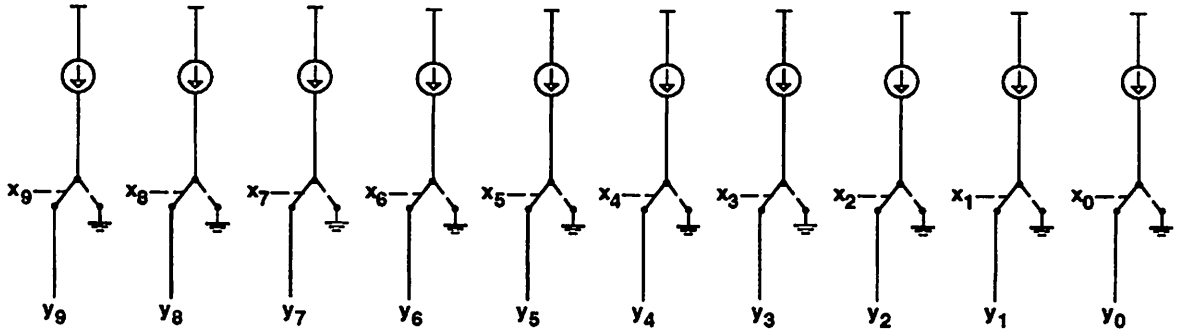
GOSSET is a very general computer program for constructing experimental designs [49]. Variables may be discrete or continuous, discrete variables may be numeric or symbolic, and continuous variables may range over a cube or a ball. The variables may be required to satisfy linear equalities or inequalities, and the model to be fitted may be any linear function (Equation 5.1). The number of tests is specified by the user, and the design may be required to include a specified set of points. The software is powerful enough to routinely minimize functions of 1000 variables.

As used in our algorithm, GOSSET finds an optimal test set for each test structure and outputs the normalized D-value corresponding to that test structure. The D-value is normalized by scaling each variable to range between -1 and +1. This normalization is essential for meaningful circuit-to-circuit comparisons of D-values.

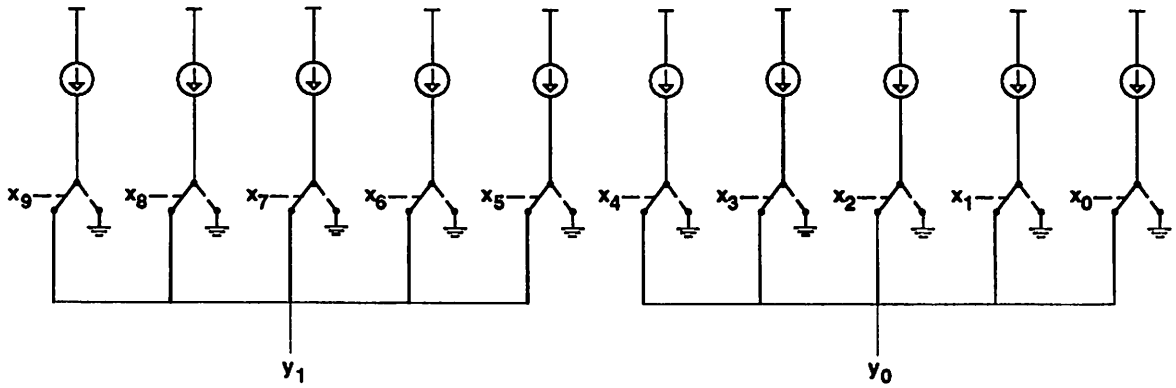
An important error condition which must be considered is the case when it is not possible to estimate all of the parameters of interest from a given test structure. This situation results in a singular X matrix, and the D-value output by the algorithm in these cases is $+\infty$.

5.3 Results

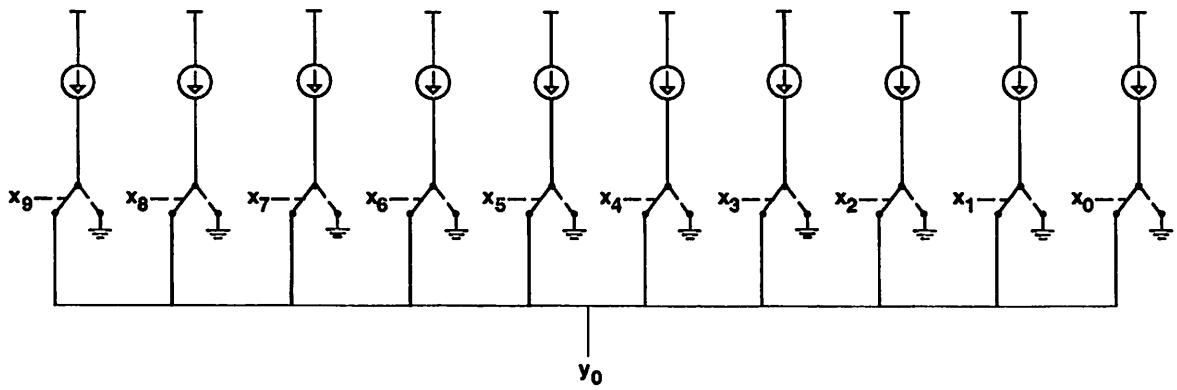
As an example of the optimal test structure design methodology, suppose we wish to design a test structure for measuring MOS transistor current mismatch. Mismatch is defined as the variance in current flowing through identically designed transistors. We plan to estimate this variance by sampling 10 transistors. Three candidate test structures are shown in Figure 5.2. The structures all contain the same 10 transistors, which are represented as switchable current sources, but differ in the extent to which the current outputs of those transistors are wired together. The 10 parameters of interest are the "on"



(a)



(b)



(c)

Figure 5.2: (a) Proposed test structure for Circuit 1. (b) Proposed test structure for Circuit 2. (c) Proposed test structure for Circuit 3.

Circuit	Optimum D-value
#1	0.1895
#2	0.1402
#3	0.1063

Table 5.1: Normalized D-values for minimum-size test sets.

currents through each of the transistors.

The GOSSET programs written to calculate the D-values are shown in Figure 5.3. The constraints imposed by the connectivity of the current source outputs are translated into constraints on groups of current sources which cannot be simultaneously observed.

The normalized D-values for the best minimum-size test sets are shown in Table 5.1. The minimum-size test sets themselves are shown in Table 5.2.

Figure 5.4 plots the optimal D-value which can be obtained as a function of the average number of transistors turned on during each test, i.e. the percentage of ones in the test set. From this graph it is apparent that the most efficient test structures are those which permit test sets containing 50% ones.

Figure 5.5 plots D-value as a function of number of tests for each of the example circuits. From this graph it is apparent that Circuit 3 is always more efficient than Circuits 1 or 2.

Six test structures similar to Circuit 3 have been fabricated by MOSIS to characterize CMOS transistor current mismatch. There were 12 replications of each test structure, with 16-64 transistors per test structure. A summary of the observed variances is shown in Table 5.3; the complete results of this mismatch characterization experiment are reported in [35].

With regard to optimal test structure design, the important conclusion to be drawn from these examples is that extracting parameters from relatively complex structures is often more efficient than measuring single devices. The reason for this phenomenon is that in the complex circuit each parameter is, in effect, sampled multiple times and hence the effective measurement noise and model inaccuracies are reduced. This result is similar to measuring a single device multiple times and averaging the measurements, but fewer tests are required when a complex circuit is used.

```
10 discrete x0 x1 x2 x3 x4 x5 x6 x7 x8 x9 0 1
20 model x0+x1+x2+x3+x4+x5+x6+x7+x8+x9
30 constraint x0+x1+x2+x3+x4+x5+x6+x7+x8+x9<1.5
```

```
10 discrete x0 x1 x2 x3 x4 x5 x6 x7 x8 x9 0 1
20 model x0+x1+x2+x3+x4+x5+x6+x7+x8+x9
30 constraint x0+x5<1.5
40 constraint x1+x5<1.5
50 constraint x2+x5<1.5
60 constraint x3+x5<1.5
70 constraint x4+x5<1.5
80 constraint x0+x6<1.5
90 constraint x1+x6<1.5
100 constraint x2+x6<1.5
110 constraint x3+x6<1.5
120 constraint x4+x6<1.5
130 constraint x0+x7<1.5
140 constraint x1+x7<1.5
150 constraint x2+x7<1.5
160 constraint x3+x7<1.5
170 constraint x4+x7<1.5
180 constraint x0+x8<1.5
190 constraint x1+x8<1.5
200 constraint x2+x8<1.5
210 constraint x3+x8<1.5
220 constraint x4+x8<1.5
230 constraint x0+x9<1.5
240 constraint x1+x9<1.5
250 constraint x2+x9<1.5
260 constraint x3+x9<1.5
270 constraint x4+x9<1.5
```

```
10 discrete x0 x1 x2 x3 x4 x5 x6 x7 x8 x9 0 1
20 model x0+x1+x2+x3+x4+x5+x6+x7+x8+x9
```

Figure 5.3: From top to bottom, GOSSET programs for Circuits 1, 2, and 3, respectively.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	1	1	0	0	1
0	0	1	0	1	0	0	0	0	0
0	1	0	1	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
0	1	0	0	0	0	1	1	0	0
0	1	1	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	1	1	1	1
1	0	1	1	0	1	1	1	1	0
1	1	0	0	0	1	0	1	1	1
1	1	0	0	1	1	1	0	1	0
1	1	0	1	0	0	1	0	1	1
1	1	1	0	0	1	1	0	0	1
1	1	1	1	1	0	0	1	0	0

Table 5.2: From top to bottom, D-optimal test sets for Circuits 1, 2, and 3, respectively.

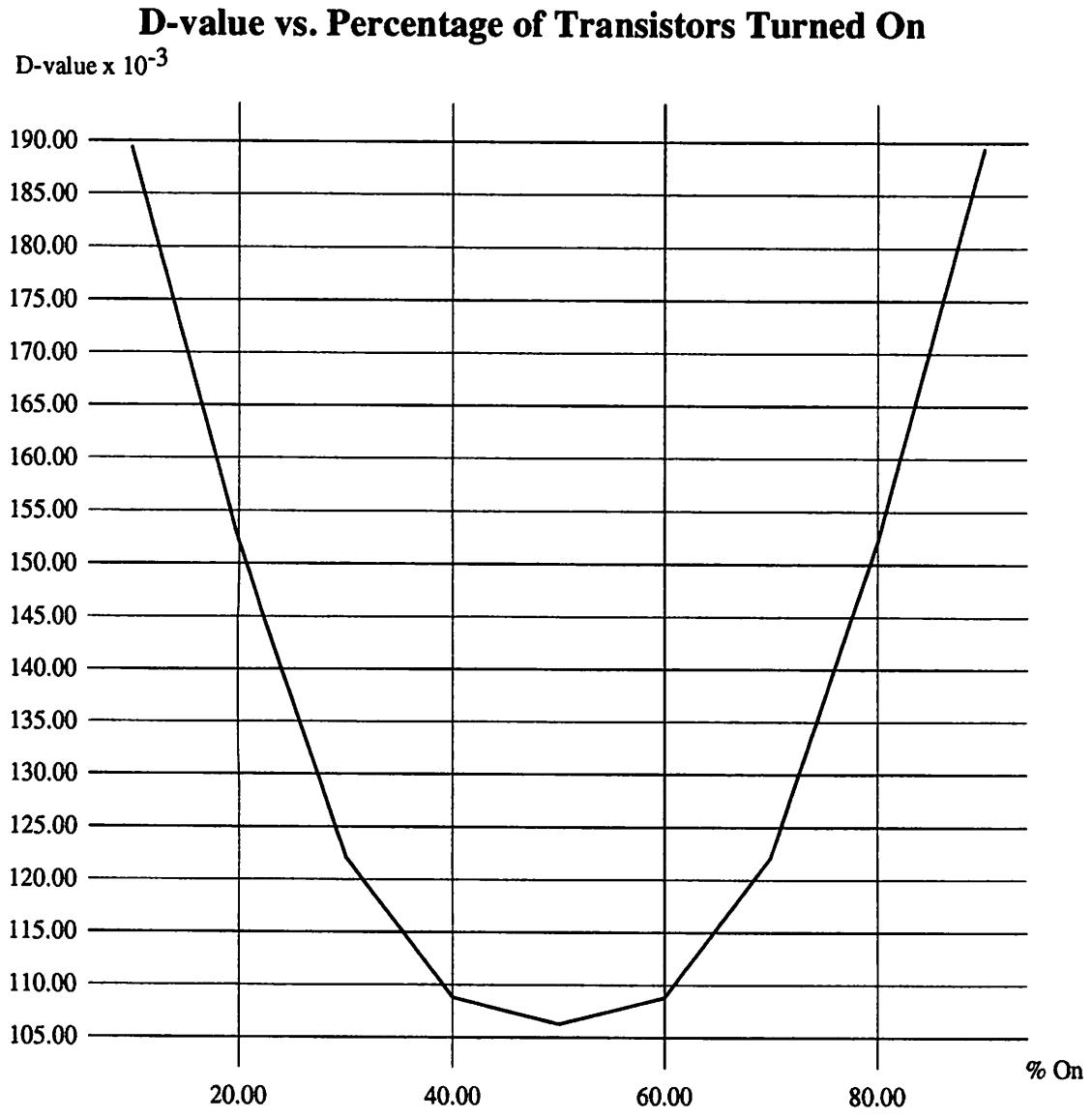


Figure 5.4: D-value vs. percentage of transistors turned on.

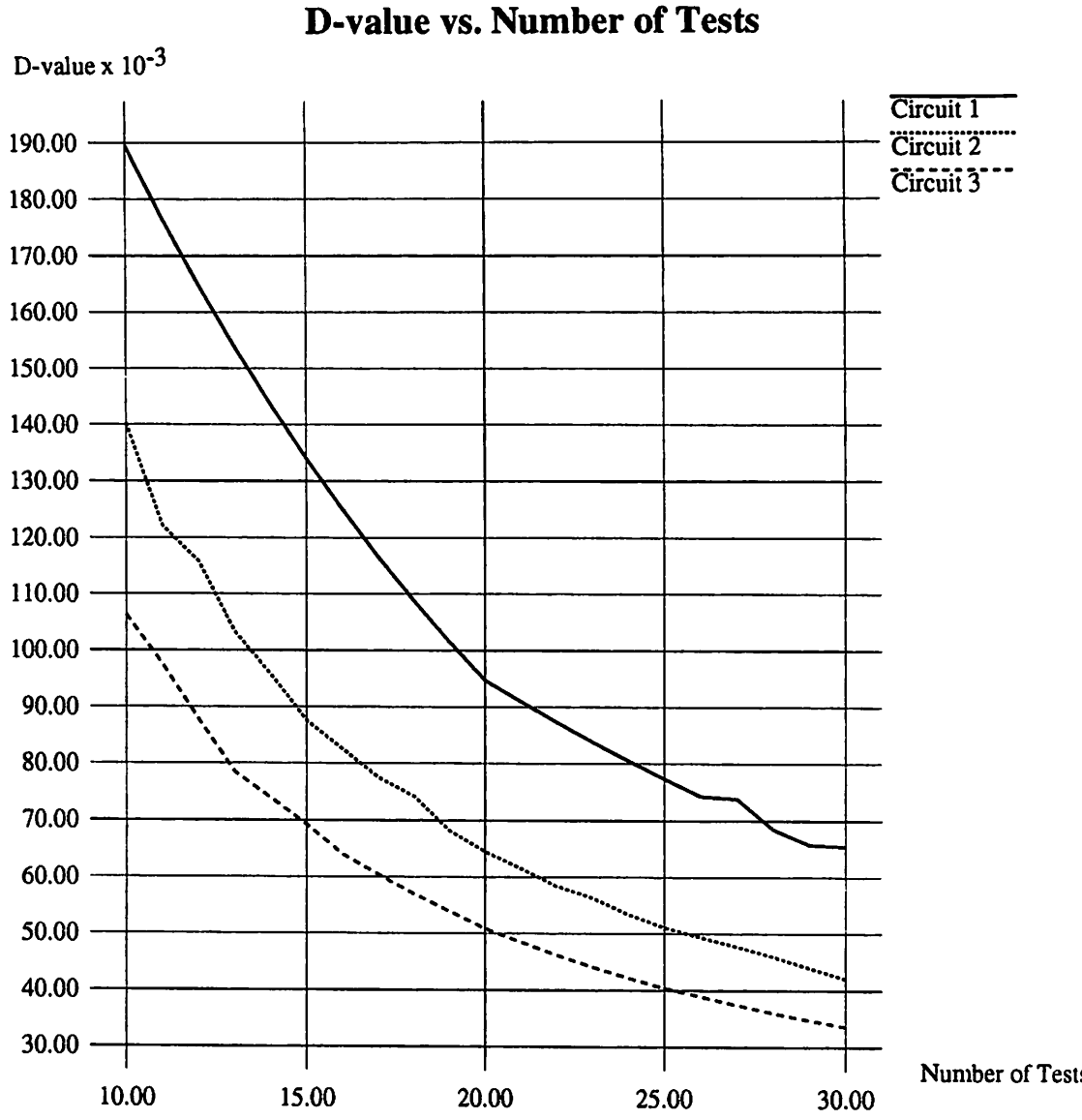


Figure 5.5: D-value vs. number of tests for each circuit.

description	# of transistors	W (μm)	L (μm)	area (μm^2)	$V_{GS} - V_T$ (V)	measured σ_I/I
DAC46_lin_1mA	192	121	24	2904	0.679	0.00374
DAC46_bin_1mA	768	48	110	5280	0.288	0.01153
DAC55_lin_1mA	384	21	24	504	1.163	0.00374
DAC55_bin_1mA	384	15	48	720	0.344	0.01005
DAC64_lin_1mA	768	28	21	588	0.660	0.00424
DAC64_bin_1mA	192	22	47	1034	0.278	0.01029
DAC46_lin_0.6mA	192	121	24	2904	0.526	0.00458
DAC46_bin_0.6mA	768	48	110	5280	0.223	0.01374
DAC64_lin_0.6mA	768	28	21	588	0.511	0.00500
DAC64_bin_0.6mA	192	22	47	1034	0.216	0.01300

Table 5.3: Measured mismatch for 10 sets of measurements on 6 "optimal" test structures.

5.4 Conclusions

We have presented a methodology for developing optimal analog IC test structures which is based on statistical principles of optimal experimental design. To compare the relative efficiency of two proposed test structures, we compare the normalized D-values of the minimum-size D-optimal test set for each circuit.

The methodology has been illustrated with three example test structures which might be used to characterize MOS transistor current mismatch. Of these three candidates, the optimal test structure was fabricated and mismatch was successfully extracted.

Chapter 6

Statistical Characterization of Analog Circuits

This chapter presents a methodology for hierarchical statistical circuit characterization which does not rely upon circuit-level Monte Carlo simulation. The methodology uses principal component analysis, response surface methodology, and statistics to directly calculate the statistical distributions of higher-level parameters from the distributions of lower-level parameters. We have used the methodology to characterize a folded cascode operational amplifier and a phase-locked loop. This methodology permits the statistical characterization of large analog and mixed-signal systems, many of which are extremely time-consuming or impossible to characterize using existing methods.

6.1 Introduction

Statistical circuit characterization is essential for estimating yield, for designing manufacturable and robust systems, for deriving "worst-case" models, and for testing. The most widely used technique for performing statistical characterization is Monte Carlo analysis [47, 104]. Unfortunately, the accuracy of results produced by a Monte Carlo analysis is only proportional to the square root of the number of simulations performed, and the number of Monte Carlo simulations required to produce a relatively accurate result increases exponentially with the number of low-level statistical parameters. Therefore Monte Carlo techniques can be very expensive, unacceptably inaccurate, or both.

One promising approach to dealing with these shortcomings involves the use of

behavioral models and hierarchical characterization. Hierarchical characterization is illustrated in Figure 6.1. This characterization method is part of a hierarchical design methodology which involves different levels of abstraction [18]. The low-level parameters typically represent transistor model parameters, such as t_{ox} and V_{T0}. The intermediate-level parameters typically represent behavioral model parameters, such as open-loop gain and offset of an operational amplifier. The high-level performances represent circuit performance specifications, such as signal-to-noise ratio of an analog-to-digital converter. A circuit simulator such as SPICE [57] is used to simulate the intermediate-level parameters as functions of the low-level parameters, and a behavioral-level simulator such as MIDAS [155] is used to simulate the high-level performances as functions of the intermediate-level parameters.

In this hierarchical design methodology, two statistical characterizations are performed. First, the statistical distributions of the intermediate-level parameters are calculated from those of the low-level parameters. Second, the statistical distributions of the high-level circuit performances are calculated from the intermediate-level parameters. The first characterization can be quickly performed with the non-Monte Carlo techniques described in this chapter. The second characterization can be performed either in the same way or using Monte Carlo analysis. Monte Carlo analysis is generally acceptable for the second characterization if the behavioral model being used is fast and involves only a relatively small number of statistical parameters, which is often the case.

The non-Monte Carlo techniques described in this chapter utilize response surface methodology (RSM) [83]. RSM involves constructing a circuit model which is locally linear or quadratic in the statistical parameters. The RSM model is constructed by performing an "experiment" in which the lower-level parameters are permuted in a regular fashion about their nominal values. For each permutation of the lower-level parameters, a simulation is performed and the resultant values of the higher-level parameters are recorded. The coefficients of the RSM model are then obtained by linear regression. SIMPILOT [118] is a commercial tool which implements RSM.

At each level of the statistical characterization it is essential to consider the *correlations* between parameters, as independent parameters are uncommon. Computing and using a variance-covariance matrix of the parameters at each level of the hierarchy can properly account for parameter correlations and, furthermore, provides an excellent conduit for incorporating parameter mismatch information into circuit models. These variance-covariance matrices are one of the most important cornerstones of our methodology.

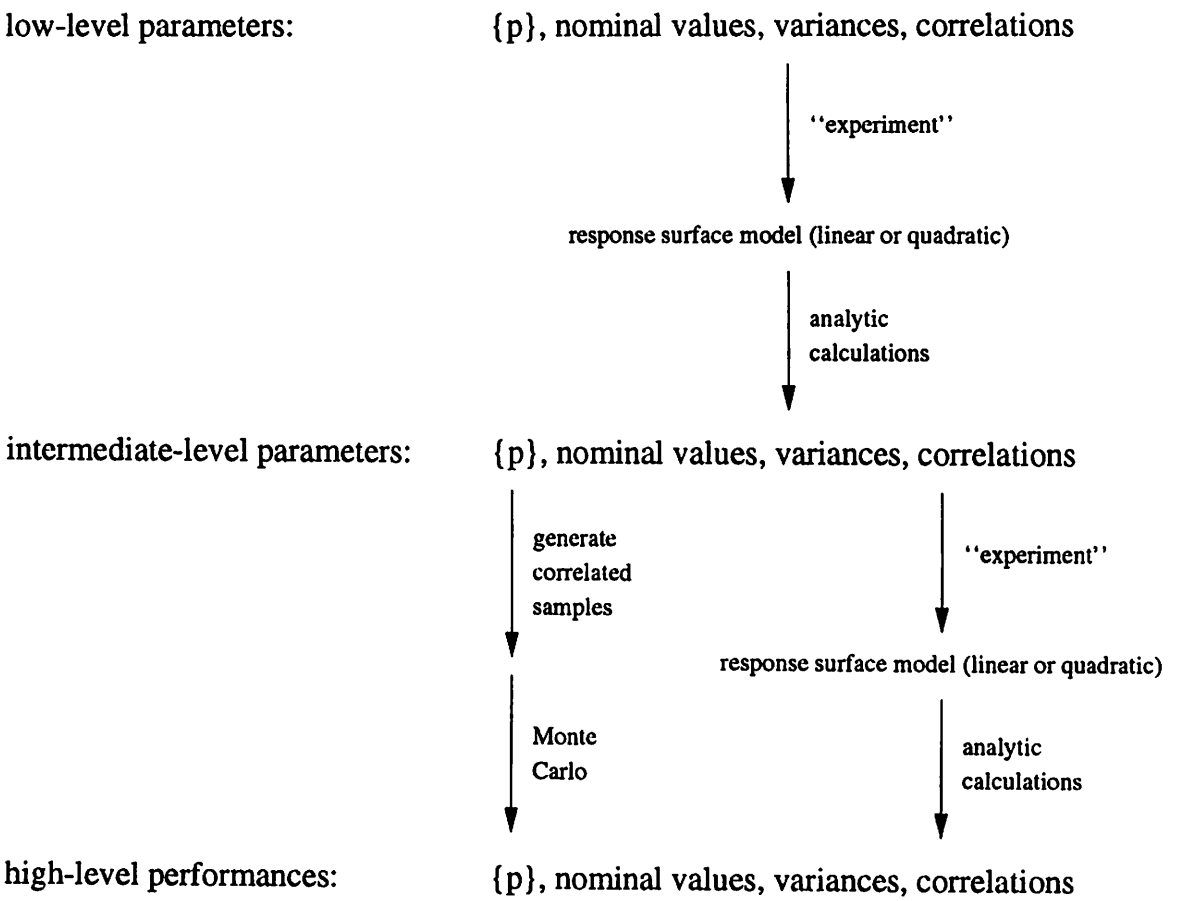


Figure 6.1: Hierarchical characterization.

With these factors in mind, a typical flow of our statistical characterization process begins with a set of low-level process parameters, their nominal values, their variances, and their correlations. We construct an experiment and carry out simulations to build the quadratic response surface models for each component in the circuit. We use analytic formulas to calculate the means, variances, and correlations of the intermediate-level parameters. We then perform Monte Carlo analysis at the behavioral level, using correlated sets of random variables, to determine the distributions and correlations of the high-level system performances.

Our key new contributions to this method of hierarchical statistical characterization, as shown in Figure 6.1, are in three areas:

1. a method for incorporating parameter mismatch and correlation into the response surface models,

2. a method for directly calculating the expected values, variances, and correlations of higher-level parameters from those of lower-level parameters, and

3. a method for generating correlated sets of parameters for Monte Carlo analysis at the behavioral level.

These contributions improve the efficiency and accuracy of statistical circuit characterization.

6.2 Parameter Mismatch and Correlation

Most MOS models are characterized by a relatively large number of parameters, only a few of which are statistically independent [158]. *Principal component analysis* (PCA) or *principal factor analysis* (PFA) can be used to extract the statistically relevant combinations of parameters and thereby reduce the number of lower-level parameters which must be considered [154, 55]. Given a set of model cards which have been extracted from fabricated devices, SPAYN [130] is a commercial tool which performs PCA and PFA. This technique typically results in 2-3 statistically relevant principal components per transistor, which can explain at least 75% of the observed variation in 15 level 3 MOS model parameters.

In order to properly account for parameter mismatch, we use a separate model card for each transistor in the circuit. Correlations between transistors are specified in the

	$M1_{pc1}$	$M1_{pc2}$	$M2_{pc1}$	$M2_{pc2}$	C_1	C_2
$M1_{pc1}$	1	0	0.9	0	0	0
$M1_{pc2}$	0	1	0	0.9	0	0
$M2_{pc1}$	0.9	0	1	0	0	0
$M2_{pc2}$	0	0.9	0	1	0	0
C_1	0	0	0	0	1	0.8
C_2	0	0	0	0	0.8	1

Figure 6.2: Example variance-covariance matrix for low-level parameters.

variance-covariance matrix. The correlation coefficients will be functions of transistor areas, distances between transistors, and V_{gs}, according to appropriate mismatch models. Parameters on the same die will typically have relatively high correlation coefficients, approaching the limiting case of 1 for no mismatch. Note that using a single model card for multiple transistors, while common, corresponds to this limiting case of no mismatch and can produce inaccurate statistical characterizations. An example variance-covariance matrix is shown in Figure 6.2. In this example, there are two orthogonal principal components for each transistor. There are two transistors whose parameters are 90% correlated. There are two capacitors which are 80% correlated to each other and uncorrelated to the transistor parameters.

6.3 Analytic Statistical Calculations

Once an appropriate variance-covariance matrix for the statistically-relevant low-level parameters has been obtained, we use SIMPILOT or a similar program to construct the linear or quadratic response surface models for each intermediate-level parameter. Constructing this model involves defining an appropriate experiment, which in SIMPILOT is typically a simplex experiment for linear models or a Latin hypercube for quadratic models, running ELDO [31] (SPICE) for each permutation in the experiment, and using linear regression to solve for the coefficients of the response surface model.

Once a linear or quadratic response surface model has been found, the expected values, variances, and correlations of the intermediate-level parameters can be directly computed, regardless of the distributions of the low-level parameters. Therefore it is usually not necessary to resort to Monte Carlo analysis, as SIMPILOT does; direct analytic solutions

are faster and more accurate.

Let X be a p-dimensional vector of random variables which represents the lower-level parameters, with $\mathcal{E}[X] = \theta$ and variance-covariance matrix $\mathcal{D}[X] = \Sigma$. Let Y be an n-dimensional vector representing the higher-level parameters. We wish to calculate $\mathcal{E}[Y]$ and $\mathcal{D}[Y]$.

Considering the linear case first, let C be an $n \times p$ matrix of constants representing the statistically significant coefficients in the linear model, so that $Y = CX$. Theorems 6.3.1 and 6.3.2 prove that $\mathcal{E}[Y] = C\theta$ and $\mathcal{D}[Y] = C\Sigma C'$, respectively. Note that these theorems do not make any assumptions about the distribution of the low-level parameters.

Theorem 6.3.1

$$\mathcal{E}[CX] = C\theta \tag{6.1}$$

Proof: Let $Y = CX$. Then $y_i = \sum_{r=1}^{m} c_{ir} x_r$, and

$$\begin{aligned}
\mathcal{E}[CX] &= [(E[y_i])_i] && (6.2)\\
&= \left[\left(\sum_{r=1}^{m} c_{ir}\mathcal{E}[x_r]\right)_i\right] && (6.3)\\
&= [(C\mathcal{E}[X])_i] && (6.4)\\
&= C\theta && (6.5)
\end{aligned}$$

Theorem 6.3.2

$$\mathcal{D}[CX] = C\Sigma C' \tag{6.6}$$

Proof: Let $Y = CX$. Then

$$\begin{aligned}
\mathcal{D}[CX] &= \mathcal{D}[Y] && (6.7)\\
&= \mathcal{E}\left[(Y - \mathcal{E}[Y])(Y - \mathcal{E}[Y])'\right] && (6.8)\\
&= \mathcal{E}\left[(CX - C\mathcal{E}[X])(CX - C\mathcal{E}[X])'\right] && (6.9)\\
&= \mathcal{E}\left[C(X - \mathcal{E}[X])(X - \mathcal{E}[X])'C'\right] && (6.10)\\
&= C\mathcal{E}\left[(X - \mathcal{E}[X])(X - \mathcal{E}[X])'\right]C' && (6.11)\\
&= C\Sigma C' && (6.12)
\end{aligned}$$

For the quadratic case, let A be a $p \times p$ symmetric matrix representing the statistically significant coefficients in the quadratic model for any one higher-level parameter

y_i, so that $y_i = X'AX$. Note that for any given coefficients in a quadratic equation, A is uniquely determined [113]. Let $tr\,(A)$ denote the trace of A. Theorems 6.3.3 and 6.3.4 show how $\mathcal{E}[y_i]$ and $var[y_i]$ can be calculated.

Theorem 6.3.3

$$\mathcal{E}\,(X'AX) = tr\,(A\Sigma) + \theta'A\theta \tag{6.13}$$

Proof:

$$\begin{aligned}\mathcal{E}\,[X'AX] &= \mathcal{E}\left[(X-\theta)'\,A\,(X-\theta) + \right.\\ &\qquad \left.\theta'AX + X'A\theta - \theta'A\theta\right] \end{aligned}\tag{6.14}$$

Since $X'A\theta = (X'A\theta)' = \theta'A'X = \theta'AX$
and $\mathcal{E}\,[\theta'AX] = \theta'A\mathcal{E}\,[X] = \theta'A\theta$,

$$\begin{aligned}\mathcal{E}\,[X'AX] &= \mathcal{E}\left[(X-\theta)'\,A\,(X-\theta)\right] + \theta'A\theta &(6.15)\\ &= \sum_i\sum_j a_{ij}\mathcal{E}\,[(x_i-\theta_i)\,(x_j-\theta_j)] + \theta'A\theta\\ &= \sum_i\sum_j a_{ij}\sigma_{ij} + \theta'A\theta &(6.16)\\ &= tr\,[A\Sigma] + \theta'A\theta &(6.17)\end{aligned}$$

Theorem 6.3.4

$$\begin{aligned}var\,[X'AX] &= \mathcal{E}\left[\left((X-\theta)'\,A\,(X-\theta)\right)^2\right] +\\ &\quad 4\mathcal{E}\left[(\theta'A\,(X-\theta))^2\right] +\\ &\quad 4\mathcal{E}\left[\theta'A\,(X-\theta)\,(X-\theta)'\,A\,(X-\theta)\right] -\\ &\quad (tr\,(A\Sigma))^2\end{aligned}\tag{6.18}$$

Proof:

$$var\,[X'AX] = \mathcal{E}\left[(X'AX)^2\right] - (\mathcal{E}\,[X'AX])^2 \tag{6.19}$$

$$X'AX = (X-\theta)'\,A\,(X-\theta) + 2\theta'A\,(X-\theta) + \theta'A\theta \tag{6.20}$$

Letting $W = X - \theta$,

$$\begin{aligned}(X'AX)^2 &= (W'AW)^2 + 4\,(\theta'AW)^2 + (\theta'A\theta)^2 +\\ &\quad 2\theta'A\theta\,(W'AW + 2\theta'AW) +\\ &\quad 4\theta'AWW'AW\end{aligned}\tag{6.21}$$

Using 6.3.3,

$$\begin{aligned}\mathcal{E}\left[(X'AX)^2\right] &= \mathcal{E}\left[(W'AW)^2\right] + 4\mathcal{E}\left[(\theta'AW)^2\right] + \\ &\quad (\theta'A\theta)^2 + 2\theta'A\theta\,(tr\,(A\Sigma)) + \\ &\quad 4\mathcal{E}\left[\theta'AWW'AW\right] \qquad (6.22)\end{aligned}$$

$$(\mathcal{E}\left[X'AX\right])^2 = (tr\,(A\Sigma))^2 + (\theta'A\theta)^2 + 2\theta'A\theta tr\,(A\Sigma) \qquad (6.23)$$

$$\begin{aligned}var\left[X'AX\right] &= \mathcal{E}\left[(W'AW)^2\right] + 4\mathcal{E}\left[(\theta'AW)^2\right] + \\ &\quad 4\mathcal{E}\left[\theta'AWW'AW\right] - (tr\,(A\Sigma))^2 \qquad (6.24)\end{aligned}$$

Evaluating (6.18) requires the second, third, and fourth moments of the joint probability density function for X and thus can be complicated in the general case. When X can be assumed to follow a multivariate normal distribution, i.e. $X \sim \mathcal{N}_p(\theta, \Sigma)$, then

$$\mathcal{E}\left[(W'AW)^2\right] = (tr\,(A\Sigma))^2 + 2tr\,(A\Sigma)^2, \qquad (6.25)$$

$$\mathcal{E}\left[(\theta'AW)^2\right] = \theta'A\Sigma A\theta, \text{ and} \qquad (6.26)$$

$$\mathcal{E}\left[\theta'AWW'AW\right] = 0. \qquad (6.27)$$

Theorem 6.3.5 follows immediately [114].

Theorem 6.3.5 If $X \sim \mathcal{N}(\theta, \Sigma)$, then

$$var\left[X'AX\right] = 2tr\,(A\Sigma)^2 + 4\theta'A\Sigma A\theta \qquad (6.28)$$

To compute the off-diagonal elements of $\mathcal{D}[Y]$, we need to compute $cov\left[y_i, y_j\right]$ for all i, j. Let A and B be the symmetric matrices representing the coefficients of the quadratic models for two higher-level parameters y_A and y_B, so that $y_A = X'AX$ and $y_B = X'BX$. Theorem 6.3.6 is used to compute $cov\left[y_i, y_j\right]$ [113].

Theorem 6.3.6 If $X \sim \mathcal{N}_p(\theta, \Sigma)$, then

$$cov\,(X'AX, X'BX) = 2tr\,(A\Sigma B\Sigma) + 4\theta'A\Sigma B\theta \qquad (6.29)$$

Proof: Let $T = [X' \;\; X']$ be the $(2p)$-dimensional vector formed by replicating X. $T \sim \mathcal{N}_{2p}(\mu, C)$, where $\mu = [\theta' \;\; \theta']$ and $C = \begin{bmatrix} \Sigma & \Sigma \\ \Sigma & \Sigma \end{bmatrix}$. Let $W = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}$. Then

$$T'WT = X'AX + X'BX \tag{6.30}$$

$$var\left[T'WT\right] = var\left[X'AX\right] + var\left[X'BX\right] - 2cov\left[X'AX, X'BX\right] \tag{6.31}$$

$$\begin{aligned} cov\left[X'AX, X'BX\right] &= \frac{1}{2}\Big(2tr\left(WC\right)^2 + 4\mu'WCW\mu - \\ &\quad \Big(2tr\left(A\Sigma\right)^2 + 4\theta'A\Sigma A\theta\Big) - \\ &\quad \Big(2tr\left(B\Sigma\right)^2 + 4\theta'B\Sigma B\theta\Big)\Big) \end{aligned} \tag{6.32}$$

$$= 2tr\left(A\Sigma B\Sigma\right) + 4\theta'A\Sigma B\theta \tag{6.33}$$

Our IC fabrication experience has shown that the low-level parameters generally *do* follow a normal or log-normal distribution, so normality of the low-level parameters, as required by Theorems 6.3.5 and 6.3.6, is a reasonable assumption. One frequently-cited theoretical justification for this assumption is the central limit theorem applied to the physical fabrication process.

If the low-level parameters X can be assumed to be multivariate normal, $X \sim \mathcal{N}_p\left[\theta, \Sigma\right]$, and a linear model is used, then the intermediate-level parameters Y will also be multivariate normal, $Y \sim \mathcal{N}_n\left[C\theta, C\Sigma C'\right]$. When X is multivariate normal and a quadratic model is used, then $\left(X-\theta\right)'A\left(X-\theta\right) \sim \chi^2_r$ if and only if $A\Sigma A = A$, where r is the rank of A [114]. Otherwise the distribution of Y does not follow an easily-computable form. In practice, however, one introduces little error by assuming that the intermediate parameters are approximately multivariate normal, even when a quadratic model is used.

Our C functions for calculating the expected values and variance-covariance matrix using (6.1), (6.6), (6.13), (6.28), and (6.29) accept as inputs the vector θ and the matrix Σ, which define the joint distributions of the low-level parameters, and a coefficient matrix C in which each row represents the appropriately-ordered response surface coefficients for one intermediate-level parameter. For example, if the response surface models for two intermediate-level parameters, p_0 and p_1, are

$$p_0 = c_{00} + c_{01}x_1 + c_{02}x_1^2 + c_{03}x_2 + c_{04}x_2x_1 + c_{05}x_2^2 \tag{6.34}$$

and

$$p_1 = c_{10} + c_{11}x_1 + c_{12}x_1^2 + c_{13}x_2 + c_{14}x_2x_1 + c_{15}x_2^2, \tag{6.35}$$

then

$$C = \begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} & c_{04} & c_{05} \\ c_{10} & c_{11} & c_{12} & c_{13} & c_{14} & c_{15} \end{bmatrix}. \tag{6.36}$$

The C functions for the linear case are straightforward. For the quadratic case, the expected value function loops over each intermediate-level parameter, calling (6.13) to compute the expected value of that parameter. Similarly, the variance-covariance function loops over each combination of intermediate-level parameters, calling (6.28) or (6.29) to compute the appropriate entry in the variance-covariance matrix for that combination. A utility function converts a row of the matrix C into a symmetric matrix of the appropriate form to be used as A or B.

6.4 Correlated Parameters at the Behavioral Level

Using the techniques outlined in Sections 2 and 3 we can calculate the nominal values, variances, and correlations of the intermediate-level parameters. If there are a large number of correlated intermediate-level parameters, then PFA or PCA can be used again, in the same fashion as for the low-level parameters, to reduce the number of parameters which must be considered for the behavioral modeling. Given the distributions of the intermediate-level parameters, the next step is to calculate the distributions of the high-level performances. We can either repeat the RSM-based procedure used to characterize the intermediate-level parameters or we can perform a Monte Carlo simulation. Monte Carlo simulations at the behavioral level are feasible if there are a relatively small number of intermediate-level parameters and each evaluation of the behavioral model is fast.

When performing these behavioral-level Monte Carlo simulations, it is essential that the correlations between the intermediate-level parameters be properly considered; treating them as independent will usually produce overly pessimistic results. The way to do this is to generate *correlated* sets of random numbers. Suppose we want a $p \times 1$ vector of random variables to be correlated, with variance-covariance matrix Σ. We can form the Cholesky decomposition of Σ to obtain an upper triangular matrix U, where

$$\Sigma = U'U \tag{6.37}$$

If we generate a $p \times 1$ vector of independent random variables X, with $\mathcal{E}[X] = 0$ and $\mathcal{D}[X] = I$, then $U'X$ will have $\mathcal{E}[U'X] = 0$ and $\mathcal{D}[U'X] = \Sigma$. Therefore pre-multiplying X by U' induces the desired correlations.

$$
\begin{aligned}
&\text{for } i = 1 \text{ to } p \ \{ \\
&\quad \sigma_{ii} = \sqrt{\sigma_{ii} - \textstyle\sum_{k=1}^{i-1} \sigma_{ki}^2} \\
&\quad \text{for } j = i+1 \text{ to } p \ \{ \\
&\qquad \sigma_{ij} = \frac{\sigma_{ij} - \sum_{k=1}^{i-1} \sigma_{ki}\sigma_{kj}}{\sigma_{ii}} \\
&\quad \} \\
&\}
\end{aligned}
$$

Figure 6.3: Pseudo-code for computing the Cholesky decomposition of Σ.

	Linear Model		Quadratic Model		Monte Carlo Analysis	
	Nominal	St. Dev.	Nominal	St. Dev.	Nominal	St. Dev.
gain	110.2 dB	2.001 dB	110.2 dB	2.086 dB	110.2 dB	2.000 dB
pole_1	902.0 Hz	310.1 Hz	911.4 Hz	336.5 Hz	917.2 Hz	347.2 Hz
pole_2	4.025 MHz	0.763 MHz	4.028 MHz	0.766 MHz	4.026 MHz	0.783 MHz
r_{in}	414.2 GΩ	19.73 GΩ	414.5 GΩ	20.34 GΩ	413.9 GΩ	19.88 GΩ
zero_1	3.971 MHz	0.752 MHz	3.973 MHz	0.754 MHz	3.971 MHz	0.770 MHz
CPU	24.1 s		120.5 s		2258.6 s	

Table 6.1: Expected values and standard deviations of intermediate-level parameters.

Pseudo-code for computing the Cholesky decomposition of a symmetric positive semidefinite $p \times p$ matrix Σ is shown in Figure 6.3. Note that all variance-covariance matrices are symmetric and positive semidefinite [114].

6.5 Results

The statistical characterization techniques described in this chapter have been tested by performing statistical characterizations of two circuits. The first circuit is a folded cascode operational amplifier, which illustrates the building of a statistical behavioral model from a SPICE-level block. The second circuit is a phase-locked loop, which illustrates our complete methodology using multiple levels of hierarchy.

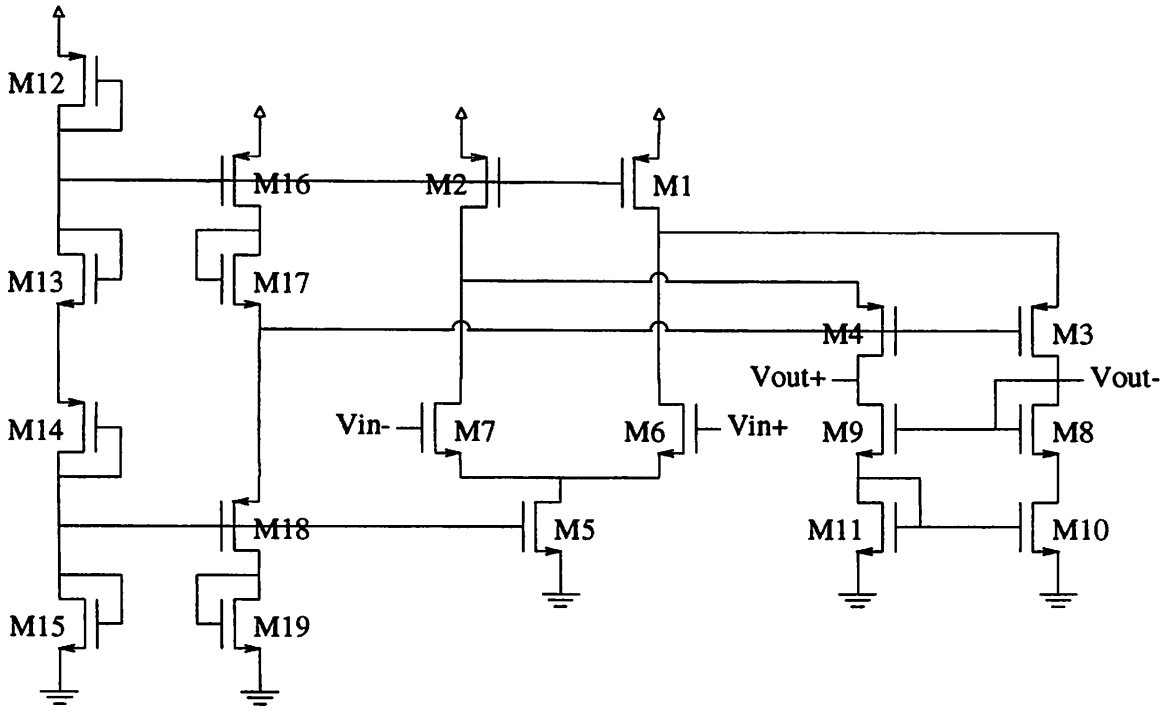


Figure 6.4: Transistor-level schematic of operational amplifier circuit.

6.5.1 Folded Cascode Operational Amplifier

A transistor-level schematic of the folded cascode operational amplifier is shown in Figure 6.4. We statistically characterized five intermediate-level parameters: gain, pole_1, pole_2, r_{in}, and zero_1. These quantities represent the parameters which might be needed for a behavioral model of this operational amplifier.

For the statistical MOS models we used the example database distributed with SPAYN, which contains level 3 parameters for both p- and n-type transistors. Since no mismatch data was available, we assumed perfect transistor parameter matching (correlation = 1).

The statistically relevant transistor parameters were found using PCA in SPAYN to be n_{pc1} and n_{pc2} for the n-type transistors and p_{pc1} and p_{pc2} for the p-type. Considering also the variations in load capacitors and DC voltage sources, the complete set of low-level parameters for this example was $\{n_{pc1}, n_{pc2}, p_{pc1}, p_{pc2}, c_1, c_2, v_1, v_2\}$.

The intermediate-level parameters were defined to be {gain, pole_1, pole_2, r_{in}, zero_1}. Offset would have also been included as an intermediate-level parameter if tran-

sistor parameter correlation (mismatch) information had been available. The linear and quadratic response surface models for these intermediate-level parameters were found using SIMPILOT. Using these models, the appropriate functions from Section 3 were used to compute the expected values, standard deviations, and variance-covariance matrix of the intermediate-level parameters. The results of these analytic calculations and the CPU times on a DEC 7000 Model 610 AXP workstation are shown in Table 6.1.

For comparison to these analytic results, a 1,000-run Monte Carlo analysis was performed on the same circuit. The resulting expected values, standard deviations, and CPU time are also shown in Table 6.1. Note that the Monte Carlo results match the analytic results quite closely.

Correlated samples of these intermediate-level parameters were generated by computing the Cholesky decomposition U of the variance-covariance matrix found for the quadratic models, as discussed in Section 4. These correlated samples can be used in behavioral-level Monte Carlo analysis when this operational amplifier is included in larger systems.

6.5.2 Phase-Locked Loop

A block diagram of a commercially available PLL which is used as a clock multiplier and for deskewing is shown in Figure 6.5. The phase/frequency detector compares the phase and frequency of the input signal to the reference signal. If the frequency of the reference signal needs to be increased, then the signal *up* is asserted and the charge pump adds charge to the node V_c. Similarly, if the frequency of the reference signal needs to be decreased, then the signal *down* is asserted and the charge pump subtracts charge from the node V_c. The voltage controlled oscillator generates a frequency corresponding to the voltage on node V_c; when the PLL is locked, the frequency generated by the oscillator is 12 times the input frequency.

The high-level performance which we wish to statistically characterize is the lock time, which we define as the time after which V_c lies in a band that is within 1.5% of its average value for the next 1 μs. Calculating the lock time of the PLL using a transistor-level netlist requires more than 24 hours of CPU time on a Sun Ultra Sparc workstation, so traditional Monte Carlo methods would require thousands of days of CPU time and hence are impractical.

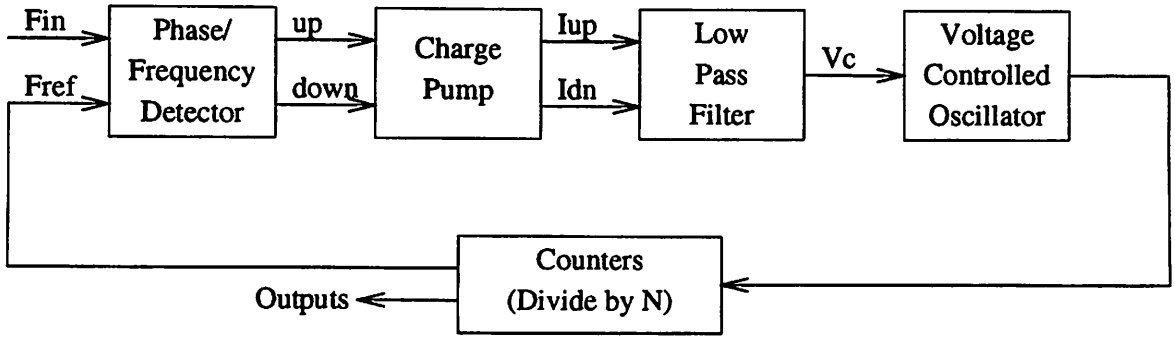


Figure 6.5: Block diagram of PLL.

The intermediate-level parameters for the behavioral model of the voltage controlled oscillator are

1. gain, in MHz/V, and

2. $f_{0.8}$, the output frequency when $V_c = 0.8$ V.

The intermediate-level parameters for the behavioral model of the phase/frequency detector and charge pump are

1. I_{up} and

2. I_{dn}.

The behavioral models are written in HDLA [51].

6.5.2.1 MOS Model Extraction

Statistical MOS models are needed to characterize the blocks in the PLL. To obtain these models, we measured a sample of 100 dies from 5 wafers and 2 lots of a 0.5 μm double poly 3.3 V technology. Each die contained 5 NMOS and 5 PMOS transistors with W/L dimensions of 10 μm/0.5 μm, 10 μm/0.4 μm, 2 μm/10 μm, 0.8 μm/10 μm, and 10 μm/10 μm. SGS-Thomson Level 3 NMOS and PMOS models were extracted for each die, with 28 parameters per model. The accuracy of the models is within 5%. An example of extraction is shown in Figure 6.6.

The total measurement time was 45 hours using UTMOST [143] and a prober driven by a Sun Sparc 10. Extracting the models from the measurements took 25 hours of CPU

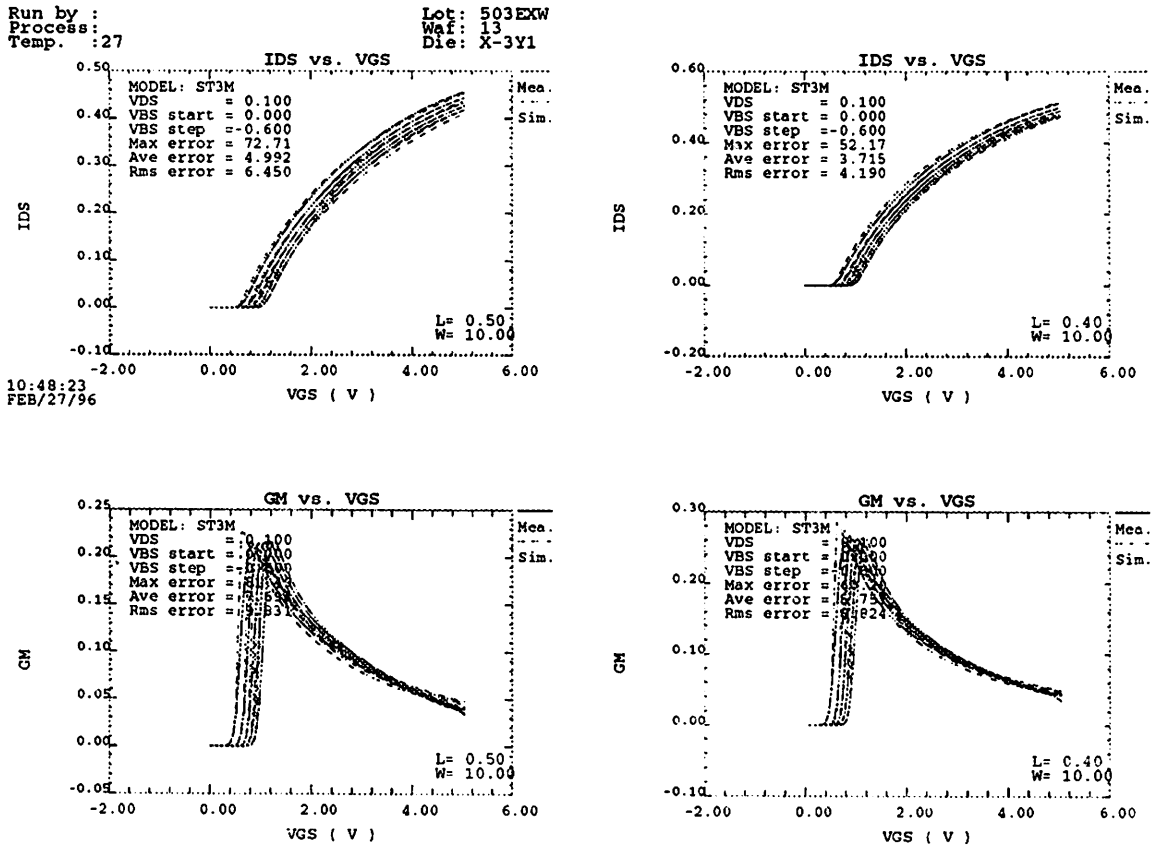


Figure 6.6: Extraction using UTMOST.

Component	Distribution
PC1	log normal
PC2	log normal
PC3	Gaussian
PC4	Gaussian
PC5	negative log normal
PC6	Gaussian
PC7	negative log normal
PC8	Gaussian

Table 6.2: Distributions of principal components of MOS models.

time on a Sun Sparc 20. The extracted models for 7 of the 100 dies were grossly inaccurate; those dies were discarded.

The model cards were analyzed using principal component analysis and 8 statistically significant principal components were found. Three distributions were considered for each principal component: Gaussian, log normal, and negative log normal. Note that a log normal distribution is the distribution of $y = e^x$ when x is Gaussian and a negative log normal is the distribution of $z = t - y$ where t is any real number. For each principal component, the distribution which produces the best fit is chosen. The resulting distributions are shown in Table 6.2 and a histogram of principal component 7 is shown in Figure 6.7. Regardless of distribution, each principal component is standardized to have mean = 0 and standard deviation = 1.

6.5.2.2 Behavioral Model Parameters

Given the statistical transistor models, the next step is to compute the distributions of the intermediate-level behavioral model parameters. We begin by building the linear and quadratic response surface models of the intermediate-level parameters as functions of the principal components of the MOS models.

To calculate the voltage-controlled oscillator parameters, gain and $f_{0.8}$, we run transient simulations at four input voltages, measuring the frequency as the average frequency of the last 25 of 120 periods at each input voltage. Gain is calculated as the slope of the least squares estimate of the straight-line function of frequency as a function of input voltage. $F_{0.8}$ is the frequency when the input is at 0.8 V. The accuracies of the linear and

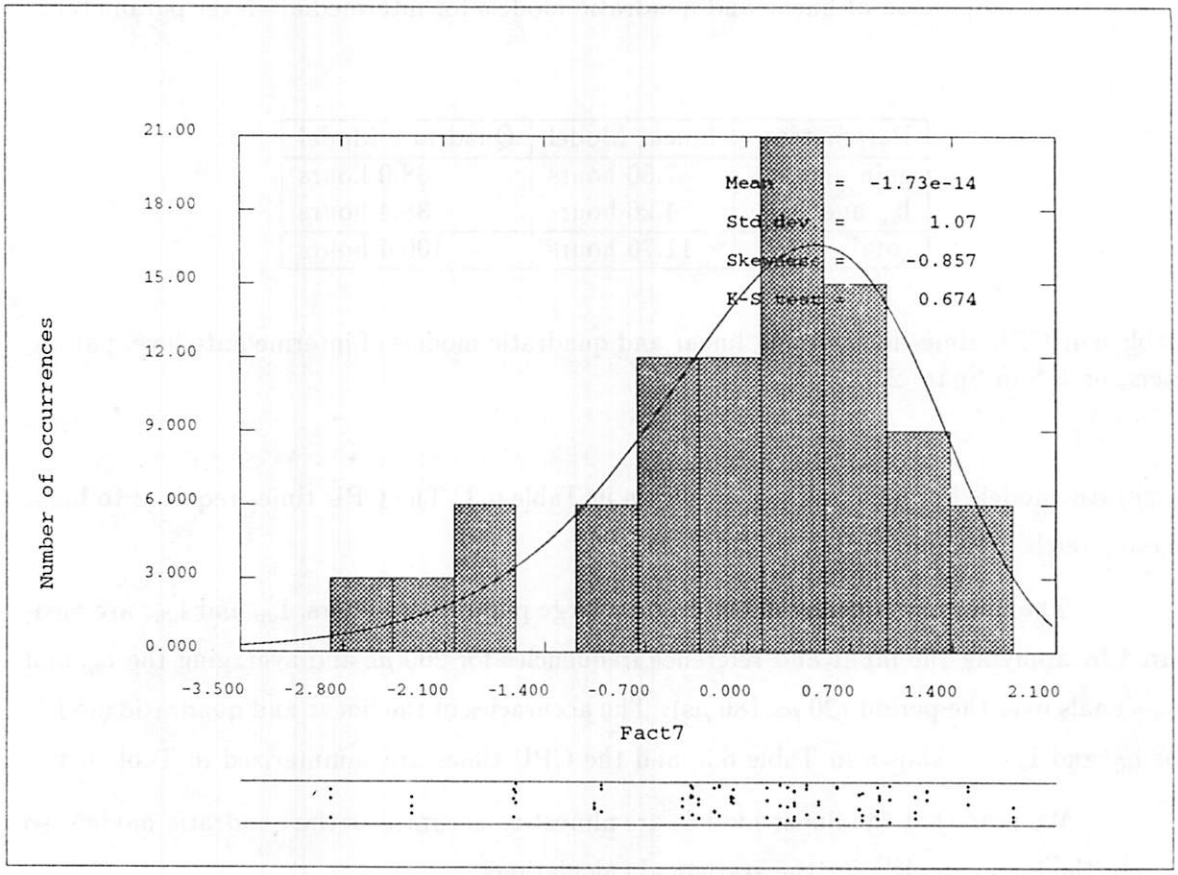


Figure 6.7: Histogram of principal component 7 of MOS models.

Parameter	Linear Model		Quadratic Model	
	Accuracy	Worst Error	Accuracy	Worst Error
gain	92.46%	2.39%	92.60%	-1.87%
$f_{0.8}$	77.53%	-13.07%	81.48%	-10.33%
I_{up}	74.18%	-7.35%	78.15%	-4.31%
I_{dn}	73.28%	-7.39%	77.42%	-4.35%

Table 6.3: Comparison of linear and quadratic models for intermediate-level parameters.

Parameters	Linear Model	Quadratic Model
gain and $f_{0.8}$	7.50 hours	68.0 hours
I_{up} and I_{dn}	4.26 hours	38.4 hours
total:	11.76 hours	106.4 hours

Table 6.4: CPU times for building linear and quadratic models of intermediate-level parameters, on a Sun Sparc 20.

quadratic models for gain and $f_{0.8}$ are shown in Table 6.3. The CPU times required to build these models are summarized in Table 6.4.

The phase/frequency detector and charge pump parameters, I_{up} and I_{dn}, are measured by applying the input and reference frequencies for 200 μs and averaging the I_{up} and I_{dn} signals over the period (20 μs,180 μs). The accuracies of the linear and quadratic models for I_{up} and I_{dn} are shown in Table 6.3, and the CPU times are summarized in Table 6.4.

We note that the linear models are almost as accurate as the quadratic models, so we use the linear models for the statistical calculations.

Since not all of the principal components of the MOS models are Gaussian, we compute the statistical distributions of the intermediate-level parameters in two different ways. The first method is the theoretical approach, using Equations 6.1 and 6.6. The second method is a 10,000-run Monte Carlo analysis using the linear RSM model. The results are summarized in Table 6.5; it is clear that the analytic method and the RSM Monte Carlo method produce almost identical results. The actual distributions obtained from the Monte Carlo analyses are shown in Figures 6.8-6.11. The matrix of the correlation coefficients of the intermediate-level parameters is shown in Figure 6.12.

Parameter	Analytic Calculations		RSM Monte Carlo	
	Nominal	St. Dev.	Nominal	St. Dev.
gain	172.3 MHz/V	2.38%	170.9 MHz/V	2.45%
$f_{0.8}$	38.71 MHz	4.03%	37.34 MHz	4.15%
I_{up}	191.1 μA	2.21%	190.8 μA	2.22%
I_{dn}	191.1 μA	2.20%	190.4 μA	2.16%

Table 6.5: Expected values and standard deviations of intermediate-level parameters.

Parameter	Linear Model		Quadratic Model	
	Accuracy	Worst Error	Accuracy	Worst Error
lock time	50.68%	-3.27%	92.53%	-0.93%

Table 6.6: Comparison of linear and quadratic models for high-level performance.

Parameter	Analytic Calculations		RSM Monte Carlo	
	Nominal	St. Dev.	Nominal	St. Dev.
lock time	7.1642 μs	1.21%	7.1643 μs	1.16%

Table 6.7: Expected values and standard deviations of high-level performance.

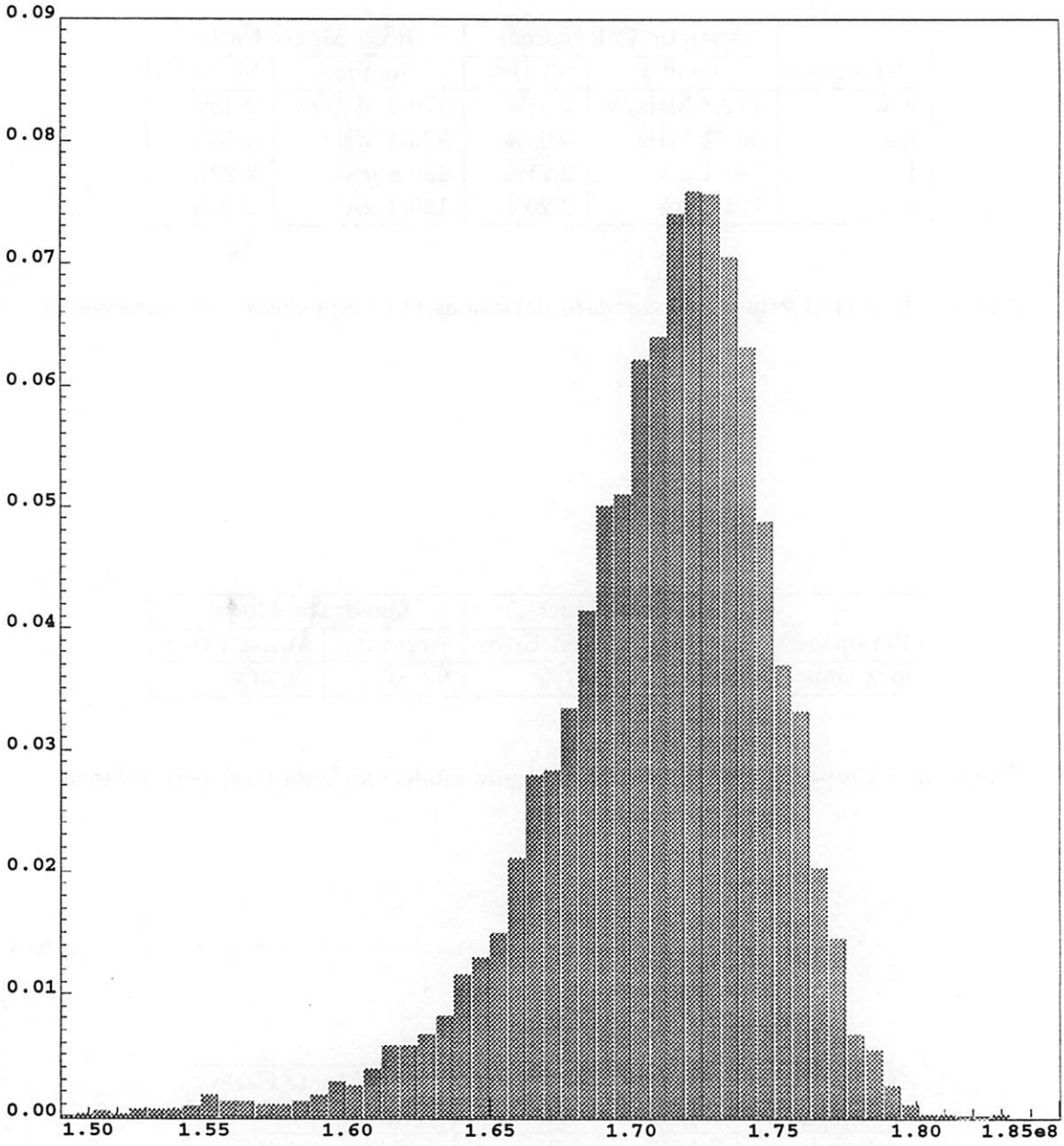


Figure 6.8: Histogram of gain.

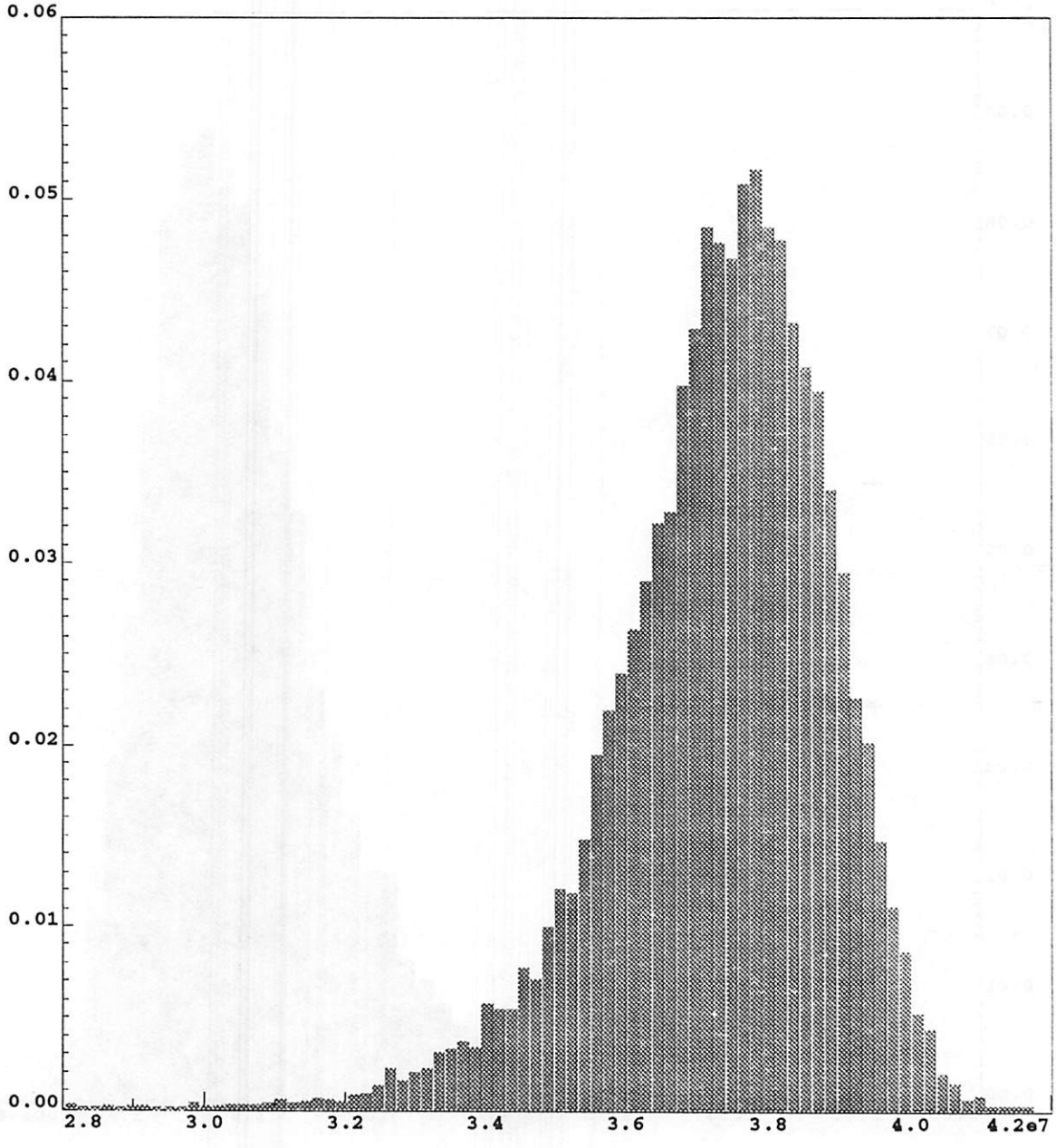


Figure 6.9: Histogram of $f_{0.8}$.

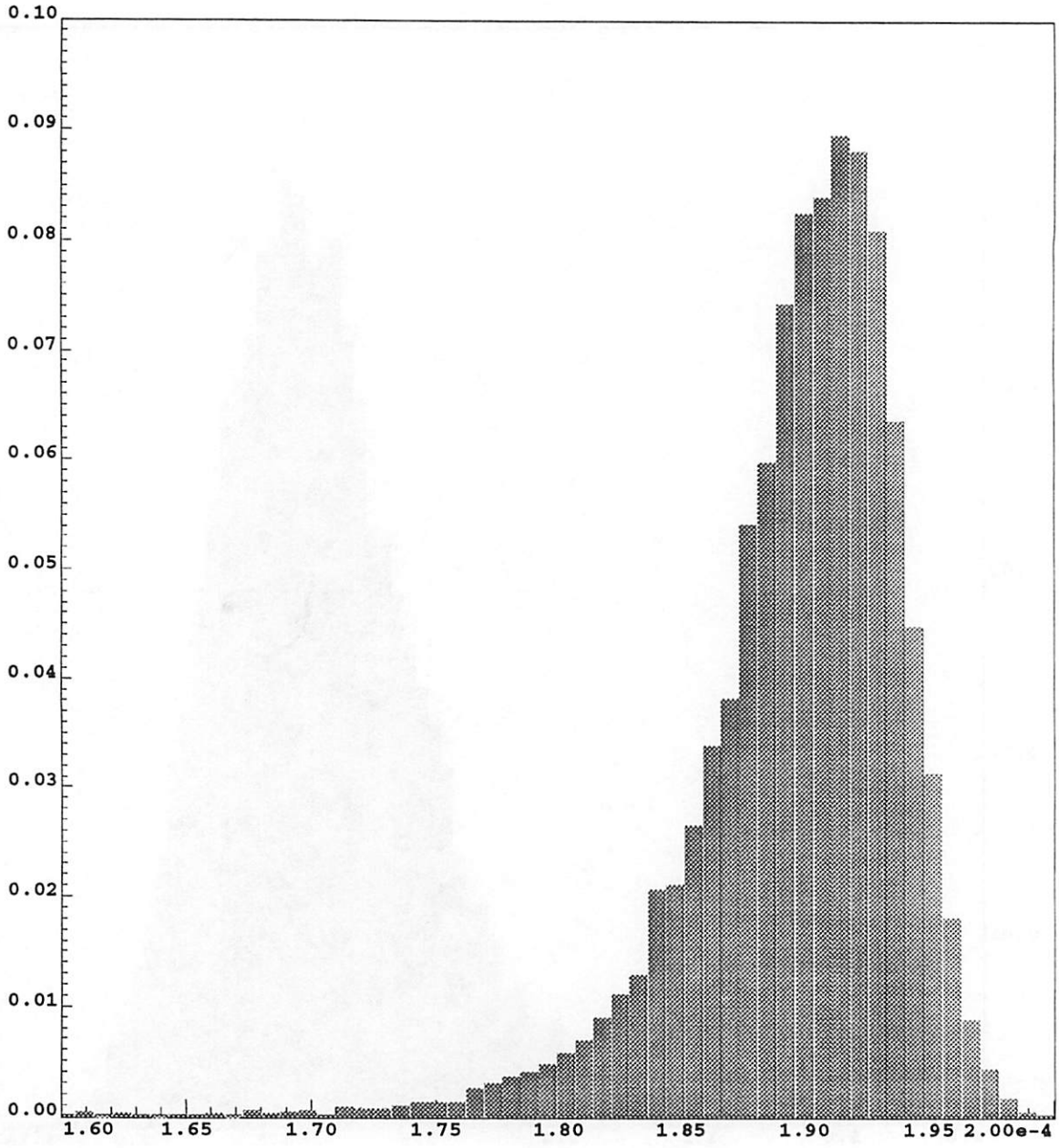


Figure 6.10: Histogram of I_{up}.

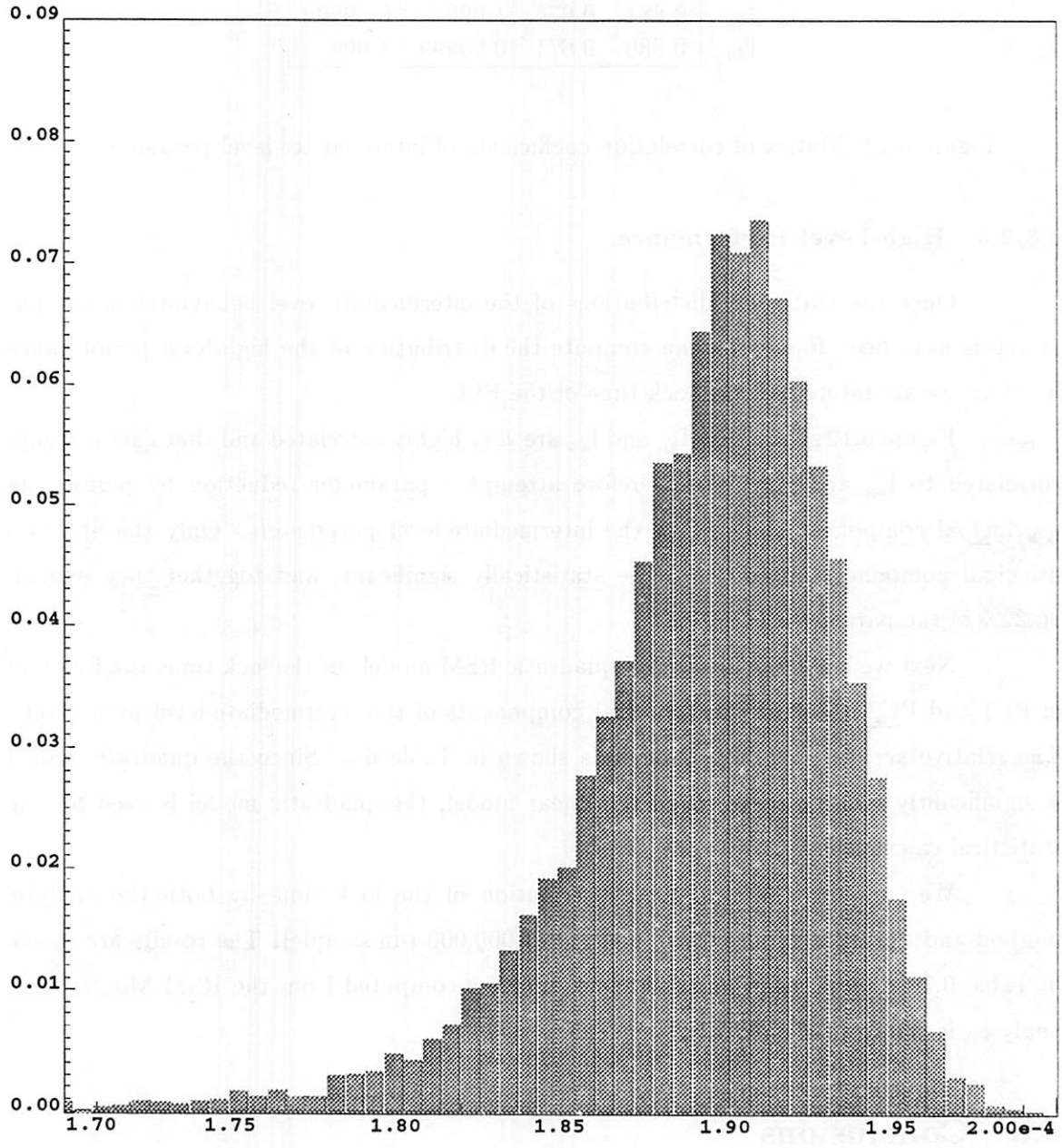


Figure 6.11: Histogram of I_{dn}.

	gain	$f_{0.8}$	I_{up}	I_{dn}
gain	1.000	0.579	0.881	0.880
$f_{0.8}$	0.579	1.000	0.670	0.671
I_{up}	0.881	0.670	1.000	0.99999
I_{dn}	0.880	0.671	0.99999	1.000

Figure 6.12: Matrix of correlation coefficients of intermediate-level parameters.

6.5.2.3 High-Level Performance

Once the statistical distributions of the intermediate-level behavioral model parameters have been found, we can compute the distribution of the high-level performance in which we are interested, the lock time of the PLL.

Figure 6.12 shows that I_{up} and I_{dn} are very highly correlated and that gain is highly correlated to I_{up} and I_{dn}. We therefore attempt a parameter reduction by performing a principal component analysis on the intermediate-level parameters. Only the first two principal components turn out to be statistically significant, and together they explain 96.22% of the parameter variation.

Next we build the linear and quadratic RSM models of the lock time as a function of PC1 and PC2, the first two principal components of the intermediate-level parameters. The relative accuracy of these models is shown in Table 6.6. Since the quadratic model is significantly more accurate than the linear model, the quadratic model is used for the statistical calculations.

We compute the statistical distribution of the lock time by both the analytic method and the RSM Monte Carlo method (1,000,000-run sample). The results are shown in Table 6.7. The distribution of the lock time, as computed from the RSM Monte Carlo analysis, is shown in Figure 6.13.

6.6 Conclusions

We have developed a complete methodology for hierarchical statistical circuit characterization which does not rely upon circuit-level Monte Carlo simulation. The main new ideas are (1) a method for incorporating parameter mismatch and correlation into RSM, (2) a method for directly calculating expected values, variances, and correlations of higher-

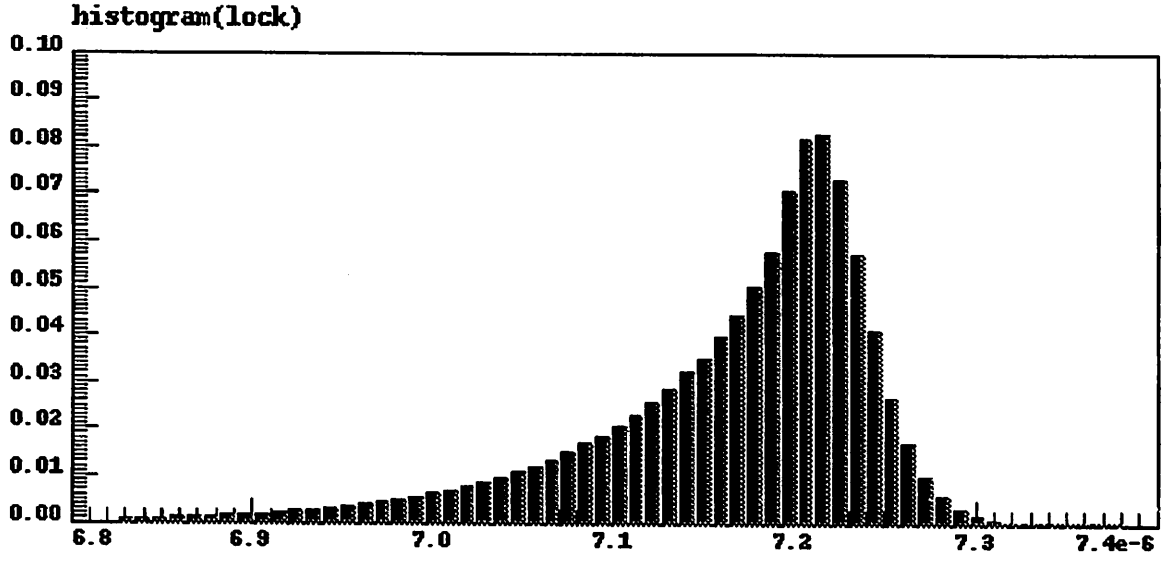


Figure 6.13: Distribution of lock time.

level parameters from those of lower-level parameters, and (3) a method for generating correlated sets of parameters for Monte Carlo analysis at the behavioral level. We have illustrated these ideas on two example circuits, a folded cascode operational amplifier and a phase-locked loop.

One main area of future work is in determining appropriate correlation coefficients to accurately model mismatch; in our example circuits we had to assume perfect matching.

We believe this methodology will be useful for yield analysis, setting realistic circuit specifications for large analog circuits, realistic worst-case modeling for both analog and digital circuits, enforcing matching constraints in constraint-driven place and route, and top-down constraint-driven design.

Chapter 7

Conclusions

Statistical techniques for analog testing and characterization can be used throughout the system design process. Automatic test pattern generation can be used to generate test vectors for production testing and to estimate testing cost during design. Optimal test structures can be designed by applying automatic test pattern generation to competing proposed test structures. And statistical performance characterization can be performed without time-consuming Monte Carlo analyses by using hierarchical modeling and direct statistical techniques.

This systematic statistical approach to the analog testing problem allows time-invariant deterministic systems to be fully characterized as efficiently as possible, eliminating the problems of over-testing, which is wasteful of production resources, and under-testing, which results in poor product quality. Test sets are generated to contain the minimum number of tests required to fully characterize the system performance specifications, considering all relevant catastrophic faults and parametric variations of system parameters. The test set is optimized to minimize the impact of measurement and modeling errors using techniques for optimal design of experiments. During production testing, nonlinear regression is used to generate statistical confidence intervals for determining whether specifications are satisfied or violated.

The result is a testing methodology which impacts analog and mixed-signal electronic products by decreasing testing time and increasing product reliability. With testing currently consuming approximately 30-50% of total product cost, there is tremendous potential impact on the overall cost of designing and manufacturing analog systems.

Bibliography

[1] A.K.B. A'ain, A.H. Bratt and A.P. Dorey, "Testing Analogue Circuits by Power Supply Voltage Control," *Electronics Letters*, vol. 30, n. 3, pp. 214–215, February 1994.

[2] M. Al-Qutayri, "Comparison of Voltage and Current Testing of Analogue and Mixed-Signal Circuits," *Proc. IEEE International ASIC Conference and Exhibit*, pp. 156–159, September 1993.

[3] J.S. Augusto and C.F.B. Almeida, "Fully Automatic DC Fault Dictionary Construction and Test Nodes Selection for Analogue Fault Diagnosis," *Proc. The European Design and Test Conference*, p. 605, March 1995.

[4] A. Balivada, J. Chen and J.A. Abraham, "Analog Testing with Time Response Parameters," *IEEE Design & Test of Computers*, vol. 13, n. 2, pp. 18–25, Summer 1996.

[5] J. Bandler and A. Salama, "Fault Diagnosis of Analog Circuits," *Proc. of the IEEE*, vol. 73, August 1985.

[6] I.M. Bell, D.A. Camplin, G.E. Taylor and B.R. Bannister, "Supply Current Testing of Mixed Analogue and Digital ICs," *Electronics Letters*, vol. 27, n. 17, pp. 1581–1583, 1991.

[7] R.S. Berkowitz, "Conditions for Network-Element-Value Solvability," *IEEE Trans. on Circuit Theory*, vol. CT-9, pp. 24–29, 1962.

[8] J.L. Bernier, J.J. Merelo, J. Ortega and A. Prieto, "Test Pattern Generation for Analog Circuits Using Neural Networks and Evolutive Algorithms," *Proc. International Workshop on Artificial Neural Networks*, pp. 838–844, June 1995.

[9] L. Bonet, J. Ganger, J. Girardeu, C. Greaves, M. Pendelton and D. Yatim, "Test Features of the MC145472 ISDN U-Transceiver," *Proc. International Test Conference*, pp. 68–79, 1990.

[10] G.E.P. Box and N.R. Draper, *Empirical Model-Building and Response Surfaces*, J. Wiley & Sons, New York, 1987.

[11] A.H. Bratt, A.M.D. Richardson, R.J.A. Harvey and A.P. Dorey, "A Design-for-Test Structure for Optimising Analogue and Mixed Signal IC Test," *Proc. European Design and Test Conference*, pp. 24–33, March 1995.

[12] T. Butzerin, A. Samad and E. Archambeau, "ASIC Testing with High Fault Coverage," *VLSI Systems Design*, vol. 9, n. 9, pp. 50–57, September 1988.

[13] R.L. Campbell and R.A. Tarbox, "Testing Goes Critical Path," *AT&T Technical Journal*, vol. 73, n. 2, pp. 4–9, March-April 1994.

[14] D.A. Camplin, I.M. Bell, G.E. Taylor and B.R. Bannister, "Can Supply Current Monitoring be Applied to the Testing of Analogue as Well as Digital Portions of Mixed ASICs?," *Proc. The European Conference on Design Automation*, pp. 538–542, March 1992.

[15] H. Chang, E. Charbon, U. Choudhury, A. Demir, E. Felt, E. Liu, E. Malavasi, A. Sangiovanni-Vincentelli and I. Vassiliou, *A Top-Down, Constraint-Driven Design Methodology for Analog Integrated Circuits*, Kluwer Academic Publishers, Boston, MA, 1997.

[16] H. Chang, E. Felt and A.L. Sangiovanni-Vincentelli, "Top-Down, Constraint-Driven Design Methodology Based Generation of a Second Order Σ-Δ A/D Converter," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 533–536, May 1995.

[17] H. Chang, E. Liu, R. Neff, E. Felt, E. Malavasi, E. Charbon, A. Sangiovanni-Vincentelli and P.R. Gray, "Top-Down, Constraint-Driven Methodology Based Generation of n-bit Interpolative Current Source D/A Converters," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 369–372, May 1994.

[18] H. Chang, A. Sangiovanni-Vincentelli, E. Charbon, U. Choudhury, E. Felt, G. Jusuf, E. Liu, E. Malavasi and R. Neff, "A Top-Down, Constraint-Driven Design Methodol-

ogy for Analog Integrated Circuits," *Proc. Workshop on "Advances in Analog Circuit Design," Scheveningen, NL*, pp. 301–325, April 1992.

[19] C.-Y. Chao, H.-J. Lin and L. Milor, "Fault-Driven Testing of LSI Analog Circuits," *Midwest Symposium on Circuits and Systems*, vol. 2, pp. 927–930, August 1992.

[20] A. Chatterjee, B.C. Kim and N. Nagi, "DC Built-In Self-Test for Linear Analog Circuits," *IEEE Design & Test of Computers*, vol. 13, n. 2, pp. 26–33, Summer 1996.

[21] K.R. Chin, "Functional Testing of Circuits and SMD Boards with Limited Nodal Access," *Proc. IEEE International Test Conference*, pp. 129–143, August 1989.

[22] F. Corsi, M. Chiarantoni, R. Lorusso and C. Marzocca, "A Fault Signature Approach to Analog Devices Testing," *Proc. European Test Conference*, pp. 116–121, April 1993.

[23] S.B. Crary, "Optimal Design of Experiments for Sensor Calibration," *Proc. IEEE International Conference on Solid-State Sensors and Actuators*, pp. 404–407, 1991.

[24] J. Machado da Silva, J. Silva Matos, I.M. Bell and G.E. Taylor, "Cross-correlation Between i_{dd} and v_{out} Signals for Testing Analogue Circuits," *Electronics Letters*, vol. 31, n. 19, pp. 1617–1618, September 1995.

[25] H. Dai and T.M. Sounders, "Time-Domain Testing Strategies and Fault Diagnosis for Analog Systems," *IEEE Trans. on Instrumentation and Measurement*, vol. 39, n. 1, pp. 157–162, February 1990.

[26] D.W. Daugherty, "Using Fault Simulation to Increase Productivity," *Electronics Test*, vol. 11, n. 9, pp. 22–24, September 1988.

[27] G. Devarayanadurg and M. Soma, "Analytic Fault Modeling and Static Test Generation for Analog ICs," *Proc. IEEE International Conference on Computer-Aided Design*, pp. 44–47, November 1994.

[28] G. Devarayanadurg and M. Soma, "Dynamic Test Signal Design for Analog ICs," *Proc. IEEE International Conference on Computer-Aided Design*, pp. 627-630, November 1995.

[29] K. Dubrowski and T. Wong, "Mixed Analog/Digital ASICs," *VLSI Systems Design*, vol. 9, n. 9, pp. 60–65, September 1988.

[30] P. Duhamel and J.C. Rault, "Automatic Test Generation Techniques for Analog Circuits and Systems: A Review," *IEEE Trans. on Circuits and Systems*, vol. CAS-26, n. 7, pp. 411–440, July 1979.

[31] *ELDO User's Manual*, ANACAD Electrical Engineering Software, ULM (Donau) Germany, 1995.

[32] M. Fares and B. Kaminska, "Fuzzy Optimization Models for Analog Test Decisions," *Journal of Electronic Testing: Theory and Applications*, vol. 5, n. 2-3, pp. 299–305, May-August 1994.

[33] P. Fasang, "Analog/Digital ASIC Design for Testability," *IEEE Trans. on Industrial Electronics*, vol. 36, n. 2, pp. 219–226, May 1989.

[34] P.P. Fasang, D. Mullins and T. Wong, "Design for Testability for Mixed Analog/Digital ASICs," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 16.5.1–16.5.4, May 1988.

[35] E. Felt, A. Narayan and A. Sangiovanni-Vincentelli, "Measurement and Modeling of MOS Transistor Current Mismatch in Analog ICs," *Proc. IEEE International Conference on Computer-Aided Design*, pp. 272–277, November 1994.

[36] E. Felt and A. Sangiovanni-Vincentelli, "Testing of Analog Systems Using Behavioral Models and Optimal Experimental Design Techniques," *Proc. IEEE International Conference on Computer-Aided Design*, pp. 672–678, November 1994.

[37] E. Felt and A.L. Sangiovanni-Vincentelli, "Optimal Design of Analog IC Test Structures," *Proc. IEEE International Mixed Signal Testing Workshop*, June 1995.

[38] E. Felt and A.L. Sangiovanni-Vincentelli, "Optimization of Analog IC Test Structures," *Proc. IEEE VLSI Test Symposium*, pp. 48–53, April-May 1996.

[39] E. Felt, S. Zanella, C. Guardiani and A.L. Sangiovanni-Vincentelli, "Hierarchical Statistical Characterization of Mixed-Signal Circuits Using Behavioral Modeling," *Proc. IEEE International Conference on Computer-Aided Design*, pp. 374–380, November 1996.

[40] R.R. Fritzemeier, J.M. Soden, R.K. Treece and C.F. Hawkins, "Increased CMOS IC Stuck-At Fault Coverage with Reduced I_{DDQ} Test Set," *Proc. IEEE International Test Conference*, pp. 427–435, September 1990.

[41] A. Ghosh, S. Devadas and A. Richard Newton, *Sequential Logic Testing and Verification*, Kluwer Academic Publishers, Boston, MA, 1992.

[42] G. Gielen, Z. Wang and W. Sansen, "Fault Detection and Input Stimulus Determination for the Testing of Analog Integrated Circuits Based on Power-Supply Current Monitoring," *Proc. IEEE International Conference on Computer-Aided Design*, pp. 495–498, November 1994.

[43] N.B. Hamida and B. Kaminska, "Analog Circuit Testing Based on Sensitivity Computation and New Circuit Modeling," *Proc. IEEE International Test Conference*, pp. 652–661, October 1993.

[44] N.B. Hamida and B. Kaminska, "Multiple Fault Analog Circuit Testing by Sensitivity Analysis," *Journal of Electronic Testing: Theory and Applications*, vol. 4, n. 3, pp. 331–343, November 1993.

[45] N.B. Hamida and B. Kaminska, "Multiple Fault Testing in Analog Circuits," *Proc. International Conference on VLSI Design*, pp. 61–66, January 1994.

[46] T. Hamilton and B. Hadley, "Solving the Simulation-to-Test Translation Problem," *Electronics Test*, vol. 11, n. 9, pp. 47–51, September 1988.

[47] J.M. Hammersley and D.C. Handscomb, *Monte Carlo Methods*, London, Methuen & Co Ltd., 1964.

[48] R.H. Hardin and N.J.A. Sloane, "A New Approach to the Construction of Optimal Designs," *Journal of Statistical Planning and Inference*, vol. 37, pp. 339–369, 1993.

[49] R.H. Hardin and N.J.A. Sloane, *Operating Manual for* GOSSET*: A General-Purpose Program for Constructing Experimental Designs (Second Edition)*, Mathematical Sciences Research Center, AT&T Bell Laboratories, 1994.

[50] A.A. Hatzopoulos, S. Siskos and J.M. Kontoleon, "A Complete Scheme of Built-In Self-Tests (BIST) Structure for Fault Diagnosis in Analog Circuits and Systems,"

IEEE Trans. on Instrumentation and Measurement, vol. 42, n. 3, pp. 689–694, June 1993.

[51] *HDLA User's Manual*, ANACAD Electrical Engineering Software, ULM (Donau) Germany, 1995.

[52] G. Hemink, B. Meijer and H. Kerkhoff, "Testability Analysis of Analog Systems," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. CAD-9, pp. 573–583, June 1990.

[53] G.J. Hemink, B.W. Meijer and H.G. Kerkhoff, "TASTE: A Tool for Analog System Testability Evaluation," *Proc. IEEE International Test Conference*, pp. 829–838, September 1988.

[54] R. Hooke and T.A. Jeeves, "'Direct Search' Solution of Numerical and Statistical Problems," *Journal of ACM*, vol. 8, pp. 212–229, 1961.

[55] J.E. Jackson, *A User's Guide to Principal Components*, John Wiley & Sons, New York, 1991.

[56] M. Jarwala and S.J. Tsai, "A Framework for Design for Testability of Mixed Analog/Digital Circuits," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 13.5.1–13.5.4, May 1991.

[57] B. Johnson, T. Quarles, A.R. Newton, D.O. Pederson and A. Sangiovanni-Vincentelli, *SPICE3 Version 3f User's Manual*, University of California, Berkeley, 1992.

[58] W.J. Kennedy, Jr. and J.E. Gentle, *Statistical Computing*, Marcel Dekker, Inc., New York, 1980.

[59] B. Kaminska and B. Courtois, "Guest Editors' Introduction: Mixed Analog and Digital Systems," *IEEE Design & Test of Computers*, vol. 13, n. 2, pp. 8–9, Summer 1996.

[60] J. Kiefer, *Collected Papers III: Design of Experiments*, Springer-Verlag, New York, 1985.

[61] F. Kuhns, "Automating Testability Analysis of Analog Circuits and Systems," *AUTOTESTCON: The IEEE Systems Readiness Technology Conference*, pp. 225–231, September 1992.

[62] D.P. Kwok and Q. Lu, "On the Testability of Analog Circuits with Directive Faults: A Rule-Based Solution and an Equation Check Approach," *International Conference on Industrial Electronics, Control, Instrumentation, and Automation. Power Electronics and Motion Control*, pp. 1556–1561, November 1992.

[63] K.-J. Lee, T.-P. Lee, R.-C. Wen and Z.-Y. Lin, "Analogue Boundary Scan Architecture for DC and AC Testing," *Electronics Letters*, vol. 32, n. 8, pp. 704–705, April 1996.

[64] D.M.W. Leenaerts and J. van Spaandonk, "DC Testing of Analog Integrated Circuits with Piecewise Linear Approximation and Interval Analysis," *Proc. IEEE International Symposium on Circuits and Systems*, pp. 1337–1340, May 1993.

[65] A. Liberatore, S. Manetti and M.C. Piccirilli, "A New Efficient Method for Analog Circuit Testability Measurement," *IEEE Instrumentation and Measurement Technology Conference*, pp. 193–196, May 1994.

[66] W.M. Lindermeir, H.E. Graeb and K.J. Antreich, "Design Based Analog Testing by Characteristic Observation Inference," *Proc. IEEE International Conference on Computer-Aided Design*, pp. 620–626, November 1995.

[67] E. Liu, W. Kao, E. Felt and A.L. Sangiovanni-Vincentelli, "Analog Testability Analysis and Fault Diagnosis using Behavioral Modeling," *Proc. IEEE Custom Integrated Circuits Conference*, 1994.

[68] E. Liu, A. Sangiovanni-Vincentelli, G. Gielen and P. Gray, "A Behavioral Representation for Nyquist Rate A/D Converters," *Proc. IEEE International Conference on Computer-Aided Design*, pp. 386–389, November 1991.

[69] P.V. Lopresti, "Extending Design-for-Test into the Analog and Mixed-Signal Domains," *AT&T Technical Journal*, vol. 73, n. 2, pp. 49–55, March-April 1994.

[70] W. Maly, "Realistic Fault Modeling for VLSI Testing," *Proc. ACM/IEEE Design Automation Conference*, pp. 173–180, June 1987.

[71] W. Mao, Y. Lu, R.K. Gulati and R. Danapani, "Test Generation for Linear Analog Circuits," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 521–524, May 1995.

[72] M.J. Marlett and J.A. Abraham, "DC-IATP: An Iterative Analog Circuit Test Generation Program for Generating DC Single Pattern Tests," *Proc. IEEE International Test Conference*, pp. 839–845, October 1988.

[73] A. McKeon and A. Wakeling, "Fault Diagnosis in Analogue Circuits Using AI Techniques," *Proc. IEEE International Test Conference*, pp. 118–123, August 1989.

[74] J. McLeod, "Growth Spurt Opens the Door for US VLSI Test System Makers," *Electronics*, pp. 89–90, September 1988.

[75] A. Meixner and W. Maly, "Fault Modeling for the Testing of Mixed Integrated Circuits," *Proc. IEEE International Test Conference*, pp. 564–572, October 1991.

[76] L. Milor and A.L. Sangiovanni-Vincentelli, "Optimal Test Set Design for Analog Circuits," *Proc. IEEE International Conference on Computer-Aided Design*, pp. 294–297, November 1990.

[77] L. Milor and A.L. Sangiovanni-Vincentelli, "Minimizing Production Test Time to Detect Faults in Analog Circuits," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, n. 6, pp. 796–813, June 1994.

[78] L. Milor and V. Visvanathan, "Detection of Catastrophic Faults in Analog Integrated Circuits," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. CAD-8, n. 2, pp. 114–130, February 1989.

[79] S. Mir, V. Kolarik, M. Lubaszewski, C. Nielsen and B. Courtois, "Built-in Self-Test and Fault Diagnosis of Fully Differential Analogue Circuits," *Proc. IEEE International Conference on Computer-Aided Design*, pp. 486–490, November 1994.

[80] S. Mir, M. Lubaszewski, V. Kolarik and B. Courtois, "Automatic Test Generation for Maximal Diagnosis of Linear Analog Circuits," *Proc. European Design and Test Conference*, pp. 254–258, March 1996.

[81] S. Mir, M. Lubaszewski, V. Liberali and B. Courtois, "Built-In Self-Test Approaches for Analogue and Mixed-Signal Integrated Circuits," *Midwest Symposium on Circuits and Systems*, pp. 1145–1150, August 1995.

[82] Y. Miura, "Real-Time Current Testing for A/D Converters," *IEEE Design & Test of Computers*, vol. 13, n. 2, pp. 34–41, Summer 1996.

[83] R.H. Myers and D.C. Montgomery, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, New York, 1995.

[84] N. Nagi and J.A. Abraham, "Hierarchical Fault Modeling for Analog and Mixed-Signal Circuits," *Proc. IEEE VLSI Test Symposium*, pp. 96–101, 1992.

[85] N. Nagi and J.A. Abraham, "Fault-Based Automatic Test Generator for Linear Analog Circuits," *Proc. IEEE International Conference on Computer-Aided Design*, pp. 88–91, November 1993.

[86] R. Naiknaware, G.N. Nandakumar and S.R. Kasa, "Automatic Test Plan Generation for Analog and Mixed Signal Integrated Circuits using Partial Activation and High Level Simulation," *Proc. IEEE International Test Conference*, pp. 139–148, October 1993.

[87] *National Technology Road Map for Semiconductors*, Semiconductor Industry Association, San Jose, CA, 1994.

[88] M.S. Nejad, L. Sebaa, A. Ladick and F. Kuo, "Analog Built-In Self-Test," *Proc. IEEE International ASIC Conference and Exhibit*, pp. 407–411, September 1994.

[89] N.K. Nguyen and A.J. Miller, "A Review of Some Exchange Algorithms for Constructing Discrete D-Optimal Designs," *Computational Statistics and Data Analysis*, vol. 14, pp. 489–498, 1992.

[90] P. Nigh and W. Maly, "Test Generation for Current Testing (CMOS ICs)," *IEEE Design & Test of Computers*, vol. 7, n. 1, pp. 26–38, February 1990.

[91] J. Novellino, "Software Generates ASIC Test Programs," *Electronic Design*, vol. 37, n. 9, pp. 59–62, April 1989.

[92] J. Novellino, "Software Creates Analog, Mixed-Signal IC Tests," *Electronic Design*, vol. 40, n. 10, pp. 104–106, May 1992.

[93] T. Olbrich and A. Richardson, "Design and Self-Test for Switched-Current Building Blocks," *IEEE Design & Test of Computers*, vol. 13, n. 2, pp. 10–17, Summer 1996.

[94] D. Organ, "enVision: The Inside Story," *Proc. IEEE International Test Conference*, pp. 530–536, September 1990.

[95] D.K. Papakostas and A.A. Hatzopoulos, "Correlation-Based Comparison of Analog Signatures for Identification and Fault Analysis," *IEEE Trans. on Instrumentation and Measurement*, vol. 42, n. 4, pp. 860–863, August 1993.

[96] D.K. Papakostas and A.A. Hatzopoulos, "Supply Current Testing in Linear Bipolar ICs," *Electronics Letters*, vol. 30, n. 2, pp. 128–130, 1994.

[97] C.M. Pieper, "The Myth and Reality of Linking Design and Test," *Electronics Test*, vol. 11, n. 4, pp. 55–61, April 1988.

[98] V.C. Prasad and N.S.C. Babu, "On Minimal Set of Test Nodes for Fault Dictionary of Analog Circuit Fault Diagnosis," *Journal of Electronic Testing: Theory and Applications*, vol. 7, n. 3, pp. 255–258, December 1995.

[99] V.C. Prasad and S.N. Rao Pinjala, "Fast Algorithms for Selection of Test Nodes of an Analog Circuit Using a Generalized Fault Dictionary Approach," *Circuits, Systems, and Signal Processing*, vol. 14, n. 6, pp. 707–724, 1995.

[100] V.C. Prasad and S.N.R. Pinjala, "Boolean Method for Selection of Minimal Set of Test Nodes for Analogue Fault Dictionary," *Electronics Letters*, vol. 29, n. 9, pp. 747–749, April 1993.

[101] R. Rastogi and K. Sierzega, "A New Approach to Mixed-Signal Diagnosis," *Proc. IEEE International Test Conference*, pp. 591–597, September 1990.

[102] A. Richardson, T. Olbrich, V. Liberali and F. Maloberti, "Design-for-Test Strategies for Analogue and Mixed-Signal Integrated Circuits," *Midwest Symposium on Circuits and Systems*, pp. 1139–1144, August 1995.

[103] M. Robson and G. Russel, "Current Monitoring Technique for Testing Embedded Analogue Functions in Mixed Signal ICs," *Electronics Letters*, vol. 32, n. 9, pp. 796–798, April 1996.

[104] R.Y. Rubinstein, *Simulation and the Monte Carlo Method*, John Wiley & Sons, 1981.

[105] S. Runyon, "There's Plenty of Activity in the Mixed-Signal Arena," *High Performance Systems*, pp. 47–49, August 1989.

[106] G. Russell and D.S. Learmonth, "Systematic Approaches to Testing Embedded Analogue Circuit Functions," *Microelectronics Journal*, vol. 25, n. 2, pp. 133–138, March 1994.

[107] M. Sachdev, "A Realistic Defect Oriented Testability Methodology for Analog Circuits," *Journal of Electronic Testing: Theory and Applications*, vol. 6, n. 3, pp. 265–276, June 1995.

[108] R. Saeks, "A Measure of Testability and its Application to Test Point Selection Theory," *Midwest Symposium on Circuits and Systems*, August 1977.

[109] R. Saeks, A. Sangiovanni-Vincentelli and V. Visvanathan, "Diagnosability of Nonlinear Circuits and Systems—Part II: Dynamical Systems," *IEEE Trans. on Circuits and Systems*, vol. CAS-28, pp. 1103–1108, November 1981.

[110] A.E. Salama, "A Unified Decomposition Approach for Fault Location in Large Analog Circuits," *IEEE Trans. on Circuits and Systems*, vol. CAS-31, n. 7, pp. 609–622, July 1984.

[111] H. Schreiber, "Fault Dictionary Based Upon Stimulus Design," *IEEE Trans. on Circuits and Systems*, vol. CAS-26, pp. 529–537, July 1979.

[112] M.H. Schultz, E. Trischler and T.M. Sarfert, "SOCRATES: A Highly Efficient Automatic Test Pattern Generation System," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, n. 1, pp. 126–137, 1988.

[113] S.R. Searle, *Linear Models*, John Wiley & Sons, New York, 1971.

[114] G.A.F. Seber, *Linear Regression Analysis*, John Wiley & Sons, New York, 1977.

[115] N. Sen and R. Saeks, "Fault Diagnosis for Linear Systems via Multifrequency Measurements," *IEEE Trans. on Circuits and Systems*, vol. CAS-26, 1979.

[116] J.P. Shen, W. Maly and F.J. Ferguson, "Inductive Fault Analysis of MOS Integrated Circuits," *IEEE Design & Test of Computers*, vol. 2, n. 6, pp. 13–26, December 1985.

[117] K. Sierzega and R. Rasogi, "Model-based Reasoning Finds Analog PCB Faults," *Electronics Test*, vol. 13, n. 3, pp. 26–30, March 1990.

[118] *SimPilot User's Manual*, ANACAD Electrical Engineering Software, ULM (Donau) Germany, 1995.

[119] M. Slamani and B. Kaminska, "Analog Circuit Fault Diagnosis Based on Sensitivity Computation and Functional Testing," *IEEE Design & Test of Computers*, vol. 9, n. 1, pp. 30–39, March 1992.

[120] M. Slamani and B. Kaminska, "Testing Analog Circuits by Sensitivity Computation," *Proc. The European Conference on Design Automation*, pp. 532–537, March 1992.

[121] M. Slamani and B. Kaminska, "Multifrequency Testability Analysis for Analog Circuits," *Proc. IEEE VLSI Test Symposium*, pp. 54–59, April 1994.

[122] E.A. Sloan, "Transfer Function Estimation, Part I, Theoretical and Practical Considerations (Testing Application)," *Proc. IEEE International Test Conference*, pp. 426–439, October 1984.

[123] M. Soma, "A Design-for-Test Methodology for Active Analog Filters," *Proc. IEEE International Test Conference*, pp. 183–192, September 1990.

[124] M. Soma, "An Experimental Approach to Analog Fault Models," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 13.6.1–13.6.4, May 1991.

[125] M. Soma, "Fault Coverage of DC Parametric Tests for Embedded Analog Amplifiers," *Proc. IEEE International Test Conference*, pp. 566–573, October 1993.

[126] M. Soma, "Structure and Concepts for Current-Based Analog Scan," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 517–520, May 1995.

[127] M. Soma and V. Kolarik, "A Design-for-Test Technique for Switched-Capacitor Filters," *Proc. IEEE VLSI Test Symposium*, pp. 42–47, April 1994.

[128] T. Souders and G. Stenbakken, "A Comprehensive Approach for Modeling and Testing Analog and Mixed-Signal Devices," *IEEE International Test Conference*, pp. 169–176, September 1990.

[129] T. Souders and G. Stenbakken, "Cutting the High Cost of Testing," *IEEE Spectrum*, pp. 48–51, March 1991.

[130] *SPAYN User's Manual*, Silvaco International, Santa Clara, California, 1995.

[131] J.A. Starzyk and H. Dai, "A Decomposition Approach for Testing Large Analog Networks," *Journal of Electronic Testing: Theory and Applications*, vol. 3, n. 3, pp. 181–195, August 1992.

[132] G. Stenbakken and T. Souders, "Test-Point Selection and Testability Measures via QR Factorization of Linear Models," *IEEE Trans. on Instrumentation and Measurement*, June 1987.

[133] G. Stenbakken, T. Souders and G. Stewart, "Ambiguity Groups and Testability," *IEEE Trans. on Instrumentation and Measurement*, vol. 38, October 1989.

[134] G.N. Stenbakken and T.M. Souders, "Linear Error Modeling of Analog and Mixed-Signal Devices," *Proc. IEEE International Test Conference*, pp. 573–581, 1991.

[135] D. Taylor, P.S.A. Evans and T.I. Pritchard, "Testing for Functional Defects in Embedded Digital-to-Analogue Converters Using Dynamic Stimuli and Transient Response Analysis," *Microelectronics Journal*, vol. 25, n. 6, pp. 415–424, September 1994.

[136] G. Temes, "Large-Change Sensitivities for Linear Digital Networks," *Midwest Symposium on Circuits and Systems*, August 1977.

[137] M.F. Toner and G.W. Roberts, "A BIST Scheme for an SNR Test of a Sigma-Delta ADC," *Proc. IEEE International Test Conference*, pp. 805–814, October 1993.

[138] M.F. Toner and G.W. Roberts, "A BIST Technique for a Frequency Response and Intermodulation Distortion Test of a Sigma-Delta ADC," *Proc. IEEE VLSI Test Symposium*, pp. 60–65, April 1994.

[139] M.F. Toner and G.W. Roberts, "A BIST Scheme for a SNR, Gain Tracking, and Frequency Response Test of a Sigma-Delta ADC," *IEEE Trans. on Circuits and*

Systems II: Analog and Digital Signal Processing, vol. 42, n. 1, pp. 1–15, January 1995.

[140] M.F. Toner and G.W. Roberts, "On the Practical Implementation of Mixed Analog-Digital BIST," *Proc. IEEE Custom Integrated Circuits Conference*, pp. 525–528, May 1995.

[141] M.F. Toner and G.W. Roberts, "A Frequency Response, Harmonic Distortion, and Intermodulation Distortion Test for BIST of a Sigma-Delta ADC," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, n. 8, pp. 608–613, August 1996.

[142] S.-J. Tsai, "Test Vector Generation for Linear Analog Devices," *Proc. IEEE International Test Conference*, pp. 592–597, October 1991.

[143] *UTMOST User's Manual*, Silvaco International, Santa Clara, California, 1995.

[144] R.J. van Rijsinge, A.A.R.M. Haggenburg, C. de Vries and H. Wallinga, "From Specification to Measurement: The Bottleneck in Analog Industrial Testing," *Proc. IEEE International Test Conference*, pp. 177–182, September 1990.

[145] V. Visvanathan and A. Sangiovanni-Vincentelli, "A Computational Approach for the Diagnosability of Dynamical Circuits," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. CAD-3, pp. 165–171, July 1984.

[146] A. Vladimirescu, A.R. Newton and D.O. Pederson, *SPICE Version 2G.1 User's Guide*, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, 1980.

[147] K.D. Wagner and T.W. Williams, "Design for Testability of Mixed Signal Integrated Circuits," *Proc. IEEE International Test Conference*, pp. 823–828, September 1988.

[148] D.M.H. Walker, *Yield Simulation for Integrated Circuits*, Kluwer Academic Publishers, Boston, MA, 1987.

[149] Z. Wang, G. Gielen and W. Sansen, "Testing of Analog Integrated Circuits Based on Power-Supply Current Monitoring and Discrimination Analysis," *Proc. Asian Test Symposium*, pp. 126–131, November 1994.

[150] C.-L. Wey, "Built-In Self-Test (BIST) Structure for Analog Circuit Fault Diagnosis," *IEEE Trans. on Instrumentation and Measurement*, vol. 39, n. 3, pp. 517–521, June 1990.

[151] C.-L. Wey and S. Krishnan, "Built-In Self-Test (BIST) Structure for Analog Circuit Fault Diagnosis with Current Test Data," *IEEE Trans. on Instrumentation and Measurement*, vol. 41, n. 4, pp. 535–539, August 1992.

[152] C.-L. Wey and R. Saeks, "On the Implementation of an Analog ATPG: the Linear Case," *IEEE Trans. on Instrumentation and Measurement*, vol. IM-34, n. 3, pp. 442–449, September 1985.

[153] C.-L. Wey and R. Saeks, "On the Implementation of an Analog ATPG: the Nonlinear Case," *IEEE Trans. on Instrumentation and Measurement*, vol. 37, n. 2, pp. 252–258, June 1988.

[154] T.D. Wickens, *The Geometry of Multivariate Statistics*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1995.

[155] L.A. Williams and B.A. Wooley, "MIDAS—A Functional Simulator for Mixed Digital and Analog Sampled Data Systems," *Proc. IEEE International Symposium on Circuits and Systems*, vol. 5, pp. 2148–2151, 1992.

[156] T.W. Williams and K.P. Parker, "Design for Testability—A Survey," *Proc. of the IEEE*, vol. 71, n. 1, pp. 98–112, January 1983.

[157] B.W. Woodall, B.D. Newman and A.G. Saumuli, "Empirical Results of Undetected Stuck-Open Failures," *Proc. IEEE International Test Conference*, pp. 166–170, September 1987.

[158] P. Yang, D. Hocevar, P. Cox, C. Machala and P. Chatterjee, "An Integrated and Efficient Approach for MOS VLSI Statistical Circuit Design," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. CAD-5, pp. 5–14, January 1986.