

A test method for analog circuits : using sensitivity analysis and the singular value decomposition

Citation for published version (APA):

Spaandonk, van, J. (1996). *A test method for analog circuits : using sensitivity analysis and the singular value decomposition*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Electrical Engineering]. Technische Universiteit Eindhoven. <https://doi.org/10.6100/IR463325>

DOI:

[10.6100/IR463325](https://doi.org/10.6100/IR463325)

Document status and date:

Published: 01/01/1996

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

A Test Method for Analog Circuits

using Sensitivity Analysis
and the Singular Value Decomposition

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de Rector Magnificus, prof. dr. M. Rem, voor een commissie aangewezen door het College van Dekanen in het openbaar te verdedigen op donderdag 5 september 1996 om 16.00 uur

door

Johannes van Spaandonk

geboren te Goirle

Dit proefschrift is goedgekeurd
door de promotoren

prof. dr. ir. W.M.G. van Bokhoven
prof. dr. ir. R.H.J.M. Otten

en de copromotor

dr. ir. D.M.W. Leenaerts

© Copyright 1996 J. van Spaandonk.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the copyright owner.

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Spaandonk, Johannes van

A test method for analog circuits : using sensitivity analysis and the singular value decomposition / by Johannes van Spaandonk. - Eindhoven :

Technische Universiteit Eindhoven, 1996. - XVI, 162 p.

Proefschrift. - ISBN 90-386-0160-3

NUGI 832

Trefw.: analoge schakelingen / geïntegreerde schakelingen ; testen /
automatische testmethoden / elektronische foutendiagnose.

aan Antoinette

Summary

This thesis presents a method for testing analog circuits that is useful for diagnostic testing as well as verification of high-level circuit behavior. The method is based on a first-order approximation (i.e., a linear model) of non linear circuit behavior. Therefore it is applicable if deviations in the circuit parameters (caused by manufacturing process deviations) are relatively small, which is often the case in analog circuits.

Linearized circuit behavior is described by a model matrix, which is obtained by computing first-order parameter sensitivities, for example. A complete orthogonal decomposition of the model matrix is found by computing its Singular Value Decomposition (SVD). This decomposition forms the basis of the methods presented in this thesis.

A circuit's limited accessibility for measurements causes redundant factors to be present in the mentioned linear model. These are eliminated by retaining only the most important orthogonal components of the SVD of the model matrix. A direct method is presented that selects one measurement for each of these components. To determine the number of necessary measurements, the accessibility for measurements as well as the magnitude of random measurement errors are taken into account. The direct approach may also be used to select testable circuit components, to analyze a circuit's testability.

In addition, an iterative algorithm is presented that tries to select measurements that minimize the influence of random measurement errors. The algorithm is based on the

D-optimality criterion. The performance of the iterative algorithm is better than that of the direct approach. This is mainly due to the fact that it optimizes a *complete* set of measurements. In contrast, the mentioned direct approach selects measurements one by one, each time optimizing only the last selected measurement. The iterative algorithm also adjusts the *number* of test points, taking into account the expected magnitude of random measurement errors. An example shows that the iterative method also obtains better results than an existing method which is based on the QR decomposition.

The discussed concepts were implemented in a computer program, which can perform a testability analysis and/or verify the functional behavior for a variety of circuits. This thesis presents examples that demonstrate the use for diagnostic testing as well as pass/fail production testing.

It may be concluded that the combination of the SVD of a linear circuit model and the mentioned iterative algorithm results in a method that optimally deals with a circuit's limited accessibility for measurements. Furthermore the method selects the smallest number of measurements, while maximally reducing the influence of random measurement errors. Thus it allows the behavior of a circuit to be verified with just a few measurements. The method is applicable even in the case of large measurement errors, because the selected measurements are of high quality, and their number is optimized.

Samenvatting

Dit proefschrift presenteert een methode voor het testen van analoge circuits die bruikbaar is voor zowel diagnostisch testen als de verificatie van hoog-niveau circuitgedrag. De methode is gebaseerd op een benadering van de eerste orde (i.e., een lineair model) van het niet-lineaire circuitgedrag. Zodoende is de methode bruikbaar als de deviaties van de circuitparameters (veroorzaakt door deviaties in het productieproces) relatief klein zijn, hetgeen vaak het geval is bij analoge circuits.

Het gelineariseerd circuitgedrag wordt beschreven door een modelmatrix, die wordt verkregen door, bijvoorbeeld, het berekenen van eerste orde gevoeligheden. Een complete orthogonale decompositie van de modelmatrix wordt gevonden door het berekenen van zijn Singuliere Waarde Ontbinding (SWO). Deze decompositie vormt de basis van de in dit proefschrift gepresenteerde methoden.

Een beperkte toegankelijkheid voor metingen van een circuit geeft aanleiding tot redundante factoren in het genoemde lineaire model. Deze worden geëlimineerd door alleen de meest belangrijke orthogonale componenten van de SWO van de modelmatrix te behouden. Een directe methode wordt gepresenteerd, die één meting kiest voor elk van deze orthogonale componenten. Om het aantal benodigde metingen te bepalen wordt zowel de toegankelijkheid voor metingen als de grootte van random meetfouten in acht genomen. De directe methode kan ook gebruikt worden om testbare circuitcomponenten te selecteren, om de testbaarheid van een circuit te analyseren.

Verder wordt een iteratief algoritme gepresenteerd dat metingen probeert te selecteren die de invloed van random meetfouten minimaliseren. Het algoritme is gebaseerd op het *D*-optimaliteitscriterium. Het iteratieve algoritme bereikt betere resultaten dan de directe aanpak. Dit is voornamelijk te danken aan het feit dat het een *complete* set metingen optimaliseert. Daarentegen selecteert de genoemde directe aanpak de metingen één voor één, waarbij telkens slechts de laatst geselecteerde meting geoptimaliseerd wordt. Het iteratief algoritme past ook het *aantal* metingen aan, waarbij rekening wordt gehouden met de verwachte grootte van de random meetfouten. Aan de hand van een voorbeeld wordt gedemonstreerd dat de iterative methode betere resultaten behaalt dan een reeds bestaande methode die gebaseerd is op de QR decompositie.

De besproken methodieken werden geïmplementeerd in een computerprogramma, hetwelk een testbaarheidsanalyse en/of selectie van metingen kan uitvoeren voor een verscheidenheid aan circuits. Dit proefschrift presenteert verschillende voorbeelden die het gebruik van de methode voor zowel diagnostisch testen als pass/fail productietesten demonstreren.

Geconcludeerd mag worden dat de combinatie van de SWO van een lineair circuitmodel met het genoemde iteratief algoritme resulteert in een methode die optimaal omgaat met de beperkte toegankelijkheid voor metingen van een circuit. Bovendien selecteert de methode het kleinste aantal metingen en reduceert ze maximaal de invloed van random meetfouten. Zodoende staat de methode een verificatie van het circuitgedrag toe met slechts enkele metingen. De methode is zelfs toepasbaar in het geval van grote meetfouten, omdat de geselecteerde metingen van hoge kwaliteit zijn en bovendien hun aantal geoptimaliseerd wordt.

Notational Conventions

a	real or complex number
$[a]$	largest integer not larger than a
\bar{a}	complex conjugate of a
\mathbf{a}	vector
\mathbf{A}	matrix
a_{ij}	element (i, j) of \mathbf{A}
$\bar{\mathbf{A}}$	complex conjugate of \mathbf{A}
X	statistical variable; x denotes an observed value of X
μ_x	mean value of X
σ_x^2, σ_{xx}	variance of X
σ_{xy}	covariance of X and Y
\mathbf{X}	vector of statistical variables; \mathbf{x} denotes an observed value of \mathbf{X}
$\hat{\mathbf{a}}$	least squares approximation of $\mathbf{Aa} = \mathbf{b}$
\mathbf{A}_r	row reduced matrix (some rows of \mathbf{A} are omitted)
\mathbf{A}_c	column reduced matrix
\mathbf{A}_{rc}	row and column reduced matrix
$(\cdot)^T$	transposed matrix or vector
$(\cdot)^*$	Hermitian (conjugate transpose) matrix or vector
(\mathbf{a}, \mathbf{b})	$\mathbf{a}'\mathbf{b}$
\mathbf{A}^{-1}	Inverse of \mathbf{A}
\mathbf{A}^+	Pseudo inverse (Moore-Penrose inverse) of \mathbf{A}
\mathbf{A}^L	Left inverse of \mathbf{A}

\mathbf{A}^R	Right inverse of \mathbf{A}
\mathbf{A}_k	Rank k approximation of \mathbf{A}
$\ \cdot\ _2$	ℓ_2 norm of matrix or vector
$ \mathbf{A} $	determinant of \mathbf{A}
$\text{rank}(\mathbf{A})$	rank of \mathbf{A}
$c(\mathbf{A})$	condition number of \mathbf{A}
$\mathbf{R}(\mathbf{A})$	range of \mathbf{A}
$\mathbf{N}(\mathbf{A})$	null space of \mathbf{A}
$\text{diag}(a_1, \dots, a_n)$	$\begin{bmatrix} a_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & a_n \end{bmatrix}$
\mathbf{I}	$\text{diag}(1, \dots, 1)$
$a \gg b$	a is much larger than b
$a \approx b$	a is about equal to b
$a \triangleq \dots$	a is defined as ...

Symbols and Abbreviations

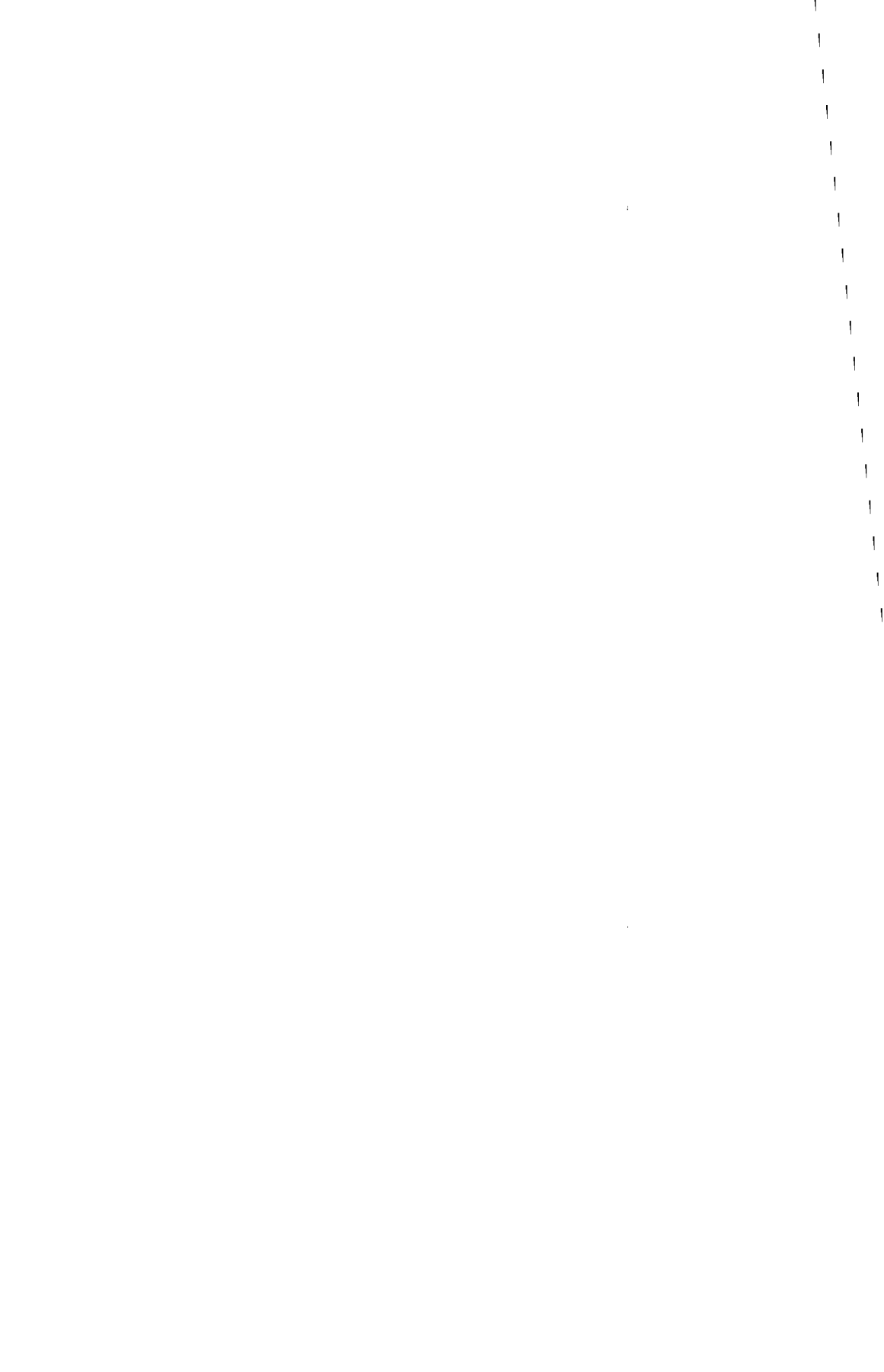
LS	Least-Squares
PL	Piecewise Linear
SVD	Singular Value Decomposition
p_i	i^{th} circuit parameter
\mathbf{p}	vector of circuit parameters
\mathbf{y}	vector of circuit outputs
F	network function
$s_{p_i}^F$	differential sensitivity of F to p_i
$\hat{s}_{p_i}^F$	normalized differential sensitivity of F to p_i
$\mathcal{S}_{p_i}^F$	incremental sensitivity of F to p_i
$\hat{\mathcal{S}}_{p_i}^F$	normalized incremental sensitivity of F to p_i
\mathbf{S}	sensitivity matrix
\mathbf{U}	part of the SVD ($\mathbf{U}\mathbf{W}\mathbf{V}^T$) of \mathbf{S}
\mathbf{V}	part of the SVD ($\mathbf{U}\mathbf{W}\mathbf{V}^T$) of \mathbf{S}
\mathbf{W}	part of the SVD ($\mathbf{U}\mathbf{W}\mathbf{V}^T$) of \mathbf{S}
\mathbf{C}_X	covariance matrix of variables \mathbf{X}

Contents

Summary	VII
Samenvatting	IX
Notational Conventions	XI
Symbols and Abbreviations	XIII
1 Introduction.....	1.1
2 Introduction to Testing Analog Circuits.....	2.1
2.1 Problems with Testing Analog Circuits.....	2.1
2.2 Review of Test Methods for Analog Circuits.....	2.6
2.3 Discussion.....	2.14
3 Exposition of Used Techniques.....	3.1
3.1 Circuit Sensitivities	3.1
3.1.1 Introduction and Sensitivity Definitions	3.1
3.1.2 Application of Sensitivities	3.7
3.1.3 Frequency Domain Sensitivities	3.8
3.1.3.1 Sensitivity to Magnitude and Phase.....	3.8
3.1.3.2 Differential/Incremental Sensitivity Relationship	3.10
3.1.3.3 Efficient Computation of Sensitivities in the Frequency Domain	3.11
3.1.4 Time Domain Sensitivities	3.14
3.2 The Singular Value Decomposition	3.17
3.2.1 The Spectral Theorem	3.17
3.2.2 The Singular Value Decomposition	3.21
3.2.3 Applications of the SVD.....	3.23

4 Using the SVD to Test Analog Circuits	4.1
4.1 Introduction	4.1
4.1.1 The Linear Circuit Model	4.1
4.1.2 Introduction to Test-Point Selection	4.7
4.2 The Influence of Measurement Errors	4.10
4.3 Direct Test-Point Selection	4.15
4.3.1 Dealing with Untestable Parameters.....	4.16
4.3.2 A Direct Method for Test-Point Selection	4.21
4.3.3 Determination of the Number of Test Points.....	4.25
4.3.4 Selecting a Subset of Testable Parameters	4.27
4.3.5 Dealing with Large Parameter Deviations.....	4.28
4.4 Discussion.....	4.29
5 Iterative Test-Point Selection with the SVD.....	5.1
5.1 The <i>D</i> -Optimality Criterion	5.1
5.2 An Iterative Approach.....	5.5
5.3 Selecting the Optimal Number of Test Points and Parameters.....	5.9
5.4 Discussion.....	5.12
6 Application of PL Techniques to Analog Testing	6.1
6.1 Introduction.....	6.1
6.2 Piecewise-Linear Modeling	6.2
6.3 Solving sets of Linear Inequalities	6.6
6.3.1 A Geometrical Solution Method.....	6.7
6.3.2 The Matrix Method of Tschernikowa	6.11
6.4 Analog Testing with PL Modeling and Interval Analysis.....	6.15
6.4.1 Solving Sets of Non Linear Equalities with Bounded Parameters.....	6.15
6.4.2 Application to Analog Testing.....	6.18
6.5 Discussion.....	6.24
7 Examples.....	7.1
7.1 Practical Implementation	7.1
7.2 2 nd order Bandpass Filter.....	7.4
7.3 10 bit Digital-to-Analog Converter.....	7.10
7.4 Application of the SVD to Black Box Testing.....	7.15
7.5 Discussion.....	7.21

8 Conclusions	8.1
8.1 Conclusions and Recommendations.....	8.1
8.2 Original Contributions.....	8.3
9 References	9.1
Appendix A <i>Efficient Calculation of the Singular Value Decomposition</i>	A.1
A.1 Introduction	A.1
A.2 Bidiagonalization.....	A.3
A.3 Diagonalization of the Bidiagonal Form.....	A.5
A.4 Discussion.....	A.6
Appendix B <i>Some Results from Statistics</i>	B.1
Appendix C <i>Geometrical Interpretation of the Least-Squares Approximation</i>	C.1
Acknowledgement	
Curriculum Vitae	



1

Introduction

In recent years, the cost of testing has become an increasingly large part of the total cost associated with the design and production of integrated circuits (ICs). This is mainly caused by continuing miniaturization of devices, which leads to complex ICs with a high functionality and many specifications that must be verified. This is true for digital as well as analog devices; in both cases it is important to reduce as much as possible the contribution of the testing overhead to the total production cost. Hence, efficient test techniques are called for in the various stages of the development cycle of an integrated circuit.

According to Figure 1.1 (based on [Huer93]) , the IC development cycle can be divided into four distinct stages. First of all, before a circuit goes into production its design is verified. In this stage, circuit simulations are used to assess if a design is functionally correct. Also it should be verified that the design is not excessively sensitive to slight fluctuations of some characteristic of its elements. Furthermore, layout extraction assesses the influence of layout-specific effects like parasitic capacitors or wiring resistances.

After design verification is completed, a prototype is manufactured and characterized, often using time-consuming (hence expensive) diagnostic techniques. Evaluation costs may be very high in this stage. After errors (e.g., layout errors) are fixed and possible

design changes are incorporated an initial production run is made, which consists of a relatively small batch of wafers. A spread of process parameters is often enforced in this initial run, to evaluate the influence of these parameters on IC performance and to obtain a yield estimate.

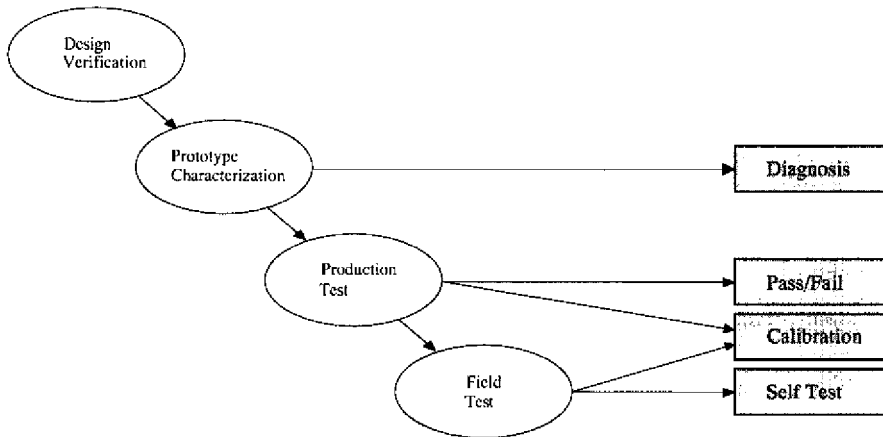


Figure 1.1. Testing techniques in various stages of a circuit's development cycle.

Yield may be improved by tolerance design and design centering. The process is illustrated in Figure 1.2 (based on [Swar87]), which depicts the two-dimensional parameter space of a circuit with parameters p_1 and p_2 . For example, p_1 and p_2 are geometrical parameters like transistor length and width. The worst-case tolerances of the parameters define the boundaries of a rectangular tolerance region R_t . The center \mathbf{p}_0 of this region corresponds to the nominal parameter values. In addition, the parameter values that result in devices with acceptable performance define in parameter space a region R_f , called the feasible region. Because the relation between the circuit performance and the parameter values is complex, R_f is often irregularly shaped. The center of this region is \mathbf{d}_0 .

In Figure 1.2, acceptable produced circuits correspond with region $R_t \cap R_f$. Yield is defined as the ratio of the number of acceptable circuits to the total number of produced circuits. It follows that 100 % yield can be obtained only if $R_t \subseteq R_f$, as in

Figure 1.2 b. In contrast, Figure 1.2 a depicts a situation where a circuit with nominal parameter values is acceptable, but yield is less than 100 % due to the parameter tolerances.

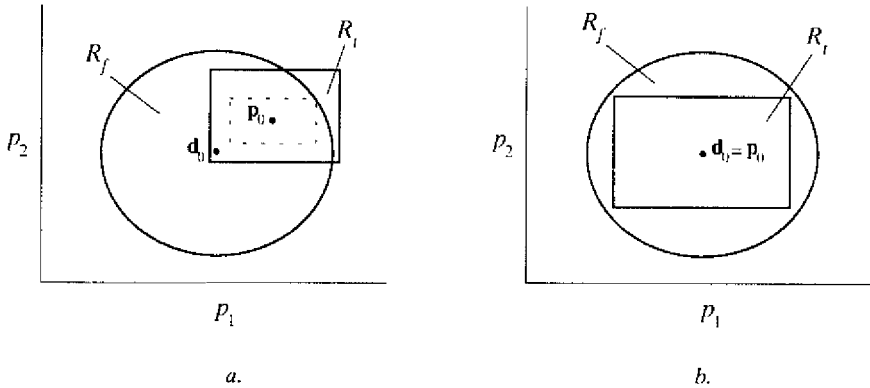


Figure 1.2. Feasible region and tolerance region.

To improve yield, the tolerances on selected parameters can be reduced (tolerance design). This increases the relative size of region $R_t \cap R_f$. In this manner tolerance design increases the relative amount of acceptable circuits. A possible result of this procedure is indicated with a dotted rectangle in Figure 1.2 a. In principle, 100 % yield can be obtained by just assigning smaller tolerances to some (possibly all) parameters, without changing the nominal values \mathbf{p}_0 . However, tightening parameter tolerances will generally increase production cost. In addition, in an integrated circuit the parameter tolerances often cannot be directly controlled. Therefore *design centering* is used. The idea behind this approach is to center R_t on \mathbf{d}_0 by adjusting the nominal parameter values \mathbf{p}_0 . In many cases design centering can be combined with a relaxation of some parameter tolerances. As indicated in Figure 1.2, design centering allowed the tolerance on both parameters to be increased. Thus design centering improves yield while the increased parameter tolerances minimize production cost. One problem in design centering is to find the feasible region R_f . For a discussion of several approaches, see [Swar87].

Once a design is verified and approved for mass production, the goal of testing changes: now it is often sufficient to *detect* deviations that are too large: The fault that causes an unacceptable circuit need not be traced because in most cases repair or calibration of a mass-produced IC is unfeasible. Therefore production testing is often a pass/fail (or go/no-go) testing. Note that time spent in production testing should be kept to a minimum to decrease production cost. Therefore the number of applied test stimuli must be minimized, so they should be carefully selected to maximize the likelihood of detecting a possible fault. In addition, the selected measurements should be easy to perform.

To meet these requirements, the first production tests are often performed through a bed of needles (test probes) on the naked die, before a device is cut from the wafer. It is often not possible to test the device at its normal (i.e., high) operating speed due to the probes' loading effects. Therefore mostly static DC measurements are used at this stage. Such tests cannot verify dynamic circuit behavior. However, they are justified because the incremental cost of detecting a fault rises approximately with a factor of 10 with each stage of the production process [Huer93]. Thus every effort is made to detect faults in an early stage of the production process. The final test checks the functional behavior of the packaged device under normal or extreme operating conditions, using only external pins.

Circuit calibration optimizes the performance of a produced IC by adjusting some circuit parameters. For example, the value of an on-chip resistor may be modified by laser trimming until IC performance is adequate. Because of its high cost, such *off-line* calibration is mostly limited to high-performance ICs. An alternative is to use *on-line* calibration, where the circuit design is enhanced to allow a form of compensation (e.g., an off-set compensated opamp) or auto-calibration during normal device operation.

Ultimately, an IC is integrated into a larger system, for example by mounting it on a printed circuit board. It is difficult to perform an external test (i.e., using external testing equipment) of the mounted IC. To ease the problem, specialized Built-In Self-

Test (BIST) circuitry can be added to the IC in the design stage. This circuitry monitors the internal operation of the IC, during its normal operation or in a special test mode. BIST techniques may also detect faults caused by aging or environmental conditions.

The general classification of development stages and testing techniques that is illustrated in Figure 1.1 applies to both analog and digital circuits. However, it is noteworthy that techniques like automatic design, design verification and test are much more mature for digital circuits than for analog circuits. Chapter 2 investigates in more detail the causes for the relative lack of progress in the analog field. It also addresses the main difficulties in analog circuits testing.

Due to the properties of analog circuits and signals there exists a wide diversity of approaches to testing analog circuits. To illustrate, Chapter 2 presents a brief overview of test methodologies. This overview shows that the focus on analog testing has been shifting from diagnostic testing to production testing and the implementation of BIST structures. This is caused by the continuing miniaturization of ICs, which often cannot be repaired and which are relatively inaccessible for measurements. As a result there is a need for test methods that *detect* faulty circuits by evaluating the *effect* of a fault. Determination of the fault *cause* is considered less important in production testing.

Many approaches to production testing maximize *fault coverage*. Such approaches select those measurements that detect the highest possible percentage of faults. It is often necessary to add extra test nodes or BIST circuitry to take into account the faults that are not detectable on the functional level. Such faults appear in additional circuitry like BIST or calibration circuitry, for example. Although these faults do not instantaneously influence IC functionality, they may decrease circuit *reliability*. Thus, maximizing fault coverage implicitly tests circuit reliability.

If reliability issues are not a concern, production testing could as well be based on a *functional* test, which may be considerably cheaper than exhaustively testing for all

possible faults. Functional testing assumes that in a properly designed circuit, parameter deviations are either discernible at the functional level, or taken into account by the circuit design. Thus only deviations of functional circuit behavior need be taken into account.

A very important issue in analog testing is the selection of measurements. The reason is that an IC offers relatively few nodes where voltages or currents can be measured. In addition, measuring *internal* currents often is impossible because connections cannot be broken. Because of the resulting limited accessibility for measurements, the selected measurements must be *revealing*. This means that the measurements should have a high probability of detecting a deviation from nominal circuit behavior. Although many current approaches to production testing focus on obtaining a high fault coverage, often little attention is paid to the effectiveness of the selected measurements.

Another important issue in analog testing is the influence of measurement noise. It is in general not possible to distinguish the signal information content from the measurement noise, due to the continuity of analog signals. Thus, the measurements must not only be revealing, but also they should be selected so that the influence of measurement noise is minimized. In addition quantitative estimates should be provided that assess the influence of measurement noise.

This thesis presents a method that is mainly intended for production testing. It accurately predicts the effect of small parameter deviations on circuit functionality. To accomplish this, it first predicts circuit behavior (e.g., DC response, frequency response) from a few carefully selected measurements. Next, performance parameters are derived from the predicted behavior to determine whether or not a circuit is at fault. For example, the maximum linearity error of a D/A converter is determined from the linearity error that is predicted at all code words.

In its current form the presented method does not provide accurate predictions in the presence of large parameter deviations. However, in analog circuits already small

deviations from nominal circuit behavior can be unacceptable. Thus the method has many practical applications. In addition, it is shown how some principles used in the method may be applied to cope with larger parameter deviations.

The method assesses the influence of small parameter deviations on circuit functionality by linearizing circuit behavior at a bias point. The first-order gradients that are obtained at the bias point describe the cumulative influence of small parameter deviations on circuit behavior. These gradients, called *differential sensitivities*, are collected in a matrix to obtain a linear circuit model. This linear model is then analyzed by a combination of numerical and statistical techniques.

First, the singular value decomposition (SVD) of the model is computed to decompose the linear model into orthogonal components. Rank-deficiency of the linear model is dealt with in an optimal manner by retaining only the largest orthogonal components (such rank deficiency is caused by a circuit's limited accessibility for measurements.) The advantage of the SVD is that it exists for a large class of matrices. In addition, it copes with rank-deficiency in an optimal manner. Also, it provides a criterion for the selection of measurements: a single measurement is selected for each of the orthogonal components. It is shown that such selection of measurements also takes into account the influence of random measurement errors.

In addition to this direct approach, an iterative approach is presented. Here measurements are selected on the basis of the *D*-optimality criterion, also used in the design of experiments. It will be demonstrated that this criterion minimizes the influence of random measurement errors. The result is a reliable complete test method that selects measurements of high quality.

The organization of the rest of this thesis is as follows.

Chapter 2 summarizes the specific properties of analog circuits that hamper the development of test methods for analog. In addition, it provides an overview of existing analog test methods.

The first part of Chapter 3 treats differential sensitivities. Their usefulness is evaluated, and a comparison is made with other types of sensitivities. Also efficient computation of sensitivities is discussed. The second part of Chapter 3 introduces the SVD, and several of its applications.

Chapter 4 presents the already discussed method that selects measurements on the basis of the SVD of the linear model. The iterative method is discussed in Chapter 5.

Chapter 6 discusses the application of piecewise-linear techniques and interval analysis to analog testing. It will be shown that the combination of these techniques offer some specific advantages. The concepts presented in Chapter 6 do not constitute a complete test method. For example, the selection of measurements has not yet been fully researched. However the discussed approach may serve as the basis of a more complete analog test method.

Chapter 7 present examples that demonstrate the applicability of the methodologies presented in Chapters 4 and 5. Direct comparisons between the direct and iterative methods for test-point selection will be made, on the basis of a realistic example. Also it is shown that the iterative method selects better test points than some existing methods. A final example demonstrates the application of the SVD to a different kind of functional testing, here called black-box testing.

Finally, Chapter 8 briefly summarizes the main results and provides some recommendations for future research.

2

Introduction to Testing Analog Circuits

This chapter summarizes the main difficulties encountered in testing analog circuits. In addition it reviews some existing approaches to analog testing. This creates a context for the new approaches that will be discussed in the following chapters. Section 2.1 explains the difficulties caused by specific properties of analog devices and signals. These give rise to a diversification of test methods, as becomes clear from the overview of Section 2.2. Section 2.3 presents a brief concluding discussion.

2.1 Problems with Testing Analog Circuits

As discussed in Chapter 1, analog test method have advanced at a less rapid rate than digital test methods. The main cause is the relative immunity (in comparison with analog signals) of digital signals to deviations in digital circuitry or disturbances of physical signals. Predefined thresholds separate the different levels of a digital signal. Therefore the information contained in the signal is relatively immune to disturbances such as thermal noise or cross talk. Furthermore a digital circuit is built from a limited library of devices with well-defined properties, like fan-in, fan-out and gate delay.

Therefore a digital test method can accurately represent signals and devices by relatively simple (i.e., abstract) models.

Naturally, testing digital circuits also presents problems. Increasing integration, higher clock speeds and lower supply voltages cause problems like reduced fan-out of gates, increased loading effects of (built-in) test circuitry (interfering with device operation), and decreased noise margins.

To illustrate, consider the timing behavior of logic circuits. Due to aggressive optimization of clock rates, many paths in optimized large combinatorial circuits have a delay that is close to the maximum circuit delay. Therefore delay fault testing becomes increasingly important [Crem96]. In addition, increasing use of non-conventional digital circuits requires error models more elaborate than stuck-at models (for example, see [Wohl96]). This is necessary because otherwise there will be too many unmodelled (hence undetectable) faults. Finally, due to sheer circuit complexity¹ it becomes very difficult to test all possible states of a sequential circuit. The problem is that some faults can only be detected by reaching these states [Raja96].

Due to such difficulties, the *analog* nature of *digital* signals and devices grows more and more pronounced. Consequently some recent approaches to *digital* testing apply non-traditional² techniques. For example, [Chat96] detects stuck-at faults in combinatorial circuits by using signal waveform analysis. [Chan96], uses a very low supply voltage (2 to 2.5 times the transistor threshold voltage) to detect “weak” CMOS chips. Such chips contain flaws that cause intermittent or early-life failures, which are otherwise not detectable.

Notwithstanding the discussed problems with digital, the situation with analog has been even more complex. The causes for this situation may be summarized as follows.

¹ In the next few years a single digital circuit may contain several million gates.

² In contrast to the established “traditional” techniques for digital testing.

First, analog signals are continuous in nature. For example, in many analog circuits the signal information is represented by the precise amplitude of voltages and currents at any time instance. Even a small disturbance of signal magnitudes may cause a serious degradation in signal quality, which amounts to loss of information. This becomes especially clear when one considers the large dynamic range (up to 120 dB) of some analog circuits. A source of signal disturbances is noise, e.g. thermal noise, or $1/f$ noise. Another source is mismatch and unbalance between parts of the circuit. In a mixed-signal device, crosstalk and clock feed-through from the digital circuitry may interfere with analog signals. In the digital domain signals are not so easily affected by such disturbances, as long as it is ensured that their common influence is below a well-defined threshold. Of course, timing problems (which may be considered a form of signal-degradation) in current-day digital chips pose serious challenges. However, it is safe to conclude that in digital circuits a relatively high immunity to disturbances is acquired at the expense of a lower signal information content.

Second, the behavior of many analog devices is inherently non linear. Many circuits composed of such devices (e.g., analog multipliers) are designed to exhibit a non linear input-output characteristic. As a result, sophisticated techniques are necessary to solve the (non linear) equations that describe circuit behavior. In contrast, within well-defined limits, non linearities in a digital device do not influence its behavior because only its “digitized” behavior is of interest.

Third, analog signals may be described in several domains, e.g., DC, frequency, time continuous and time discrete. In principle, each domain needs its own methodology to describe device and circuit behavior. However, sometimes it is possible to use interrelations between different domains. For example, an analog filter may be characterized in terms of its poles, zeros and DC gain. This defines the filter’s behavior in three domains: the DC domain (DC gain), frequency domain (Bode plots of amplitude and phase) and time domain (impulse response). This implies that its impulse

response can be used to implicitly test its frequency behavior, saving valuable test time. However, such an approach is not always possible.

The three discussed items result in the following problems in analog testing:

1. A test method for analog must take into account the influence of measurement errors as an inherent part of the method. For analog signals it is in general not possible to define discrete levels that allow the signal information content to be distinguished from measurement noise. Therefore, methods for analog testing should carefully select measurements that minimize the influence of random or systematic measurement errors.

For example, consider the case where the value of some device parameter (e.g., input impedance) must be checked. In order to detect a high percentage of faulty devices³, the parameter of interest must be computed from the measurements to at least a certain precision. To estimate whether this precision is actually obtained with a certain set of measurements, the measurement errors must be translated into an uncertainty on the computed parameter. This translation may be problematic, since the relation between measurements and a computed parameter is in general non linear. To simplify the situation, it is often necessary to make certain assumptions about the measurement errors (for example, that they are uncorrelated). However the resulting statistical model of measurement errors does not necessarily accurately describe measurements made in an actual test environment.

2. Even a relatively simple analog circuit is characterized by a diversity of specifications. Consider a four-transistor Operational Transconductance Amplifier, for example. This simple circuit is characterized by performance parameters like gain, bandwidth, rise time, (complex) input impedance and output impedance. Furthermore, the relation between such parameters is generally non linear.

³ And also to ensure that good devices are not rejected.

Therefore it is common practice to measure all performance parameters separately. This introduces a redundancy in the measurements that is in principle unnecessary. Additionally, due to the complex behavior of analog circuit, it is much more difficult to define macro models for analog than for digital circuits.

3. Accurate analog fault models are not always available [Soma91]. The basic problem is to find a circuit description that models the behavior of a good as well as a deviating circuit, for all types of faults that occur in practice. In contrast, for digital circuits, simple models like stuck-at and short have been very successful, due to the mentioned abstract nature of digital signals.
4. Due to the continuing miniaturization of integrated circuits, some fundamental problems arise. First, properties of analog devices change with increasing down-scaling. For example, due to shorter channel lengths of a MOS device its voltage gain decreases [Host90]. Such trends result in new circuit techniques and more complicated descriptions of analog devices. Second, the influence of thermal noise, crosstalk and parasitic capacitors increases, decreasing the relative precision of measurements of analog signals. Third, the accessibility for measurements decreases because the complexity of the circuits rises faster than the number of pins through which it is accessible.
5. There are problems with the implementation of a BIST structure for analog circuits [Chin90]. In digital circuits, a BIST structure can verify circuit behavior by applying a number of input vectors generated by (for example) a relatively cheap on-chip pseudo-random generator. Output vectors may be read from internal nodes through added scan-elements. An output vector can be evaluated on-chip by some extra logic. Also test vectors may be stored in a shift register and transferred serially to the outside world, using only a single extra pin.

These principles cannot be directly applied to analog circuits, because there it is very difficult to accurately store or transfer signals without losing some of the

information content (e.g., magnitude, phase). The solution seems to be to design specialized (and difficult to design) testable analog subcircuits, instead of just adding some BIST circuitry to off-shelf subcircuits, as is more common in digital.

For example, designs for configurable opamps are proposed that are capable of operating in a normal configuration and a special test configuration [Vasq96], [Reno96]. In its test configuration the function of the circuit is modified to increase the controllability and observability of internal nodes. Special care has to be taken in the design of such analog BIST structures. Switches should not be introduced in the signal path, and loading of internal nodes by test circuitry should be kept to a minimum.

Not surprising, most of the discussed problems in analog test also hamper automatic *design* of analog circuits. Although some successful approaches are reported (a summary is given in [Rute93]), major breakthroughs are needed before the automatic design of an arbitrary analog circuit becomes feasible.

The mentioned problems result in a variety of analog test methods, each tailored to a specific application. The next section reviews some existing approaches.

2.2 Review of Test Methods for Analog Circuits

This section review some existing approaches to testing analog circuits. The outline of this section is based on Figure 1.1. In addition to fault location techniques (which can be used for diagnostic testing), production testing and tolerance analysis will be briefly treated. The discussion of these subjects provides a context for the techniques presented in Chapter 3.

Several comprehensive reviews of test methods for analog can be found in the literature. For example, [Band85] classifies fault location methods according to the tree of Figure 2.1. A first distinction is made according to the stage in the testing

process at which some form of simulation of the tested circuit occurs. Simulation before test has the advantage that most of the work is performed off-line, which reduces testing cost.

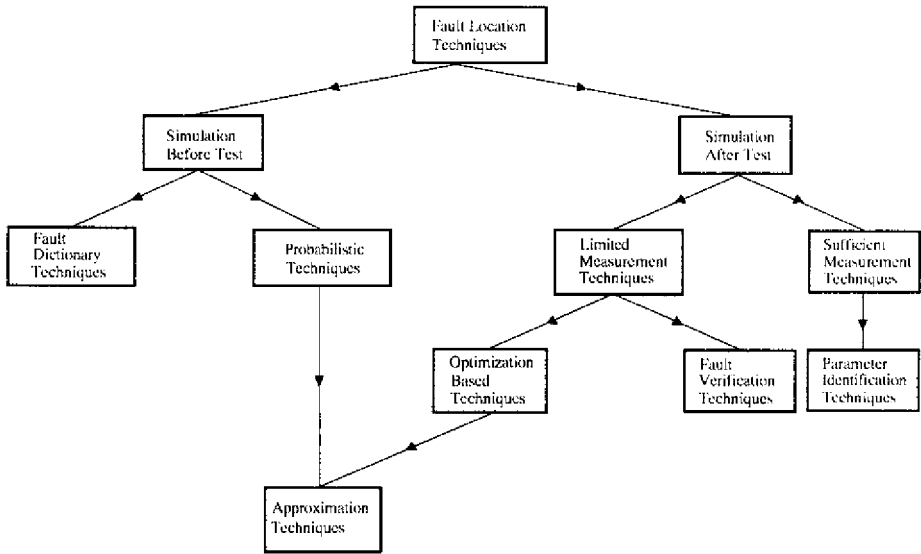


Figure 2.1. Classification of fault location techniques, according to [Band85].

The *fault dictionary technique* has been widely applied in the digital world. Early applications tested digital circuits implemented on printed circuit boards. A system was tested by comparing failed-board output levels with a set of pre-stored outputs on the automatic test equipment, under the application of selected input vectors.

An early application of the fault dictionary technique to analog testing is described in [Hoch79], which presents a method that uses DC input stimuli to test a non linear analog circuit. The approach may be outlined as follows. First, a test engineer decides which type of faults are taken into account. This includes opens, shorts, and low-gain devices, for example. The faults are modeled at the circuit level. Then they are introduced one by one, and their effect on a number of test nodes is evaluated with the use of a circuit simulator. In this process the circuit's input stimuli are selected to exercise the "on," "off," and "linear" state of its semiconductors. Each simulation

obtains a *fault signature*. This is a set of circuit responses, corresponding to certain input stimuli and a certain fault. A *fault dictionary* is built by storing the output signatures.

After this stage of *fault simulation* is completed, an algorithm is used to divide faults into disjunct *ambiguity groups*. The faults in an ambiguity group generate approximately the same output levels, causing these levels to be in a certain *fault band*. Thus the fault dictionary associates with each ambiguity group a fault band and specific input stimuli. Next the most useful input stimuli are obtained by manipulating the ambiguity groups. This process determines what faults can be isolated and what input stimuli provide the highest degree of isolation.

To test a circuit, the preselected input stimuli are applied, and the circuit's output signature is measured. Now an ambiguity group is selected by comparison of the output signature with the fault signatures stored in the fault dictionary. Like this, output deviations are traced to the smallest group of faults whose influence is indistinguishable.

The advantage of dictionary-based approaches is that they generally have a low on-line computational requirement: most of the work is associated with building the dictionary. However, for very large circuits an excessive amount of time may be spent in dictionary construction, as well as in post-test location of faulty elements. In addition, small deviations from the elements' nominal values are hard to detect because they are unlikely to cause the measured outputs to deviate out of a certain fault band. Therefore the dictionary approach is most suitable for catastrophic and large deviation faults.

Approximation techniques try to locate the most likely faulty elements with a limited number of measurements. Of these approaches, *probabilistic techniques* resemble the fault dictionary approach. For example, the *inverse probability technique* [Brow81], is a technique that is primarily applied to locate single faults. In the pre-test phase a

statistical diagnostic database is constructed from Monte-Carlo simulations. In the post-test stage, extensive on-line computations are necessary to calculate from a limited number of measurements for each network element the probability that it is at fault.

Optimization-based techniques use some form of optimization criteria, like the l_2 (least squares) criterion [Rans73]. That approach determines the deviations of faulty elements from nominal that would cause output deviations that most closely match the actually measured output deviations.

Parameter identification techniques try to solve all network element values, for which generally a large number of test nodes and measurements are needed. After the values are computed, the faulty elements are identified by checking what element values are outside the design tolerance margins. There exist techniques for linear as well as non linear networks, in the DC, frequency and time domain. Also theoretic results are derived that give conditions under which the values of elements of a specific network are computable. For instance, [Navi79] derives the necessary and sufficient conditions for element-solvability of linear resistive networks.

Finally, *fault verification* techniques determine the values of faulty elements or try to localize faults to a small sub network.

The mentioned methods for fault location may be denoted as *fault-driven* tests, since they evaluate the influence of specific faults on circuit behavior, with the help of analog fault models. Specifically, deviations of circuit behavior are traced to one or several faulty circuit elements. This property is shared by many early methods, for example [Saek72], [John79], [Navi79], [Bier81]. Such methods are useful when a circuit needs to be diagnosed in a design process, or repaired at a test facility.

To properly diagnose the cause of a failed circuit, the number of accessible (controllable and/or observable) test nodes must be large. For example, to illustrate the

effectiveness of the dictionary approach discussed above, [Hoch79] presents an example of a video amplifier that is implemented on a printed circuit board. In such an application the available number of test nodes is large, so the accessibility of the circuit under test is excellent. Furthermore repair of a faulty circuit is entirely feasible.

However, over the years the relative number of accessible circuit nodes has steadily decreased due to increasing circuit miniaturization. This seriously limits an integrated circuit's accessibility for measurements. Furthermore, repair or adjustment of a mass-produced integrated circuit is often not feasible, and therefore it is common practice to discard faulty devices. As a result, currently much attention is given to pass/fail production testing.

Apart from its use in diagnostic testing, *fault simulation* can also be successfully applied to production testing. However, in this application it is not important to locate a fault; instead, only the influence of faults on circuit behavior or reliability is of interest. One approach uses analog fault models to simulate various fault conditions in a circuit model [Bell95]. Then the effect of these faults on the circuit output is evaluated and the fault coverage is assessed. A disadvantage of such a circuit-level approach is that a large number of possible faults must be simulated, while it is not certain whether or not the simulated faults accurately represent the faults that occur in a device that is produced with a certain process. This last disadvantage can be remedied by using a *defect-oriented* approach.

A *defect-oriented* approach evaluates the influence of physical defects on an integrated circuit. Such an approach uses information about the manufacturing process to assess the types and magnitude of physical errors that may occur. For example, [Harv95] simulates a large number of defects, such as oxide pin holes and extra metal, using a kind of physical simulator. These defects are translated to faults at the circuit level, using a fault model consisting of a range of resistances. Some defects do not result in faults, for example defects that appear in an unused region of the layout. After the faults are available at the circuit level, a fault simulation approach is used to assess the

influence of the faults, and to estimate the fault coverage. The advantage of the discussed defect-oriented method is that specific information about the production process may be taken into account to derive realistic circuit-level fault models. A disadvantage is that a lot of time may be involved with the simulation of defects.

To obtain a higher fault coverage it is possible to include *Built-In Self-Test* (BIST) circuitry on the chip that performs checks on-chip [Olbr95]. Also it is possible to increase the number of test nodes to improve the circuit's accessibility for testing. For example, measurements at internal test nodes may be stored in a built-in analog shift register [Chin90] and accessed externally in a later stage through a few pins. This approach is borrowed from digital testing. However, the disadvantages that were discussed in Section 2.1 still apply.

The various forms of fault-driven testing, including various BIST structures, can result in a very high fault coverage. This may be of advantage, because also faults are detected that are not apparent from measurements at the circuit outputs. For example, such faults may decrease a circuit's reliability.

An interesting time domain approach to BIST uses the impulse response of a linear time invariant analog circuit as a signature to determine whether or not it is at fault [Pan96]. It is very difficult to measure the impulse response of a circuit directly. For example, the energy in a pulse is so small that the output response would be corrupted by the noise. Increasing the energy level (height) of the pulse will drive the circuit into non linear operation. Therefore the following approach is used. The circuit is enclosed by a D/A converter (DAC) and an A/D converter (ADC). A digital pseudo-random generator is then used to drive the circuit input through the DAC. The digital circuit input (delayed by m cycles) and digitized circuit output are used to calculate the cross-correlation between the circuit's input and output signals. This is a good estimate of the impulse response at time point m , if the number of samples is large enough. To obtain the impulse response at N time points, the above procedure is repeated N times, each time with a different delay m .

In the pre-testing phase a large number of good and faulty circuits is generated by randomly varying *device parameters* (like gate width and length, and threshold voltage) that are assumed to be normally distributed. This Monte-Carlo simulation obtains a large number (1000) circuits, which may be good or faulty. By using SPICE simulations of the circuits their *performance parameters* (like DC gain) are obtained. Comparison of the performance parameters with their allowed range allows the circuits to be classified as good or faulty. Then the signatures (impulse responses at a certain time point) of the circuits are calculated. Thus, a lot of points in the signature space are obtained, and for each point it is known if it corresponds to a good or a faulty circuit. Now the optimal signature set (time points at which the impulse response is considered) is found. This set allows the best separation in the signature space between the good and the faulty circuits.

The described method uses a form of tolerance analysis, where deviations of the parameter tolerances are translated (in this case, using Monte-Carlo analysis) to a certain distribution of a circuit performance. Such an approach can also be used to compute the yield of a circuit, in the initial stages of manufacturing. If the yield is to low, some parameter tolerances can be narrowed. However this will increase the manufacturing cost of the circuit, as discussed in Chapter 1.

If circuit reliability is not an issue, complete fault coverage is not all-important. Thus a fault-driven approach is not necessary; instead a kind of *specification-driven* test may be used. Here, faults are considered to be deviations from a circuit's nominal functional behavior [Huer93]. For example, suppose that the DC behavior of a D/A converter is specified at a functional level by its response to all code words. Then, to decide if a converter is faulty (with respect to its DC behavior) it suffices to compute somehow its maximum linearity error. If this error is within allowed limits then the converter is considered functional. Such an approach may also be called a *functional test*: It checks a circuit's functionality without establishing the values of its components. This may lead to a considerable reduction of testing time (hence cost), because faults that do not

directly influence the circuit behavior are not taken into account. A disadvantage of a specification-driven test is the difficulty of obtaining estimates of the percentage of faulty circuits that is recognized by a certain test.

Instead of using a Monte-Carlo analysis to translate parameter tolerances to variations in circuit performance, it is also possible to use a truncated Taylor expansion of the nominal circuit description [Bray80]. In particular, many methods use first-order approximations called *differential sensitivities* [Sten87], [Dai90], [Hemi90]. As will be discussed in Section 3.1.1, a differential sensitivity is a first-order approximation of the influence of a parameter deviation on a circuit output. The differential sensitivities may be collected in a *sensitivity matrix* that models the *cumulative* influence of small parameter deviations on the circuit outputs.

A sensitivity-based approach can be used for fault diagnosis as well as specification-driven testing. For example, [Sten87] uses differential sensitivities to trace deviations of circuit behavior to certain components. A measure of component testability is defined that takes into account random measurement errors as well as the accessibility of the circuit for measurements.

Sensitivity-based methods use various approaches to select input stimuli that sensitize the circuit outputs to component deviations. For example, [Hemi90] uses an ad hoc approach that evaluates the independence of pairs of rows or columns of the sensitivity matrix to select testable components and input stimuli. Alternatively, [Sten87] uses a QR decomposition of the sensitivity matrix. The advantage of the latter approach over [Hemi90] is that it has well-known numerical properties, due to the application of a stable matrix decomposition technique.

The disadvantage of first-order sensitivities is that they are only useful if the deviations of the circuit components are small. If this is not the case then it is possible to use another type of sensitivities: *incremental sensitivities* [Slam92], [Slam94]. A precise definition is given in Section 3.1.1. Incremental sensitivities accurately model the

influence of arbitrarily large component deviations, provided that all components are linear. The obvious advantage is a more precise test method, in the case of linear circuits. A complication with the use of incremental sensitivities is that they result in a non linear circuit model. Therefore it is less straightforward to select test points than when differential sensitivities are used. Furthermore it is not easy to evaluate the influence of random measurement errors on the predictions of component deviations and high-level circuit behavior. A final limitation is that for non linear circuits incremental sensitivities do not provide exact predictions. Thus, for these circuits incremental sensitivities do not offer a larger precision than differential sensitivities, unless the non linearity is very modest.

This section is concluded by some remarks on the relative merits of testing in the DC, frequency or time domain. As remarked in the previous section, the DC domain has the advantage that the measurements are very simple to make. Because parasitic capacitances associated with measurement probes are not a problem, DC measurements are often used as a pre-screening test. In such a test measurements are made with probes placed directly on the wafer. A disadvantage of DC measurements is that they are not as revealing as, for example, frequency measurements. Therefore, DC measurements seem to be mainly useful when circuit accessibility is high. Another advantage of frequency measurements is that in most cases they are fairly easy to make. However it is rather time-consuming to test a device at a large number of frequencies. In that case time-domain measurements are an alternative. For example, the circuit impulse response may be used, as discussed above. The advantage of a time-domain approach is that the testing time is very short. Furthermore, deviations in the DC and frequency domain will also show in the time domain. Thus, time domain testing allows a good distinction to be made between good and faulty circuits.

2.3 Discussion

This chapter reviewed a few common approaches to analog testing. The bottom line is that, at this moment, there does not seem to be a method that has decisive advantages

over other methods. This is caused by the fact that analog test-methods are strongly application-dependent, due to specific properties of analog devices and signals. However, it is possible to extract from the preceding discussion a few key-items that should be addressed by an analog test method.

- In the production of integrated circuits, the relative cost of analog testing increases. Therefore the number of selected measurements should be minimized, to decrease testing time. This means that redundant measurements must be eliminated.
- The influence of noise cannot be neglected in analog testing. Thus it is prudent that this influence is taken into account as a fundamental part of the method. The measurements should be selected so that the influence of measurement noise is reduced to a minimum.
- It is important to keep into mind goal of testing, to prevent superfluous test. For instance, if circuit reliability is not important, then it is not necessary to check for the existence of faults that do not influence the circuit's functional behavior. In this case some form of functional test may be used to reduce testing cost.
- A test method should use all available information. For example, information about the manufacturing process can be used to rank possible faults, so that the first measurements try to detect the most common faults. This enhances the probability that a fault is detected with only a few measurements.

The method that is presented in this thesis addresses the first three items. Because information about the manufacturing process was not available, the method was not designed to take such information into account.

3

Exposition of Used Techniques

This chapter presents the techniques that will be applied to analog testing in Chapters 4 and 5. Section 3.1 introduces various types of circuit sensitivities and summarizes some applications. Also the efficient computation of frequency-domain and time domain sensitivities is discussed. Section 3.2 derives the Singular Value Decomposition (SVD). This is a well-known complete matrix decomposition whose properties will prove to be useful for the testing of analog circuits. Applications of the SVD to data analysis, the approximation of matrices and the least-squares approximation are discussed. These subjects are closely connected to analog testing, as will be explained in Chapter 4.

3.1 Circuit Sensitivities

3.1.1 Introduction and Sensitivity Definitions

Circuit sensitivities are widely applied in circuit simulation, optimization and testing. A comprehensive description may be found in several text books, e.g., [Bray80], [Swar87], [Ogro94] and [Vlac94]. In general terms, a circuit sensitivity is a quantitative measure of the influence of a parameter change on (some aspect of) circuit behavior. Formal sensitivity definitions are given later. Several factors contribute to parameter changes, e.g., deviations of the manufacturing process, temperature variations, and component aging. The influence of parameter changes on circuit

properties like voltage transfer function or delay time may be assessed by evaluating the corresponding circuit sensitivities. This process is commonly referred to as a *sensitivity analysis*. Section 3.1.2 presents some applications of a sensitivity analysis.

In general, the time-dependent behavior of a circuit containing linear or non linear elements may be described as

$$\begin{aligned} \mathbf{q}' - \mathbf{E}\mathbf{x} &= \mathbf{0}; & \mathbf{q}_0 &= \mathbf{q}(0) \\ \mathbf{f}(\mathbf{q}, \mathbf{x}, \mathbf{w}, \mathbf{p}, t) &= \mathbf{0} \end{aligned} \quad (3.1)$$

This notation assumes that the terminal equations for capacitors and inductors are defined in terms of charges and fluxes, collected in \mathbf{q} . The elements of matrix \mathbf{E} are either 1 or 0, and \mathbf{x} represents the circuit variables (nodal voltages or branch currents). All non linearities are incorporated in the algebraic system $\mathbf{f}(\mathbf{q}, \mathbf{x}, \mathbf{w}, \mathbf{p}, t) = \mathbf{0}$, so the differential equations $\mathbf{q}' - \mathbf{E}\mathbf{x} = \mathbf{0}$ are linear. The initial conditions are represented by \mathbf{q}_0 . Furthermore, \mathbf{w} is a vector of excitations, and \mathbf{p} contains the circuit parameters like parameters of linear or non linear components (e.g., capacitance, transistor threshold voltage), or parameters of circuit excitations (e.g., voltage of a voltage source). An element of \mathbf{p} may also be a (non linear) function of the circuit parameters. It is assumed that for each \mathbf{p} there is only one solution of \mathbf{x} .

Definition 3.1 A *network function* $F(\mathbf{x}, \mathbf{p}, \gamma)$ is defined as a non linear scalar function of the circuit variables \mathbf{x} . In addition, $F(\mathbf{x}, \mathbf{p}, \gamma)$ is differentiable with respect to the circuit parameters \mathbf{p} . In the time domain $\gamma = t$, with t the time. In the frequency domain $\gamma = s$ with s the complex frequency. In the DC domain γ is omitted.

Often a network function is simply defined as a circuit variable (e.g., output voltage) [Ogro94]. In the frequency domain, a network function is often defined as the fraction of two circuit variables (e.g., voltage transfer function) [Bray80]. For convenience of

notation, some or all arguments of F may be omitted. For example $F(\mathbf{p})$ may denote a network function in the DC, frequency or time domain.

Consider the second-order lowpass filter depicted in Figure 3.1. This circuit will be used to illustrate the concepts presented in this chapter as well as in Chapters 4 and 5. Only the filter's frequency domain behavior is examined, under a sinusoidal excitation with $s = j\omega$. The complex voltage transfer function $F(j\omega) = v_o(j\omega)/v_i(j\omega)$ describes the filter's transfer behavior when it is loaded with a zero admittance:

$$F(j\omega, R, L, C) = \frac{1}{1 - \omega^2 LC + j\omega RC}. \quad (3.2)$$

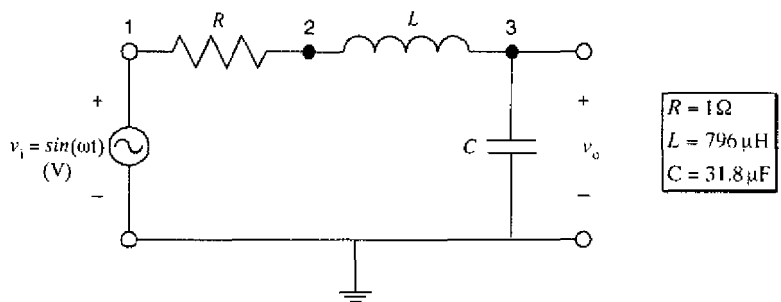


Figure 3.1. Second-order lowpass filter.

The filter's center frequency is $\omega_0 = 1/\sqrt{LC}$ and its quality factor is $Q = \sqrt{L/C}/R$.

With these definitions, (3.2) can be rewritten as

$$F(j\omega, \omega_0, Q) = \frac{1}{1 - \left(\frac{\omega}{\omega_0}\right)^2 + j \frac{\omega}{\omega_0} \frac{1}{Q}}. \quad (3.3)$$

In (3.3) each of the circuit parameters ω_0 and Q is a non linear combination of the parameters R , L and C . Using the component values of Figure 3.1, $\omega_0 = 2\pi \cdot 10^3 \text{ s}^{-1}$ and $Q = 5$. From (3.3) it follows that $|F(j\omega)|_{\max} = Q$, and it is reached for $\omega = \omega_0$. Figure 3.2 plots $|F(j\omega)|$ as a function of frequency $f = \omega/2\pi$.

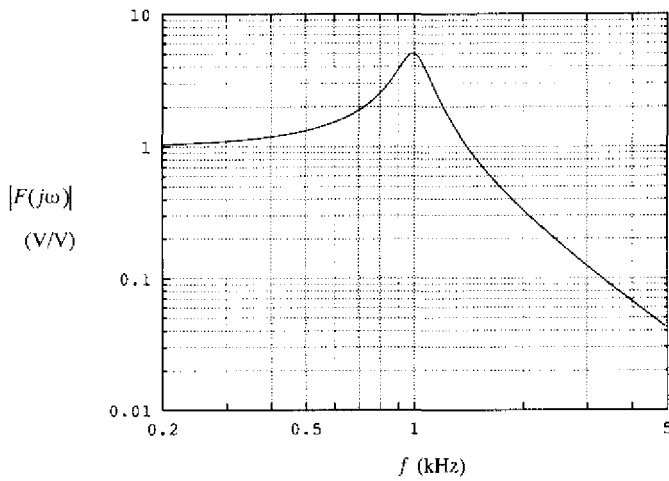


Figure 3.2. Voltage transfer of the second-order lowpass filter.

At this stage, sensitivity can be defined in a more precise manner. The literature on analog testing distinguishes two types of sensitivity: *incremental sensitivity* and *differential sensitivity* [Huer93].

Definition 3.2 Let Δp_i denote an arbitrarily large change from the nominal value p_i of the i^{th} circuit parameter. ΔF is the resulting change of a network function F . The *incremental sensitivity* of F to a change in the i^{th} circuit parameter is defined as

$$S_{p_i}^F \triangleq \frac{\Delta F}{\Delta p_i}.$$

Definition 3.3 Let p_i , Δp_i , F and ΔF be given by Definition 3.2. The *differential sensitivity* of F to a change in p_i is defined as¹

¹ Henceforward, p_i will be used to denote either the i^{th} circuit parameter or its nominal value.

$$s_{p_i}^F \triangleq \lim_{\Delta p_i \rightarrow 0} \frac{\Delta F}{\Delta p_i}.$$

Thus incremental sensitivity is the exact (generally non linear) relation between an arbitrarily large deviation Δp_i and the resulting deviation ΔF . Differential sensitivity is a first-order approximation to the incremental sensitivity: $s_{p_i}^F = \lim_{\Delta p_i \rightarrow 0} \mathcal{S}_{p_i}^F = \frac{\partial F}{\partial p_i}$, where ∂ denotes partial differentiation.

Consider a circuit with a response that is non linear in n parameters. According to Definition 3.3, changes in the n parameters are *linearly* related to the resulting ΔF , if the parameter changes are infinitesimally small:

$$\Delta F = (\mathbf{s}_p^F)^T \Delta \mathbf{p}, \quad (3.4)$$

where $\mathbf{s}_p^F \triangleq (s_{p_1}^F, \dots, s_{p_n}^F)^T$, and $\Delta \mathbf{p} \triangleq (\Delta p_1, \dots, \Delta p_n)^T$. In practice (3.4) provides a good approximation of the cumulative influence on circuit behavior of small (up to about 5 %) changes $\Delta p_1, \dots, \Delta p_n$. However, for larger deviations $\Delta p_1, \dots, \Delta p_n$ (3.4) is not valid. In that case there are several alternative approaches.

Perhaps the most straightforward solution is to take into account higher-order terms of the infinite Taylor expansion of F :

$$F(\mathbf{p}) = F_0 + (\mathbf{s}_p^F)^T \Delta \mathbf{p} + \frac{1}{2} \Delta \mathbf{p}^T \mathbf{H} \Delta \mathbf{p} + \dots \triangleq F_0 + \Delta F. \quad (3.5)$$

According to (3.4), $\Delta F = F(\mathbf{p}) - F_0$ is obtained with arbitrary precision by the inclusion of all higher order terms. The second term of expansion (3.5) is the first-order approximation (3.4) of ΔF . The third term incorporates second-order effects. It provides a first-order compensation for the fact that the differential sensitivities are exact only for a circuit with nominal parameter values \mathbf{p} , whereas in fact the parameters deviate from nominal by the amount $\Delta \mathbf{p}$. \mathbf{H} is commonly called the Hessian

matrix. Its elements are the second-order sensitivities $h_{ij} = \partial^2 F(\mathbf{p}) / \partial p_i \partial p_j$. Methods exist for the efficient calculation of such sensitivities [Bray80].

If more than the first two terms are included, (3.5) is a non linear relation. Then non linear techniques are necessary to solve $\Delta \mathbf{p}$ from measurements of ΔF , and to select test points. The application of higher-order sensitivities is not further discussed in this thesis.

The magnitude of the sensitivities conform Definitions 3.2 and 3.3 depends upon the unit in which F and p_1, \dots, p_n are expressed. To change the sensitivities to scale-free quantities, normalized versions of F and p_1, \dots, p_n are introduced. On the basis of Definition 3.3 *normalized differential sensitivities* are obtained as

$$\widehat{S}_{p_i}^F \triangleq \frac{\partial \ln F}{\partial \ln p_i} = \frac{p_i}{F} \frac{\partial F}{\partial p_i} = \frac{p_i}{F} S_{p_i}^F. \quad (3.6)$$

Normalized incremental sensitivity is likewise defined as $\widehat{S}_{p_i}^F \triangleq \frac{p_i}{F} S_{p_i}^F$.

If the nominal value of either F or p_i is zero, normalization (3.6) cannot be used. This thesis assumes that faults occur in circuit parameters with non-zero nominal values. Thus the influence of, for example, parasitic capacitances (which ideally have zero nominal values) is not taken into account. Under this condition, normalization to the nominal parameter value is not a problem. If F is always non-zero then (3.6) may be used without problems. Otherwise (3.6) can be replaced by the semi-normalized sensitivity

$$\frac{\partial F}{\partial \ln p_i} = p_i S_{p_i}^F.$$

Normalized sensitivities offer some advantages. For example, $\widehat{S}_{p_i}^F$ is independent of the unit of p_i . Furthermore, normalization to F allows the sensitivity of different circuit designs to be compared directly, regardless of, for example, the magnitude of the supply voltage or the input signal. In this sense (3.6) offers a more objective estimate

of the influence of a parameter deviation. In general the choice for a particular type of normalization is application-dependent. A discussion of sensitivity normalization for analog testing is deferred until Section 4.1.1. For the moment it is assumed that (3.6) is used.

3.1.2 Application of Sensitivities

As discussed in the previous section, the differential sensitivities \mathbf{s}_p^f provide a first-order approximation of $\Delta F / \Delta \mathbf{p}$, for example in a region around a certain frequency. Thus \mathbf{s}_p^f is a gradient vector that can be used in *design optimization* to iteratively minimize a certain objective function. For example, [Vlac] discusses the automatic optimization of an analog filter by minimization of a mean-square objective function.

Sensitivity computations can also help in judging the quality of a particular circuit design. A high sensitivity to a particular component parameter means that a slight disturbance of this parameter might cause circuit behavior to deviate outside the *tolerance region* that defines acceptable circuit behavior. This suggests that the component of interest should be manufactured with a high precision. If it is not possible to obtain the necessary precision with a certain production process then the circuit design may be adapted to make it less sensitive to component changes. A commonly used technique for integrated circuits is to let the circuit behavior be determined by the difference in some property of two geometrically close components. The motivation is that fluctuations of the manufacturing process will now introduce highly correlated deviations of the property of interest in both components. By designing the circuit such that the sensitivity to these deviations is equal in magnitude for the two components, but different in sign, process fluctuations practically do not affect the circuit behavior. Because a circuit design that uses this method may be more costly, it should only be applied to the components for which such measures appear necessary. As explained, such components may be detected by observing their sensitivities.

Finally, a sensitivity analysis may be used in circuit testing. Assume that the parameter sensitivities are calculated for all possible input stimuli (e.g., frequencies, voltages). The stimuli that sensitize one of the circuit's outputs to changes in the parameters are associated with the highest sensitivities. Therefore, measurements that are made at the output while those stimuli are applied to the circuit input are most likely to detect deviations of circuit behavior. How circuit sensitivities are used to select these measurements is discussed in detail in Chapters 4 and 5.

3.1.3 Frequency Domain Sensitivities

3.1.3.1 Sensitivity to Magnitude and Phase

In the small-signal frequency domain, a network function F is in general complex-valued. However, in the “real” world the value of F is not measured directly. Instead, measurements of its magnitude and phase are made. Therefore, formulas that relate the sensitivity of the magnitude and phase of F to its complex sensitivity are derived next.

Because a complex (non zero) network function F can be written in terms of its magnitude $|F|$ and phase ϕ_F as

$$F = |F|e^{j\phi_F},$$

it follows that

$$\ln F = \ln|F| + j\phi_F. \quad (3.7)$$

Partial differentiation of both sides of (3.7) to p_i gives

$$\frac{1}{F} \frac{\partial F}{\partial p_i} = \frac{1}{|F|} \frac{\partial |F|}{\partial p_i} + j \cdot \frac{\partial \phi_F}{\partial p_i}. \quad (3.8)$$

Separating the real and imaginary parts of (3.8) results in

$$\begin{cases} \widehat{s}_{p_i}^{|F|} \triangleq \frac{p_i}{|F|} \frac{\partial |F|}{\partial p_i} = p_i \cdot \operatorname{Re} \left(\frac{1}{F} \frac{\partial F}{\partial p_i} \right) \\ \widehat{s}_{p_i}^{\phi_F} \triangleq \frac{p_i}{\phi_F} \frac{\partial \phi_F}{\partial p_i} = \frac{p_i}{\phi_F} \cdot \operatorname{Im} \left(\frac{1}{F} \frac{\partial F}{\partial p_i} \right) \end{cases},$$

or

$$\begin{cases} \widehat{s}_{p_i}^{|F|} = p_i \cdot \operatorname{Re} \left(\frac{1}{F} s_{p_i}^F \right) \\ \widehat{s}_{p_i}^{\phi_F} = \frac{p_i}{\phi_F} \cdot \operatorname{Im} \left(\frac{1}{F} s_{p_i}^F \right) \end{cases}. \quad (3.9)$$

According to (3.9) the sensitivity to $|F|$ and ϕ_F may be obtained directly from the complex-valued differential sensitivity $s_{p_i}^F$ and complex network function F .

In the most straightforward case, p_i in (3.9) is a parameter of a circuit component. To illustrate, Figure 3.3 depicts $\widehat{s}_{p_i}^{|F|}$ for the three components of the low-pass filter example discussed in Section 3.1.1. This sensitivity plot reveals some information about the lowpass filter. For example, $\widehat{s}_R^{|F|}$ is negative, meaning that a small increase in R leads to a decrease in $|V_o|$. Furthermore, at the center frequency $f_0 = 1$ kHz $\widehat{s}_R^{|F|}$ obtains its maximum magnitude, which is about equal to the magnitude of $\widehat{s}_C^{|F|}$, while $\widehat{s}_L^{|F|} = 0$. Thus at $f = f_0$ the simultaneous influence of R and C may be observed. Finally the sensitivity plot reveals that deviations in L and C have a similar (in sign and magnitude) influence on the circuit behavior. This suggests that it is very hard to distinguish the influence of a small deviation in L from a small deviation in C by observing $|V_o(j\omega)/V_i(j\omega)|$. At high frequencies the relative influence of R is negligible, so there the simultaneous influence of L and C may be observed. Thus it seems that there are only *two* independent degrees of freedom for this circuit with *three* components. This hypothesis will be verified in Chapter 4, which continues the analysis of the low-pass filter.

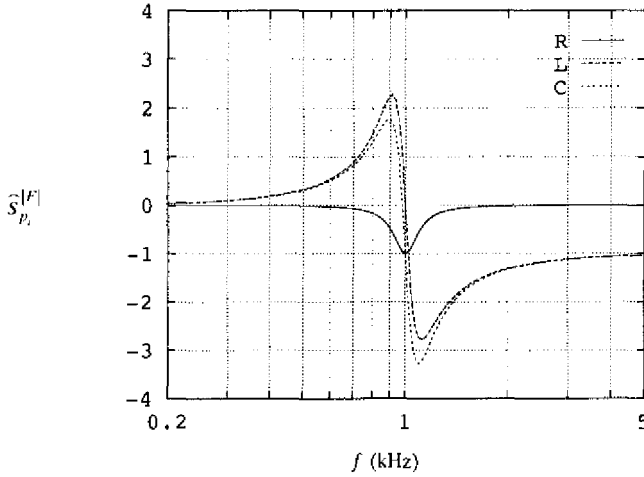


Figure 3.3. Normalized parameter sensitivities of the lowpass filter.

3.1.3.2 Differential/Incremental Sensitivity Relationship

This section considers the special case of a linear circuit with a bilinear network function

$$F(p_i) = \frac{a + bp_i}{c + dp_i} \frac{\Delta N}{D}. \quad (3.10)$$

It is not difficult to show [Fid72] that if (3.10) holds, then

$$S_{p_i}^F = \frac{s_{p_i}^F}{1 + \Delta p_i D' / D}, \quad (3.11)$$

where the prime (') denotes partial differentiation to p_i . This formula provides a relation between differential and incremental sensitivity. According to (3.11), if $s_{p_i}^F = 0$ at a particular frequency, then also $S_{p_i}^F = 0$. This renders the parameter p_i redundant at that frequency. From (3.11) it follows that

$$\Delta F(\mathbf{p}) = \frac{\sum_{i=1}^n s_{p_i}^F \Delta p_i}{1 + \sum_{i=1}^n \Delta p_i D' / D}, \quad (3.12)$$

which is an exact relation, regardless of the magnitude of $\Delta p_1, \dots, \Delta p_n$ [Slam92]. However, for a circuit with non linear components (3.12) is just a first-order approximation, like (3.4). Furthermore, also for some *linear* circuits (3.10) may not be valid (For example, when it contains a gyrator). Thus incremental sensitivities are only applicable for a subclass of linear circuits. A more general way of dealing with (multiple) large parameter deviations is an iterative calculation of differential sensitivities, as explained in the next chapter.

In many situations, relatively small (less than approximately 5 %) deviations from nominal circuit behavior are of interest. In these cases differential sensitivities model with a high precision the influence of multiple parameter deviations, even without iteration. This will be demonstrated with examples at a later stage.

3.1.3.3 Efficient Computation of Sensitivities in the Frequency Domain

In the small-signal frequency domain, circuit behavior is described by a set of linear complex equations

$$\mathbf{Y}(\mathbf{p}, j\omega) \mathbf{x}(\mathbf{p}, j\omega) = \mathbf{w}(\mathbf{p}, j\omega), \quad (3.13)$$

with \mathbf{Y} the complex system matrix, \mathbf{w} the complex excitation vector and \mathbf{x} the vector of circuit variables. \mathbf{Y} may be a nodal admittance matrix, for example.

Differential sensitivities can be computed with repeated circuit simulations, each evaluating the effect of a small disturbance in a single parameter. Suppose only p_j is disturbed by a small deviation Δp_j (for example, 1 %). To obtain the resulting deviation $\Delta \mathbf{x}$, (3.13) is solved twice: once for p_j to obtain \mathbf{x} , and then for $p_j + \Delta p_j$ to obtain $\mathbf{x} + \Delta \mathbf{x}$. Next, the sensitivities are calculated conform Definition 3.3 as

$$s_{p_j}^{\mathbf{x}} \approx \frac{(\mathbf{x} + \Delta \mathbf{x}) - \mathbf{x}}{(p_j + \Delta p_j) - p_j} = \frac{\Delta \mathbf{x}}{\Delta p_j}, \quad (3.14)$$

The procedure must be repeated for all parameters and all frequencies of interest.

There are two disadvantages to (3.14). First, for n parameters and m frequencies, a total of $m(n+1)$ circuit simulations must be performed. Second, $\Delta \mathbf{x}$ and Δp_j are computed by subtracting slightly different numbers. This may result in a significant loss of precision. Because of these disadvantages (3.14) is not used to generate sensitivities.

A method that computes *precise* differential sensitivities is based on (3.11). It uses parameter perturbations of arbitrary large size [Fid175]. However, also that approach is not efficient. Therefore the method of adjoint equations [Dire69], [Bray80], [Swar87], [Ogro94] is used in this thesis. Next it is shown how this approach computes the sensitivity of a single circuit variable x_i to *all* circuit parameters.

Differentiation of (3.13) to p_j yields

$$\frac{\partial \mathbf{Y}}{\partial p_j} \mathbf{x} + \mathbf{Y} \frac{\partial \mathbf{x}}{\partial p_j} = \frac{\partial \mathbf{w}}{\partial p_j},$$

so

$$\frac{\partial \mathbf{x}}{\partial p_j} = -\mathbf{Y}^{-1} \left(\frac{\partial \mathbf{Y}}{\partial p_j} \mathbf{x} - \frac{\partial \mathbf{w}}{\partial p_j} \right), \quad (3.15)$$

where the arguments of \mathbf{Y} , \mathbf{x} , and \mathbf{w} have been omitted for clarity. Now

$$\frac{\partial x_i}{\partial p_j} = -\mathbf{e}_i^T \mathbf{Y}^{-1} \left(\frac{\partial \mathbf{Y}}{\partial p_j} \mathbf{x} - \frac{\partial \mathbf{w}}{\partial p_j} \right), \quad (3.16)$$

where \mathbf{e}_i is the i^{th} unit vector. At this stage, define \mathbf{x}_a as the solution of

$$\mathbf{Y}^T \mathbf{x}_a = -\mathbf{e}_i, \quad (3.17)$$

which is called the *adjoint* system of equations. With (3.17), (3.16) can finally be written as

$$\frac{\partial x_i}{\partial p_j} = s_{p_j}^{x_i} = \mathbf{x}_a^T \left(\frac{\partial \mathbf{Y}}{\partial p_j} \mathbf{x} - \frac{\partial \mathbf{w}}{\partial p_j} \right). \quad (3.18)$$

This expression is used to calculate $s_{p_i}^x$ at m frequencies. The procedure at one frequency can be summarized as follows:

1. \mathbf{x} and \mathbf{x}_a are obtained by solving the original system (3.13) and the adjoint system (3.17). Because both systems are based on the nominal parameter values, these systems need to be solved only once for each frequency of interest. For m frequencies this takes $2m$ circuit simulations. The adjoint system (3.17) can be efficiently solved by using the decomposition $\mathbf{Y} = \mathbf{LU}$ which is available after (3.13) is solved. With this LU decomposition (3.17) can be written as

$$\mathbf{U}^T \mathbf{L}^T \mathbf{x}_a = -\mathbf{e}_i,$$

which is solved by a simple forward and backward substitution.

2. The matrix $\partial \mathbf{Y} / \partial p_j$ and the vector $\partial \mathbf{w} / \partial p_j$ are formed for every p_j , and the right-hand side of (3.18) is evaluated. Advantage can be taken of the special zero-non zero structure of $\partial \mathbf{Y} / \partial p_j$ and $\partial \mathbf{w} / \partial p_j$. For example, if p_j is not a parameter of an excitation, then $\partial \mathbf{w} / \partial p_j = 0$.

For many circuit elements, the right-hand side of (3.18) is just a multiplication of elements of \mathbf{x} and \mathbf{x}_a . For example, for a capacitor C , connected between nodes i and j , $s_c^x = j\omega(V_i - V_j)(V_{i,a} - V_{j,a})$. V_i and V_j are the voltage of nodes i and j , respectively. $V_{i,a}$ and $V_{j,a}$ are the corresponding adjoint voltages. The precision with which s_c^x is computed is about equal to the precision of a circuit simulation.

Essentially no work is involved in step 2. Therefore the total work load involved in computing s_p^x at m frequencies amounts to $2m$ circuit simulations.

Next, consider a sensitivity s_p^f , where $F(\mathbf{x}, \mathbf{p}) = \mathbf{d}^T \mathbf{x}$, with \mathbf{d} an arbitrary vector:

$$s_{p_i}^F = \frac{\partial F}{\partial p_i} = \frac{\partial(\mathbf{d}^T \mathbf{x})}{\partial p_i} = \mathbf{d}^T \frac{\partial \mathbf{x}}{\partial p_i}.$$

By substituting (3.15) into this expression it follows that $s_{p_i}^F$ can be computed from (3.18) if (3.17) (the adjoint system) is replaced by

$$\mathbf{Y}^T \mathbf{x}_a = -\mathbf{d}.$$

The case that F (differentiable to \mathbf{p}) is a *non linear* function of circuit variables \mathbf{x} is not discussed here. It is treated in [Vlac94].

3.1.4 Time Domain Sensitivities

Time domain sensitivities may be computed for the general case of a non-linear, time-dependent circuit described by (3.1). This section describes an approach that computes the sensitivities $s_{p_i}^{\mathbf{x}}$ of *all* circuit variables to a single parameter p_i . These sensitivities are obtained in parallel with the computation of the circuit's transient response.

First the DC solution of (3.1) is computed by solving the system

$$\begin{aligned} -\mathbf{E}\mathbf{x}_0 &= \mathbf{0} \\ \mathbf{f}(\mathbf{q}_0, \mathbf{x}_0, \mathbf{w}_0, p_i, 0) &= \mathbf{0} \end{aligned} \quad (3.19)$$

which is derived from (3.1) by setting $\mathbf{q}^1 = \mathbf{0}$. The solution $(\mathbf{q}_0, \mathbf{x}_0)$ of (3.19) is found by *Newton-Raphson* iteration. In general, this technique finds the solution of a non linear system $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ by iteratively solving the Newton-Raphson equation [Vlac94]

$$\mathbf{J}^k \Delta \mathbf{x}^k = -\mathbf{g}(\mathbf{x}^k), \quad (3.20)$$

where \mathbf{J}^k is the Jacobian of \mathbf{g} , with $(\mathbf{J}^k)_{ij} = \partial g_i / \partial x_j^k$. Iteration starts with estimate \mathbf{x}^0 . After $\Delta \mathbf{x}^k$ is computed in the k^{th} iteration, \mathbf{x}^{k+1} is found as $\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta \mathbf{x}^k$, and the next iteration starts. If $\Delta \mathbf{x}^k$ is below a certain threshold iteration stops. In the case of (3.19), the Newton-Raphson equation is

$$\begin{bmatrix} \mathbf{0} & -\mathbf{E} \\ \frac{\partial \mathbf{f}}{\partial \mathbf{q}_0} & \frac{\partial \mathbf{f}}{\partial \mathbf{x}_0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{q}_0 \\ \Delta \mathbf{x}_0 \end{bmatrix} = - \begin{bmatrix} -\mathbf{E} \mathbf{x} \\ \mathbf{f} \end{bmatrix}, \quad (3.21)$$

which is solved by iteration (for simplicity it is assumed that the excitations \mathbf{w} do not depend on p_j).

After the DC solution $(\mathbf{q}_0, \mathbf{x}_0)$ is obtained, the DC sensitivities are computed. Differentiation of (3.19) with respect to p_j results in the linear system:

$$\begin{bmatrix} \mathbf{0} & -\mathbf{E} \\ \frac{\partial \mathbf{f}}{\partial \mathbf{q}_0} & \frac{\partial \mathbf{f}}{\partial \mathbf{x}_0} \end{bmatrix} \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{z}_0 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -\frac{\partial \mathbf{f}}{\partial p_j} \end{bmatrix}, \quad (3.22)$$

where $\mathbf{y}_0 = \partial \mathbf{q}_0 / \partial p_j$ and $\mathbf{z}_0 = \partial \mathbf{x}_0 / \partial p_j$ are introduced to simplify the notation. Note that (3.22) and (3.21) have the same system matrix. Therefore (3.22) can be solved efficiently by using the LU factorization of the Jacobian that was computed at the last iteration of (3.21).

Now the sensitivity of (3.1) to p_j is computed. Differentiation of (3.1) to p_j gives the linear, time-varying system

$$\begin{aligned} \mathbf{y}' - \mathbf{E} \mathbf{z} &= \mathbf{0}; \quad \mathbf{y}_0 = \mathbf{y}(0) \\ \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \mathbf{y} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{z} + \frac{\partial \mathbf{f}}{\partial p_j} &= \mathbf{0} \end{aligned}, \quad (3.23)$$

where $\mathbf{y} = \partial \mathbf{q} / \partial p_j$ and $\mathbf{z} = \partial \mathbf{x} / \partial p_j$. At each time point the circuit sensitivities are obtained by solving (3.23) after the original system (3.1) is solved. Suppose, for example, that a k^{th} order Backward Differentiation Formula (BDF) is used, with the corrector

$$(\mathbf{q}')^{n+k} = -\frac{1}{\Delta t} \sum_{i=0}^{k-1} a_i \mathbf{q}_{n+k-i}, \quad (3.24)$$

where the coefficients a_i depend upon the order k of the BDF formula. After substituting (3.24) into (3.1), the Newton-Raphson equation is derived as

$$\begin{bmatrix} -\frac{a_0}{\Delta t} \mathbf{1} & -\mathbf{E} \\ \frac{\partial \mathbf{f}}{\partial \mathbf{q}} & \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{q}_{n+k} \\ \Delta \mathbf{x}_{n+k} \end{bmatrix} = - \begin{bmatrix} -\frac{1}{\Delta t} \sum_{i=0}^{k-1} a_i \mathbf{q}_{n+k-i} - \mathbf{E} \mathbf{x}_{n+k} \\ \mathbf{f}(\mathbf{q}_{n+k}, \mathbf{x}_{n+k}, \mathbf{w}_{n+k}, p_j, t_{n+k}) \end{bmatrix}. \quad (3.25)$$

Iteration on this system provides the solution $(\mathbf{q}_{n+k}, \mathbf{x}_{n+k})$.

Substituting a k^{th} order BDF formula in (3.23) gives the *linear* system

$$\begin{bmatrix} -\frac{a_0}{\Delta t} \mathbf{1} & -\mathbf{E} \\ \frac{\partial \mathbf{f}}{\partial \mathbf{q}} & \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \end{bmatrix} \begin{bmatrix} \mathbf{y}_{n+k} \\ \mathbf{z}_{n+k} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t} \sum_{i=0}^{k-1} a_i \mathbf{y}_{n+k-i} \\ -\frac{\partial \mathbf{f}}{\partial p_j} \end{bmatrix}. \quad (3.26)$$

Thus (3.25) and (3.26) have the same system matrix. The LU factorization of this matrix is available after (3.25) is iteratively solved. Then a simple forward and backward substitution solves (3.26). For each parameter the right-hand side of (3.26) is different and the forward and backward substitution must be repeated.

Time domain sensitivities can also be computed with a version of the adjoint equations method [Dire69], [Vlac94]. This method obtains the sensitivity of a single circuit variable to *all* circuit parameters at a time point t_f in one pass. This is useful since in analog testing the number of observed circuit variables is in most cases considerably smaller than the number of parameters. However, the adjoint equations methods has distinct disadvantages when applied to the time domain, as explained next.

For example, the adjoint network must be integrated *backwards* in time, so it cannot be solved simultaneously with the normal network. Rather, the responses of the normal network at all selected time points need to be stored before the adjoint network can be solved. The storage requirements may be prohibitive. An additional problem is that the time steps of the adjoint network and the normal network are in general not identical.

This necessitates interpolation between time steps of the stored responses of the original circuit. Finally, although the sensitivity at a time t_f is computed, the sensitivities at time points $t < t_f$ are not available. Because of these difficulties computing time domain sensitivities with the adjoint method is more difficult than with the method described above.

3.2 The Singular Value Decomposition

The Singular Value Decomposition (SVD) [Wilk71], [Dewi88], [Golu89] is a well-known complete matrix decomposition with many applications. It is used in mathematical proofs [Golu89], signal processing [Depr88], and principal component analysis [Joll86], for example.

3.2.1 The Spectral Theorem

According to the spectral theorem any hermitian matrix may be diagonalized. The theorem has applications in, for example, quadratic and bilinear forms. The spectral theorem is presented² as a preliminary step before the SVD is introduced.

Theorem 3.1 (*Spectral Theorem*) Let \mathbf{H} be a hermitian $n \times n$ matrix. Then there exist an $n \times n$ unitary matrix \mathbf{V} and n real numbers $\lambda_1, \dots, \lambda_n$ such that

$$\mathbf{V}^* \mathbf{H} \mathbf{V} = \mathbf{\Lambda} = \text{diag}\{\lambda_1, \dots, \lambda_n\}, \quad (3.27)$$

where $\lambda_1, \dots, \lambda_n$ are the eigenvalues of \mathbf{H} , and the columns of \mathbf{V} are the eigenvectors of \mathbf{H} .

² Proofs of the presented theorems may be found in standard text books (e.g., [Robi91], [Golu89]) and are not repeated here.

For the special case $\mathbf{H} = \mathbf{A}^* \mathbf{A}$, with \mathbf{A} a complex $m \times n$ matrix, the eigenvalues of \mathbf{H} are positive real numbers. Hence there exists an ordering $\lambda_1 \geq \dots \geq \lambda_n \geq 0$, where $\lambda_n > 0$ iff $\text{rank}(\mathbf{A}) = n$.

If \mathbf{A} is real-valued then \mathbf{H} is symmetric and \mathbf{V} is orthogonal.

Applications of the spectral theorem to data analysis are discussed comprehensively in [Joll86]. There the theorem is used to find the important trends (*principal components*) in a data set. Next, a simple example in \mathbb{R}^2 gives a geometric interpretation of this application. The discussed principles are closely related to the method of test-point selection that will be presented in Chapter 4.

Consider n observations of two statistical variables³ X and Y . The i^{th} observation obtains a *data point* (x_i, y_i) . After n data points are collected the means μ_x and μ_y are computed (using (B.5), Appendix B). Then the data is normalized by subtracting, for all data points, μ_x from x_i and μ_y from y_i . Figure 3.4 shows a data set that is normalized (in fact it is translated) in this manner. Each dot represents a data point. The data set is centered at the origin, due to the normalization. Stacking the 7 data points gives a 7×2 data matrix \mathbf{M} .

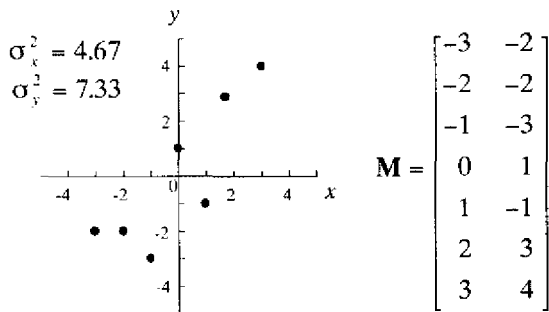


Figure 3.4. Normalized data set and corresponding data matrix.

³ The physical significance of the variables is not important here.

To estimate the variance and covariance of X and Y , the symmetric 2×2 covariance matrix \mathbf{C} of \mathbf{M} (conform (B.6), Appendix B) is calculated as

$$\mathbf{C} = \frac{1}{n-1} \mathbf{M}^T \mathbf{M} = \frac{1}{6} \begin{bmatrix} 28 & 30 \\ 30 & 44 \end{bmatrix}. \quad (3.28)$$

The diagonal elements of \mathbf{C} are the variances σ_x^2 and σ_y^2 , shown Figure 3.4. Each off-diagonal element of \mathbf{C} is the covariance σ_{xy} . Because σ_{xy} is comparable in magnitude to σ_x^2 and σ_y^2 , x and y are strongly correlated. The correlation is visible in Figure 3.4, where the data is spread approximately around the line $x = y$. For a more precise examination of the properties of the data set, the spectral decomposition is used.

The spectral decomposition of \mathbf{C} is found by calculating its eigenvalues and eigenvectors. The eigenvalues λ_1 and λ_2 are the solutions of the characteristic equation $|\mathbf{C} - \lambda \mathbf{I}| = 0$. The eigenvectors \mathbf{v}_1 and \mathbf{v}_2 are the solutions of $\mathbf{C}\mathbf{v} = \lambda\mathbf{v}$ for $\lambda = \lambda_1$ and $\lambda = \lambda_2$, respectively. After the eigenvectors are normalized to unit length, the spectral decomposition of \mathbf{C} is obtained as

$$\mathbf{C} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T = \begin{bmatrix} 0.6 & 0.8 \\ 0.8 & -0.6 \end{bmatrix} \begin{bmatrix} 11.17 & 0 \\ 0 & 0.83 \end{bmatrix} \begin{bmatrix} 0.6 & 0.8 \\ 0.8 & -0.6 \end{bmatrix}. \quad (3.29)$$

Figure 3.5 depicts decomposition (3.29). The two arrows point in the direction of \mathbf{v}_1 and \mathbf{v}_2 . The length of the arrows is made proportional to the magnitude of λ_1 and λ_2 .

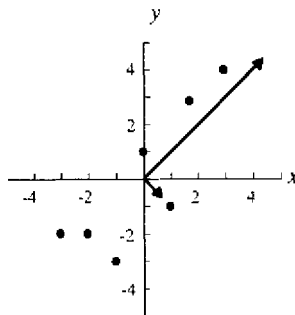


Figure 3.5. Graphical depiction of the spectral decomposition of \mathbf{M} .

To determine the variance in the direction of \mathbf{v}_1 and \mathbf{v}_2 , \mathbf{M} is first post-multiplied by \mathbf{V} . This transforms \mathbf{M} into a new matrix \mathbf{M}' according to

$$\mathbf{M}' = \mathbf{M}\mathbf{V}. \quad (3.30)$$

Row i of \mathbf{M}' contains the coordinates of the i^{th} data point, with respect to the orthonormal base formed by \mathbf{v}_1 and \mathbf{v}_2 . The covariance matrix \mathbf{C}' of \mathbf{M}' is

$$\begin{aligned} \mathbf{C}' &= \frac{1}{n-1}(\mathbf{M}\mathbf{V})^T \mathbf{M}\mathbf{V} = \mathbf{V}^T \left(\frac{1}{n-1} \mathbf{M}^T \mathbf{M} \right) \mathbf{V} \\ &= \mathbf{V}^T \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T \mathbf{V} \\ &= \mathbf{\Lambda} \end{aligned} \quad (3.31)$$

where (3.29) and the orthogonality of \mathbf{V} were used. Because \mathbf{C}' is a diagonal matrix, the columns of \mathbf{M}' correspond with new, uncorrelated variables. These are denoted by x' and y' . Figure 3.6 provides a geometrical interpretation of the transformed data set. It is obtained by rotating and mirroring the data set of Figure 3.5. Since \mathbf{V} is orthogonal, the data set is not deformed, and thus no information is lost. As a result, the total variance $\sigma_x^2 + \sigma_y^2$ is equal to $\lambda_1 + \lambda_2$ [Jenn92]. According to (3.31), $\sigma_{x'}^2 = \lambda_1$ and $\sigma_{y'}^2 = \lambda_2$. In the general case, λ_i is the part of the total variance that is associated with \mathbf{v}_i . Therefore \mathbf{v}_i is said to define the i^{th} *principal direction*. Like this, the i^{th} *principal component* is defined as a vector with length λ_i , pointing in the i^{th} *principal direction*. Thus the arrows in Figure 3.5 are the principal components of the original data set.

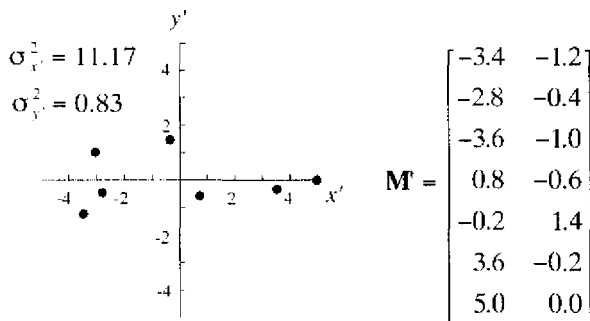


Figure 3.6. Transformed data set and corresponding data matrix.

According to Theorem 3.3 $\lambda_1 \geq \dots \geq \lambda_n \geq 0$, so the largest variance is in the direction of \mathbf{v}_1 . In the example λ_1 accounts for 93 % of the total variance. Geometrically, this means that the data set is spread out mainly in the direction of \mathbf{v}_1 , as confirmed by Figure 3.5. The fact that the most important information is contained in the first columns of \mathbf{M}' is used as an heuristic for test-point selection in Chapter 4.

3.2.2 The Singular Value Decomposition

This section introduces the singular value decomposition (SVD), and summarizes some of its properties.

Theorem 3.2 (*Singular value decomposition*) Let \mathbf{A} be a complex $m \times n$ matrix. Then there exist:

- a unitary $m \times m$ matrix \mathbf{U} ,
- a unitary $n \times n$ matrix \mathbf{V} ,
- $p = \min(m, n)$ real numbers $w_1 \geq \dots \geq w_r > w_{r+1} = \dots = w_p = 0$,
($0 \leq r \leq p$), called the *singular values* of \mathbf{A} ,

such that

$$\mathbf{U}^* \mathbf{A} \mathbf{V} = \mathbf{W}, \quad (3.32)$$

where \mathbf{W} is the real $m \times n$ matrix

$$\mathbf{W} = \left[\begin{array}{c|c} \mathbf{W}_1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right], \quad (3.33)$$

with $\mathbf{W}_1 = \text{diag}\{w_1, \dots, w_r\}$ a real $r \times r$ diagonal matrix.

As a result

$$\mathbf{A} = \mathbf{U}_1 \mathbf{W}_1 \mathbf{V}_1^*, \quad (3.34)$$

where

$$\mathbf{U} \stackrel{\Delta}{=} [\mathbf{U}_1 \mid \mathbf{U}_2] = [\mathbf{u}_1 \dots \mathbf{u}_r \mid \mathbf{u}_{r+1} \dots \mathbf{u}_m], \quad (3.35)$$

and

$$\mathbf{V} \stackrel{\Delta}{=} [\mathbf{V}_1 \mid \mathbf{V}_2] = [\mathbf{v}_1 \dots \mathbf{v}_r \mid \mathbf{v}_{r+1} \dots \mathbf{v}_n]. \quad (3.36)$$

The singular values w_1, \dots, w_p are the lengths of the semi-axis of the hyper ellipsoid defined by $\mathbf{y} = \mathbf{A}\mathbf{x}$, with $\|\mathbf{x}\|_2 = 1$.

The SVD is closely connected to the spectral decomposition (3.27). Define the hermitian matrices $\mathbf{H}_1 = \mathbf{A}^* \mathbf{A}$ and $\mathbf{H}_2 = \mathbf{A} \mathbf{A}^*$. If the SVD of \mathbf{A} is given by (3.32), then $\mathbf{H}_1 = \mathbf{V} \mathbf{W}^2 \mathbf{V}^*$ and $\mathbf{H}_2 = \mathbf{U} \mathbf{W}^2 \mathbf{U}^*$. Thus the spectral decompositions of \mathbf{H}_1 and \mathbf{H}_2 are implicitly obtained by computing the SVD of \mathbf{A} . The eigenvectors of \mathbf{H}_1 and \mathbf{H}_2 are the columns of \mathbf{V} and \mathbf{U} , respectively. The eigenvalues of \mathbf{H}_1 and \mathbf{H}_2 are the (diagonal) elements w_1^2, \dots, w_n^2 of \mathbf{W}^2 .

From $\mathbf{A}^* \mathbf{U} = \mathbf{V} \mathbf{W}$ it follows that $(\mathbf{v}_1, \dots, \mathbf{v}_r)$ is an orthonormal base for $\mathbf{R}(\mathbf{A}^*)$ (the *range* of \mathbf{A}^* , which is the *domain* of \mathbf{A}). In terms of bases, (3.30) is an orthonormal transformation of the unit base $(\mathbf{e}_1, \mathbf{e}_2)$ of the domain space \mathbf{R}^2 of \mathbf{A} . The base of \mathbf{R}^2 is changed to the base $(\mathbf{v}_1, \mathbf{v}_2)$. These base vectors are the eigenvectors of $\mathbf{M}^T \mathbf{M}$, associated with the eigenvalues w_1^2, w_2^2 of $\mathbf{M}^T \mathbf{M}$ (where w_1, w_2 are the singular values of \mathbf{M}). Most of the variance in the transformed data set of Figure 3.6 is in the direction of the first base vector, because \mathbf{v}_1 is associated with the largest singular value.

Similarly, from $\mathbf{A} \mathbf{V} = \mathbf{U} \mathbf{W}$ it follows that $(\mathbf{u}_1, \dots, \mathbf{u}_r)$ is an orthonormal base for $\mathbf{R}(\mathbf{A})$.

The SVD of a matrix \mathbf{A} provides information about \mathbf{A} . For example, $\text{rank}(\mathbf{A}) = r$. Furthermore, since \mathbf{U} and \mathbf{V} are unitary, the *norm* of \mathbf{A} is a function of the singular values. For example, the ℓ_2 -*norm* of \mathbf{A} is

$$\|\mathbf{A}\|_2 \triangleq \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = w_1. \quad (3.37)$$

The *condition number* of \mathbf{A} is

$$c(\mathbf{A}) \triangleq \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \bigg/ \min_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \frac{w_1}{w_p}, \text{ with } p \leq r. \quad (3.38)$$

It follows that $c(\mathbf{A}^* \mathbf{A}) = c(\mathbf{A} \mathbf{A}^*) = \{c(\mathbf{A})\}^2$. This means that numerical precision is lost when the products $\mathbf{A}^* \mathbf{A}$ or $\mathbf{A} \mathbf{A}^*$ are formed. This results in inaccuracies when the eigenvectors and eigenvalues of $\mathbf{A}^* \mathbf{A}$ or $\mathbf{A} \mathbf{A}^*$ are calculated directly (in the manner of Section 3.2.1). Thus the SVD is not computed like this. Instead, an approach is used that does not explicitly form the products $\mathbf{A}^* \mathbf{A}$ or $\mathbf{A} \mathbf{A}^*$. This is explained in Appendix A.

The next section presents some applications of the SVD. The concepts presented there are applied to analog testing in Chapters 4 and 5.

3.2.3 Applications of the SVD

According to Theorem 3.2, a complex $m \times n$ matrix \mathbf{A} of rank r may be written as

$$\mathbf{A} = \sum_{i=1}^r w_i \mathbf{u}_i \mathbf{v}_i^* \triangleq \mathbf{A}^{(1)} + \cdots + \mathbf{A}^{(r)}, \quad (3.39)$$

where r is the number of nonzero singular values, with $r \leq \min(m, n)$. The vectors \mathbf{u}_i and \mathbf{v}_i are defined by (3.35) and (3.36), respectively. Furthermore, $\text{rank}(\mathbf{A}^{(i)}) = 1$, for $i = 1, \dots, r$.

In (3.39), \mathbf{A} is formed from the first r triplets $(\mathbf{u}_i, \mathbf{v}_i, w_i)$. This means that \mathbf{A} can be completely specified by $r(m + n + 1)$ numbers. In its original, unfactored form \mathbf{A} is completely specified by $m \cdot n$ numbers. This implies that in the case $r \ll \min(m, n)$, \mathbf{A} is more efficiently stored in its factored form (3.39). Thus the SVD is used for *lossless* data compression.

Suppose that \mathbf{A} is obtained from measurements of a system with a limited number of parameters, causing $\text{rank}(\mathbf{A}) < \min(m, n)$ (an example is given in Section 7.3). However, \mathbf{A} may have full numerical rank due to measurement noise, causing $r = \min(m, n)$. In this case the SVD can be used to approximate \mathbf{A} with a matrix \mathbf{A}_k of precise rank k in an optimal manner. This is formally stated by the following theorem.

Theorem 3.3 Let the SVD of a complex $m \times n$ matrix \mathbf{A} be given by (3.39).

Furthermore define $\mathbf{A}_k = \sum_{i=1}^k w_i \mathbf{u}_i \mathbf{v}_i^*$. Then for $0 \leq k < r = \text{rank}(\mathbf{A})$

and for any complex $m \times n$ matrix \mathbf{B}

$$\min_{\text{rank}(\mathbf{B})=k} \|\mathbf{A} - \mathbf{B}\|_2 = \|\mathbf{A} - \mathbf{A}_k\|_2 = w_{k+1}. \quad (3.40)$$

According to this theorem, the best rank- k approximation of \mathbf{A} is obtained by including only the first k terms in (3.39). The ℓ_2 -norm of the error matrix $\mathbf{A} - \mathbf{A}_k$ is w_{k+1} . Thus w_{k+1} is a measure of the precision with which \mathbf{A}_k approximates \mathbf{A} . For example, suppose that \mathbf{A} is obtained from measurements, as discussed. If the measurements cause a maximum imprecision of ϵ in the elements of \mathbf{A} , a good heuristic is to select k in order to let $w_{k+1} \approx \epsilon$.

Next the SVD is used to find the *least-squares solution* of a system of m linear equations with n variables.

Definition 3.4 Consider the system $\mathbf{Ax} = \mathbf{b}$, where \mathbf{A} is a complex $m \times n$ matrix of rank $r \leq \min(m, n)$, \mathbf{x} is a complex n vector and \mathbf{b} is a complex m vector. The *least-squares approximation* $\hat{\mathbf{x}}$ of \mathbf{x} minimizes $\|\mathbf{b} - \mathbf{Ax}\|_2$.

Appendix C shows that the problem can be expressed in terms of the normal equations

$$\mathbf{A}^+ \mathbf{A} \mathbf{x} = \mathbf{A}^+ \mathbf{b}. \quad (3.41)$$

Substituting the SVD (3.34) of \mathbf{A} in (3.41) gives

$$\mathbf{x} = \mathbf{A}^+ \mathbf{b}, \quad (3.42)$$

where

$$\mathbf{A}^+ \triangleq \mathbf{V}_1 \mathbf{W}_1^{-1} \mathbf{U}_1'. \quad (3.43)$$

The inverse \mathbf{A}^+ is also called the *Moore-Penrose inverse*. It can be proven that (3.42) is the smallest minimizer of $\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2$ [Golub89]. In the special case where $r = n < m$, $\mathbf{V}_1 = \mathbf{V}$, and $\mathbf{A}^+ \mathbf{A} = \mathbf{I}$. Thus (3.43) reduces to the left-inverse of \mathbf{A} , conform (C.5). Alternatively, when $r = m > n$, $\mathbf{U}_1 = \mathbf{U}$, and $\mathbf{A} \mathbf{A}^+ = \mathbf{I}$. Thus (3.43) reduces to the right-inverse of \mathbf{A} , conform (C.6). Finally, if $r = m = n$ then $\mathbf{V}_1 = \mathbf{V}$, $\mathbf{W}_1 = \mathbf{W}$ and $\mathbf{U}_1 = \mathbf{U}$, so $\mathbf{A}^+ = \mathbf{V} \mathbf{W} \mathbf{U}' = \mathbf{A}^{-1}$. A geometrical interpretation of the least-squares approximation is provided Appendix C.

Note that (3.42) obtains the least-squares solution in a numerically stable manner, because the normal equations (C.4) are not formed.

Finally, the advantage of the SVD for detecting near rank-deficiency will be briefly discussed. Let the SVD of \mathbf{A} be given by (3.32), where $r = \text{rank}(\mathbf{A})$. It can be shown [Golub89] that

$$|\sigma_k - \hat{\sigma}_k| < \epsilon \sigma_1, \quad (3.44)$$

where $\hat{\sigma}_k$ is the computed value of σ_k , for $k = 1, \dots, n$, and ϵ is a small multiple of the machine precision. Thus all singular values are computed with a high accuracy. This means that near rank-deficiency of \mathbf{A} cannot escape detection when the SVD of \mathbf{A} is used.

The rank of a complex $m \times n$ matrix \mathbf{A} may also be detected with other decompositions, like for instance the QR decomposition

$$\mathbf{A}\Pi = \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad (3.45)$$

where \mathbf{Q} is unitary $m \times m$ matrix, \mathbf{R}_{11} is a $r \times r$ upper-triangular matrix of rank r , and Π is a permutation. The QR algorithm can detect rank deficiency if, in the course of the computation of (3.45), a diagonal element of the completed part of \mathbf{R}_{11} becomes small. The problem is, however, that a matrix may be nearly rank-deficient without such a diagonal element becoming small. The result is that QR decomposition with column pivoting is by itself not as reliable as the SVD for determining rank deficiency. An example may be found in [Golu89].

4

Using the SVD to Test Analog Circuits

This chapter presents a test method for analog circuits based on the techniques of Chapter 3. Section 4.1 introduces a linear circuit model that uses a normalized form of differential sensitivities. Also it briefly describes how selection of measurements and testable parameters results in a row and column reduction of the linear model. Section 4.2 discusses the influence of measurement errors on the predictions made with the linear model. Formulas are derived that relate normally distributed measurement errors to errors in the predictions. Section 4.3 presents a direct test-point selection method. It will appear that the SVD offers distinct advantages, when applied to analog testing. For example, it is not necessary to select a subset of testable circuit parameters. With the SVD the method of data analysis that was presented in Section 3.2.1 can be extended to a direct method for test-point selection.

4.1 Introduction

4.1.1 The Linear Circuit Model

Section 3.1.1 introduced differential sensitivities, which form a linear relation between parameter deviations $\Delta p_1, \dots, \Delta p_n$ and the resulting deviation ΔF of a network function, according to (3.4). For example, using normalized sensitivities (3.6),

$$\frac{\Delta F(\omega_i)}{F(\omega_i)} = \sum_{j=1}^n \hat{s}_{p_j}^{F(\omega_i)} \frac{\Delta p_j}{p_j}. \quad (4.1)$$

Figure 3.3 shows the normalized sensitivities $\hat{s}_{p_i}^{|F|}$ for the circuit of Figure 3.1, where F is the voltage transfer function. Consider the influence of L and C on $|F|$, given by

$$\begin{aligned} \frac{\Delta|F|}{|F|} &= \hat{s}_L^{|F|} \frac{\Delta L}{L} + \hat{s}_C^{|F|} \frac{\Delta C}{C} \\ &= \left(\frac{L}{|F|} \frac{\partial|F|}{\partial L} \right) \frac{\Delta L}{L} + \left(\frac{C}{|F|} \frac{\partial|F|}{\partial C} \right) \frac{\Delta C}{C}. \end{aligned} \quad (4.2)$$

Although this is not completely clear from Figure 3.3, $\hat{s}_L^{|F|}$ and $\hat{s}_C^{|F|}$ do not decrease to zero for high frequencies but asymptotically converge to a constant value. Thus the relative influence of L and C is constant for high frequencies. This suggests that it does not matter if the influence of these components is considered at 10 kHz, or at 20 kHz, for example.

Now the crucial question is: Does this imply that the deviation $\Delta|F|$ can be measured with equal precision at 10 kHz and 20 kHz? To examine this question consider the deviation $\Delta|F|$,

$$\Delta|F| = \left(L \frac{\partial|F|}{\partial L} \right) \frac{\Delta L}{L} + \left(C \frac{\partial|F|}{\partial C} \right) \frac{\Delta C}{C}, \quad (4.3)$$

which follows directly from (4.2).

Of course $\Delta|F|$ cannot be measured with infinite precision, as measurement errors must be taken into account. Assume for the moment that the maximum measurement error is e_{\max} . Thus in the worst case

$$\begin{aligned} \Delta|F|_{\text{meas}} &= \Delta|F| + e_{\max} \\ &= \left(L \frac{\partial|F|}{\partial L} \right) \frac{\Delta L}{L} + \left(C \frac{\partial|F|}{\partial C} \right) \frac{\Delta C}{C} + e_{\max}, \end{aligned} \quad (4.4)$$

where $\Delta|F|_{\text{meas}}$ denotes the measured value of $\Delta|F|$. It is reasonable to demand $\Delta|F| > e_{\max}$, in order to be able to measure $\Delta|F|$ with sufficient accuracy.

According to Figure 3.2, the nominal output voltage, and thus $|F|$, decreases to zero for high frequencies. As discussed, $\widehat{s}_L^{F(\omega_i)}$ and $\widehat{s}_C^{F(\omega_i)}$ remain at a constant value for high frequencies. Using the definition

$$\widehat{s}_{p_i}^{F(\omega_i)} = \frac{p_j}{|F(\omega_i)|} \frac{\partial |F(\omega_i)|}{\partial p_j} \quad (4.5)$$

it follows that,

$$\begin{cases} \lim_{\omega \rightarrow \infty} L \frac{\partial |F|}{\partial L} = 0 \\ \lim_{\omega \rightarrow \infty} C \frac{\partial |F|}{\partial C} = 0 \end{cases}, \quad (4.6)$$

because $p_j \neq 0$ in (4.4). Thus, according to (4.4) and (4.6), the influence of deviations in L and C becomes less measurable for increasing frequencies. This is not apparent from the sensitivities $s_L^{F|}$ and $s_C^{F|}$. Thus using normalized sensitivities (3.6) to select measurements does not necessarily lead to useful results. It is *not* apparent from these sensitivities that measurements of the deviation $\Delta|F|$ are more precise at 10 kHz than at 20 kHz, for example.

This problem may be remedied by including the limited measurement precision in the sensitivities. As will be discussed in Section 4.2, the measurement error at ω_i may be modeled as a random variable E_i , with standard deviation σ_{E_i} . Thus the normalized standard deviation of E_i is $\sigma_{E_i}/F(\omega_i)$. The limited measurement precision may be incorporated into (4.1) by dividing the sensitivities $\widehat{s}_{p_i}^{F|}$ by $\sigma_{E_i}/F(\omega_i)$. Thus (4.1) develops into

$$\begin{aligned} \frac{\Delta F(\omega_i)}{F(\omega_i)} \frac{1}{\sigma_{E_i}/F(\omega_i)} &= \sum_{j=1}^n \left(\widehat{s}_{p_i}^{F(\omega_i)} \frac{1}{\sigma_{E_i}/F(\omega_i)} \frac{\Delta p_j}{p_j} \right) \\ \Leftrightarrow \frac{\Delta F(\omega_i)}{\sigma_{E_i}} &= \sum_{j=1}^n \left(\frac{p_j}{\sigma_{E_i}} s_{p_i}^{F(\omega_i)} \frac{\Delta p_j}{p_j} \right) \end{aligned}$$

which can be written as

$$\frac{\Delta F(\omega_i)}{\sigma_{E_i}} = \mathbf{s}_i^T \Delta \mathbf{p}, \quad (4.7)$$

where $\mathbf{s}_i \triangleq \left(\frac{p_1}{\sigma_{E_i}} s_{p_1}^{F(\omega_i)}, \dots, \frac{p_n}{\sigma_{E_i}} s_{p_n}^{F(\omega_i)} \right)^T$, and $\Delta \mathbf{p} \triangleq \left(\frac{\Delta p_1}{p_1}, \dots, \frac{\Delta p_n}{p_n} \right)^T$.

The sensitivities according to (4.7) are depicted in Figure 4.1, for the circuit of Figure 3.1. It is assumed that all measurement errors have a standard deviation of 1 mV. In contrast to Figure 3.3, in the case of Figure 4.1 the sensitivities vanish for higher frequencies. This means that at high frequencies $\Delta|F|$ cannot be measured with sufficient accuracy. Furthermore (4.7) also models the situation where the measurement errors are different in (expected) magnitude for different measurements. In this case the standard deviation σ_{E_i} may be different for each measurement.

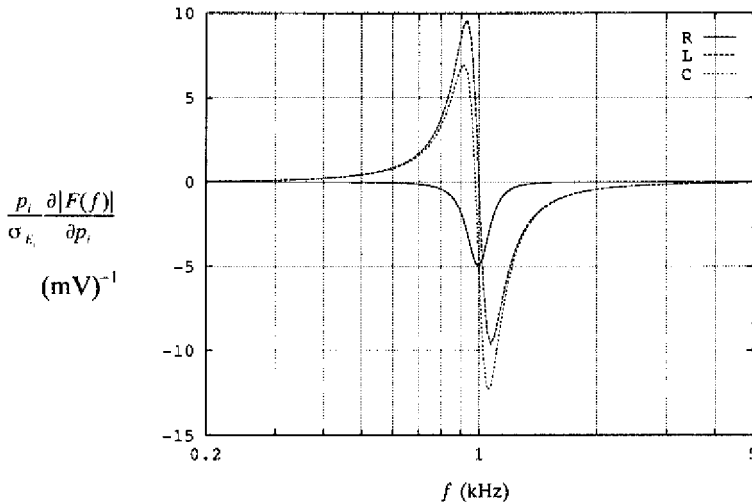


Figure 4.1. Normalized sensitivities for the circuit of Figure 3.1, taking into account the measurement errors.

Assume that the m row vectors $\mathbf{s}_1^T, \dots, \mathbf{s}_m^T$ are obtained by calculating the sensitivities at m different frequencies. Then m relations (4.7) are obtained, which may be stacked to obtain the linear model

$$\mathbf{S}\Delta\mathbf{p} = \Delta\mathbf{F}, \text{ with } \mathbf{S} \triangleq \begin{bmatrix} \mathbf{s}_1^T \\ \vdots \\ \mathbf{s}_m^T \end{bmatrix} \text{ and } \Delta\mathbf{F} \triangleq \begin{bmatrix} \frac{\Delta F(\omega_1)}{\sigma_{E_1}} \\ \vdots \\ \frac{\Delta F(\omega_m)}{\sigma_{E_m}} \end{bmatrix}. \quad (4.8)$$

Here \mathbf{S} is the $m \times n$ complex *sensitivity matrix*. Element (i, j) of \mathbf{S} is the normalized sensitivity $\frac{P_j}{\sigma_{E_i}} s_{p_i}^{F(\omega_i)}$. If the relative parameter deviations $\Delta\mathbf{p}$ are known then (4.8) predicts the circuit behavior at m frequencies. In the rest of this chapter, the right-hand side of (4.8) will be referred to as *output vector*.

The m frequencies are provided by a human designer before the sensitivities are calculated. It is common practice to select a fixed number of frequencies per decade, equally spaced on a logarithmic scale. The frequency range depends upon the frequency range in which circuit behavior must be verified. Consider the second-order lowpass filter of Figure 3.1, for example. According to Figure 3.2, the breakpoint is at 1 kHz. Therefore the frequency range is centered at 1 kHz. The range of the useful measurement frequencies is estimated from sensitivity plots like Figure 4.1. Whether or not a measurement is useful depends upon the expected deviation of the output, relative to both the maximum expected deviation and the magnitude of the measurement errors. The number of frequencies per decade depends upon the rate of change of the circuit sensitivities as a function of frequency. This in turn depends upon the order of the filter: For a higher-order filter the number of sensitivities per decade should be higher.

Until now only frequency behavior has been discussed. However a linear model of the form (4.8) is also applicable in the DC domain, as well as in the time domain [Dai90], [Sten87]. For DC testing of non linear circuits, the circuit behavior is linearized for each value of the circuit input voltage. (This is equally valid in the case of multiple circuit inputs.) Like this, the complete non linear behavior is described at a discrete

number of input stimuli, analogous to the discretization of the frequency range that was described above.

A model of the form (4.8) can also describe the behavior of mixed analog/digital circuits. This will be demonstrated in Section 7.3, for a digital to analog converter that uses a ladder network of resistors. Each switch combination corresponds with a linear circuit, and therefore for each switch combination the DC sensitivities may be calculated. In this case the rows of \mathbf{S} correspond with all possible switch combinations. For an r -bits converter, \mathbf{S} consists of $m = 2^r$ rows.

To use a model of the form (4.8) in the time domain, time-dependent circuit behavior is discretized in time, and each row of \mathbf{S} corresponds with a specific time. The result is a linear system of the form

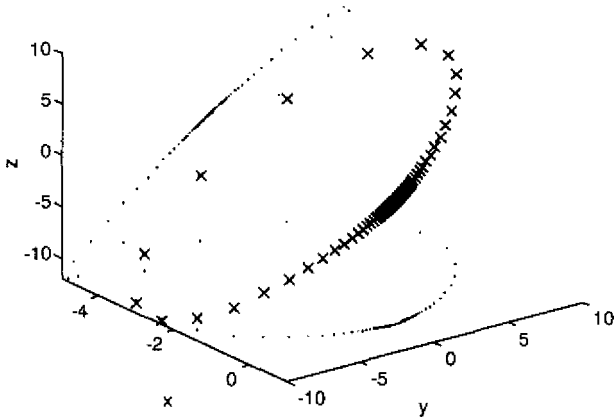
$$\mathbf{S}\Delta\mathbf{p} = \Delta\mathbf{F}, \text{ with } \mathbf{S} = \begin{bmatrix} \frac{P_1}{\sigma_{E_1}} S_{P_1}^{F(t_1)} & \dots & \frac{P_n}{\sigma_{E_n}} S_{P_n}^{F(t_1)} \\ \vdots & & \vdots \\ \frac{P_1}{\sigma_{E_m}} S_{P_1}^{F(t_m)} & \dots & \frac{P_n}{\sigma_{E_n}} S_{P_n}^{F(t_m)} \end{bmatrix} \text{ and } \Delta\mathbf{F} \triangleq \begin{bmatrix} \frac{\Delta F(t_1)}{\sigma_{E_1}} \\ \vdots \\ \frac{\Delta F(t_m)}{\sigma_{E_m}} \end{bmatrix}.$$

Efficient computation of the sensitivities $\frac{P_j}{\sigma_{E_i}} S_{P_j}^{F(t_i)}$ was discussed in Section 3.1.4.

In principle the number of rows of \mathbf{S} may be chosen arbitrarily large, since the test method will cope with possible redundancy in \mathbf{S} . However, if the number of rows is very large then the selection of test points may become time consuming. If test-point selection must be performed on-line¹ then this is a problem and an effort should be made to limit the number of rows of \mathbf{S} .

¹ A situation where on-line test-point selection occurs is with the detection of large parameter deviations by repeated application of a linear model, as will be discussed in Section 4.3.5.

The sensitivities depicted in Figures 3.3 and 4.1 were calculated for a frequency range of 0.2 kHz to 5 kHz. Per decade the sensitivities were evaluated at 100 frequencies, equally spaced on a logarithmic scale, producing a total of 140 sensitivities per component. In Figures 3.3 and 4.1 the sensitivities for each component were joined by line segments to obtain a continuous curve. The segmentation is hardly noticeable because of the large number of frequencies per decade. In Figure 4.1 each of the three graphs corresponds with a *column* of \mathbf{S} . On the other hand, Figure 4.2 plots the sensitivities in a different manner. There each symbol corresponds with a *row* of \mathbf{S} , giving a total of 140 symbols. The discretization is purposely left visible. Figure 4.2 shows that the sensitivities are in a two-dimensional plane. This demonstrates more clearly the correlation between the sensitivities that was already discussed in Section 3.1.3.1. Plots like Figure 4.2 will be used to explain test-point selection.



$$x = \frac{R}{\sigma_{E_i}} \frac{\partial |F|}{\partial R} \text{ (mV)}^{-1} \quad y = \frac{L}{\sigma_{E_i}} \frac{\partial |F|}{\partial L} \text{ (mV)}^{-1} \quad z = \frac{C}{\sigma_{E_i}} \frac{\partial |F|}{\partial C} \text{ (mV)}^{-1}$$

Figure 4.2. 3-dimensional plot of the sensitivities as a function of frequency, for the circuit of Figure 3.1.

4.1.2 Introduction to Test-Point Selection

In the presented approach, test-point selection, response prediction and the determination of a subset of testable circuit parameters are based on the linear model

(4.8). To facilitate the discussion of these topics some definitions are given. An effort has been made to ensure that the defined terms comply as much as possible with the accepted nomenclature in the testing community.

Definition 4.1 An *input stimulus* is defined as a circuit variable (branch voltage or current) that is generated by a voltage source or current source, thus serving as a *circuit input*.

A *circuit output* is defined as a measurable network function.

A *test node* is defined as any circuit node that is accessible for measurements.

Test nodes are accessible through test needles on the bonding pads of an integrated circuit, for example. In integrated circuits it is often not possible to measure the currents through internal circuit branches. Thus network functions that include such currents do not qualify as circuit outputs.

Definition 4.2 Consider a circuit with k input stimuli (x_1, \dots, x_k) and l outputs (y_1, \dots, y_l) .

A *test-point* is defined as a $(k+1)$ -tuple (x_1, \dots, x_k, y_i) , with $i \in [1, \dots, l]$. This should be interpreted as follows: A test-point specifies the measurement of a circuit output while a specific input stimulus is applied. Each test-point corresponds with an integer number: the index of the corresponding row of \mathbf{S} .

Note that, according to Definitions 4.1 and 4.2, a test-point has no one-to-one correspondence with a test node. Different test points may specify measurements at a single test node, but for different frequencies or time points, for example.

Chapters 1 discussed applications of analog testing. As far as the current chapter is concerned, the goal of analog testing is either to verify the circuit behavior or to

determine the values of the testable circuit parameters. Whether or not a circuit parameter is testable depends upon the relative accuracy with which deviations of its value can be determined, as will be explained in Section 4.2.

To accomplish the mentioned goals, an analog test method should select the input stimuli that sensitize the circuit outputs to possible errors in the circuit components. This accomplished by test-point selection. Clearly the number of test points must be kept to a minimum to make the test as cheap as possible. If a method is to be useful in a practical situation then it should take into account the influence of measurement errors, as will be explained in Section 4.2.

As remarked, a test-point has a one-to-one correspondence with a row in \mathbf{S} . Thus, as far as this chapter is concerned, the goal of a test method is to select a subset of the rows of \mathbf{S} . Like this the set of rows of \mathbf{S} may be considered as a set of *candidate test points*. A test method tries to select a minimal and sufficient subset of the set of candidate test points. The selected test points correspond with a row reduced \mathbf{S} , denoted by² $\mathbf{S}_{r\cdot}$. In general it is not possible to detect changes in all the circuit parameters, due to a circuit's limited accessibility for measurements. Therefore a subset of testable parameters is selected, corresponding with a column-reduced \mathbf{S} , denoted by $\mathbf{S}_{\cdot r}$. Like this the selection of test points and testable parameters results in a row and column reduced sensitivity matrix $\mathbf{S}_{r\cdot}$. In the rest of this chapter it is assumed that k test points and r testable parameters are selected, so $\mathbf{S}_{r\cdot}$ is a $k \times r$ matrix.

The number of test points is not necessarily equal to the number of testable parameters. In fact it is often advantageous to select more test points than parameters, so $k > r$. The reason is that the parameters may be solved with higher accuracy from an overdetermined system in the presence of random measurement errors. In this chapter the situation $k < r$ does not occur, since this would not allow the values of all the

² The subscript "r" in $\mathbf{S}_{r\cdot}$ is shorthand for "reduced" and should not be confused with the variable r .

testable parameters to be determined. Furthermore it is assumed that S_{rr} has full column rank, because its columns correspond with testable parameters.

4.2 The Influence of Measurement Errors

Section 4.1.1 showed that a test method should take into account the influence of measurement errors. Such errors unavoidably occur because physical quantities like voltages or currents cannot be measured with infinite accuracy. Measurement errors may coarsely be divided into two categories: systematic errors and random errors.

Systematic errors may be caused by equipment that is not properly calibrated, or an offset voltage that has not been taken into account, for example. In addition, systematic errors need not be constant but may change when the range of a voltage meter is changed, for example. However, systematic errors may be eliminated by a careful design and construction of the test bed and measuring equipment. Therefore it is assumed that no systematic errors are present.

On the other hand, random errors are caused by limited precision of the measurement equipment and random noise. Random errors will always be present, regardless of the quality of the measuring equipment, or the carefulness with which a test bed is constructed. One can only hope to reduce the influence of such errors so that the circuit property of interest may be measured with sufficient accuracy. Often it is necessary to make more measurements than strictly necessary to further reduce the influence of random fluctuations.

If frequency measurements are made, the influence of random measurement errors can be reduced by using lock-in amplifiers. Such a device can be set to measure a circuit variable at a certain frequency f_m . At the input of the lock-in amplifier is a high-order bandpass filter centered at f_m . In addition, the measured circuit variable is averaged in time. Thus, the influence of measurement noise is greatly reduced.

This section takes into account the random measurement errors by representing them as random variables. Suppose that k test points (rows of \mathbf{S}) are selected, corresponding with k measurements to be performed. The measurement errors are assumed to be mutually independent. Thus they are represented³ by uncorrelated random variables E_1, \dots, E_k , where E_i has mean μ_{E_i} and a variance $\sigma_{E_i}^2$. Because there are no systematic errors $\mu_{E_i} = 0$. Furthermore the covariance $\sigma_{E_i E_j} = 0$, for $i \neq j$, because E_i and E_j are uncorrelated. Thus the covariance matrix \mathbf{C}_E (conform (B.3), Appendix B) of E_1, \dots, E_k is

$$\mathbf{C}_E = \text{diag}(1, \dots, 1) = \mathbf{I}. \quad (4.9)$$

where the normalization according to (4.7) is applied. The observed (i.e., actually measured) values of E_1, \dots, E_k are collected in the k -vector $\mathbf{e} = (e_1, \dots, e_k)^T$.

Like the measurement errors, also the observed deviations of the circuit outputs (again normalized according to (4.7)) are represented by a vector of random variables: $\Delta \mathbf{Y}_r = (\Delta Y_1, \dots, \Delta Y_k)^T$. An observation of $\Delta \mathbf{Y}_r$ is denoted by $\Delta \mathbf{y}_r = (\Delta y_1, \dots, \Delta y_k)^T$. Now define

$$\Delta \mathbf{Y}_r \stackrel{\Delta}{=} \mathbf{S}_r \Delta \mathbf{p}_r + \mathbf{E}. \quad (4.10)$$

Because $\mathbf{S}_r \Delta \mathbf{p}_r$ is constant⁴ for a given $\Delta \mathbf{p}_r$, it follows from (4.8) and (4.10) that,

$$\begin{bmatrix} \mu_{\Delta Y_1} \\ \vdots \\ \mu_{\Delta Y_k} \end{bmatrix} = \mathbf{S}_r \Delta \mathbf{p}_r,$$

³ The used notation is defined in Appendix B.

⁴ The nominal parameter values are assumed to be constant, and the input vector is assumed to be deterministic, and not disturbed by fluctuations.

since $\mu_{E_i} = 0$. Furthermore $\sigma_{Y_i}^2 = \sigma_{E_i}^2$, for $i = 1, \dots, k$. It follows that the covariance matrix \mathbf{C}_{Y_i} of Y_1, \dots, Y_k is equal to \mathbf{C}_E :

$$\mathbf{C}_{Y_i} = \mathbf{C}_E = \mathbf{I}. \quad (4.11)$$

From (4.10) it follows that

$$\Delta \mathbf{y}_r = \mathbf{S}_r \Delta \mathbf{p}_r + \mathbf{e}, \quad (4.12)$$

which is the relation between measurement errors, parameter deviations and observed circuit outputs. In the hypothetical case of zero measurement errors, $\Delta \mathbf{y}_r = \mathbf{S}_r \Delta \mathbf{p}_r$.

Assume that $\Delta \mathbf{y}_r$ is obtained by k measurements. Now an estimate for the parameter deviations $\Delta \mathbf{p}_r$ must be obtained. According to Definition 3.4, the least squares estimate $\Delta \hat{\mathbf{p}}_r$ of $\Delta \mathbf{p}_r$ minimizes the residual

$$\|\Delta \mathbf{y}_r - \mathbf{S}_r \Delta \hat{\mathbf{p}}_r\|_2^2. \quad (4.13)$$

According to (C.5), Appendix C, the least squares approximation of $\Delta \mathbf{p}_r$ is obtained as

$$\Delta \hat{\mathbf{p}}_r = (\mathbf{S}_r^* \mathbf{S}_r)^{-1} \mathbf{S}_r^* \Delta \mathbf{y}_r, \quad (4.14)$$

which may be obtained using the pseudo-inverse of \mathbf{S}_r , as explained in Section 3.2.3.

The influence of measurement errors on the estimated parameter deviations is determined on the basis of (4.14). Analogous to the measured output deviations, the estimated parameter values may be modeled by a vector of random variables $\hat{\mathbf{P}} = (\hat{P}_1, \dots, \hat{P}_r)^T$, where r is the number of selected testable parameters. It was explained that the measured output deviations in (4.14) are observations of random variables ΔY_r , with an associated covariance matrix \mathbf{C}_Y given by (4.11). Thus, according to (4.14)

$$\Delta \hat{\mathbf{P}}_r = \mathbf{A} \Delta Y_r, \quad (4.15)$$

with $\mathbf{A} = (\mathbf{S}_r^* \mathbf{S}_r)^{-1} \mathbf{S}_r^T$. Because $\mu_{Y_i} = 0$ for $i = 1, \dots, k$ also $\mu_{\hat{P}_i} = 0$ for $i = 1, \dots, r$.

Furthermore the covariance matrix $C_{\hat{p}_r}$ may be determined from (B.4) as

$$\begin{aligned} C_{\hat{p}_r} &= (\mathbf{S}_r^* \mathbf{S}_r)^{-1} \mathbf{S}_r^* C_y \left[(\mathbf{S}_r^* \mathbf{S}_r)^{-1} \mathbf{S}_r^* \right] \\ &= (\mathbf{S}_r^* \mathbf{S}_r)^{-1} \mathbf{S}_r^* \mathbf{I} \mathbf{S}_r (\mathbf{S}_r^* \mathbf{S}_r)^{-1} \\ &= (\mathbf{S}_r^* \mathbf{S}_r)^{-1} \end{aligned} \quad (4.16)$$

This expression models the influence of measurement errors on the estimated parameter deviations. The magnitude of the i^{th} diagonal element of $C_{\hat{p}_r}$ indicates the precision with which the value of the i^{th} parameter can be estimated: A large variance signifies a low parameter testability. Like this a parameter is considered testable if the variance of its estimated deviation is below a certain limit. The off-diagonal elements of $C_{\hat{p}_r}$ contain the parameter covariances. Note that (4.16) is only valid in the case of a linear model of the form (4.8). Thus, the predictions (4.16) are valid only if the parameter deviations $\Delta \mathbf{p}_r$ are sufficiently small. Chapter 7 shows an example where (4.16) is accurate for parameter deviations smaller than approximately 5 %.

The influence of the measurement errors on the predicted circuit response can be determined in the same manner as shown above for the predicted parameter deviations. After the estimates (4.14) of the parameter deviations are obtained the complete circuit behavior may be predicted as

$$\Delta \hat{\mathbf{y}} = \mathbf{S}_r \Delta \hat{\mathbf{p}}_r \quad (4.17)$$

Here $\hat{\mathbf{y}}$ is an m -vector, containing the response estimates (the predicted output vector) at all m candidate test points. \mathbf{S}_r is an $m \times r$ matrix that contains those columns of \mathbf{S} that correspond with testable parameters, as explained in Section 4.1.2. The predicted output (4.17) may be represented by a vector of random variables $\hat{\mathbf{Y}} = (\hat{Y}_1, \dots, \hat{Y}_m)^T$.

Thus

$$\Delta \hat{\mathbf{Y}} = \mathbf{S}_r \Delta \hat{\mathbf{P}}_r \quad (4.18)$$

Because $\mu_{\hat{p}_i} = 0$ for $i = 1, \dots, r$, also $\mu_{\hat{y}_i} = 0$ for $i = 1, \dots, m$. Note that, like (4.15), (4.18) is a linear map between two vectors of random variables. Thus, using (A.3) and (4.16), the covariance matrix $\mathbf{C}_{\hat{y}}$ may be determined as

$$\begin{aligned} \mathbf{C}_{\hat{y}} &= \mathbf{S}_r \mathbf{C}_{\hat{p}} \mathbf{S}_r^* \\ &= \mathbf{S}_r (\mathbf{S}_r^* \mathbf{S}_r)^{-1} \mathbf{S}_r^* \end{aligned} \quad (4.19)$$

Note that, because of the normalization used in (4.7), the i^{th} row of $\mathbf{C}_{\hat{y}}$ is normalized on $\sigma_{\hat{y}_i}^2$, the measurement variance at the i^{th} candidate test-point. Thus, if the i^{th} diagonal element of $\mathbf{C}_{\hat{y}}$ is 1, the output at the i^{th} candidate test-point is predicted with a precision equal to the measurement precision. Section 4.3 will use the prediction (4.16) to determine the number of selected test points.

As yet no assumptions have been made about the probability distribution of the measurement errors. The only assumptions were that the circuit model is linear, that there are no systematic measurement errors, and that the measurement errors are mutually independent. Next the probability distribution of the measurement errors is taken into account.

The selected test points correspond with measurements of a physical quantity, like a voltage or current. The normal, or Gaussian distribution [Bevi69], [Walp93] is especially suitable to model such measurements. Let Y be a random variable with mean μ_Y and variance σ_Y^2 . If Y is normally distributed then its probability density function is

$$n(Y, \mu_Y, \sigma_Y) = \frac{1}{\sqrt{2\pi}} e^{-1/2[(Y - \mu_Y)/\sigma_Y]^2}. \quad (4.20)$$

Figure 4.3 depicts the standardized normal distribution. It has several well-known properties. In the context of this section the following property (which is not proven here) is of interest. Assume that the measurement errors are normally distributed. Then, according to (4.20), the i^{th} measured circuit output is normally distributed. Now consider the linear mapping (4.15). Because the random variables in ΔY_r are normally

distributed, also the random variables in $\Delta\hat{\mathbf{P}}_r$ are normally distributed. In this sense the probability distribution has a reproductive property. Note that this reproductive property is certainly not shared by all probability distributions. Because the parameter estimates are normally distributed, it is now clear from the linearity of mapping (4.18) that also the predicted outputs are normally distributed random variables.

Furthermore, if the parameters estimates are normally distributed then the least squares approximate is also the maximum likelihood estimate [Walp93]. This means that the likelihood of measuring the deviations Δy_r is at a maximum for the least squares estimate $\Delta\hat{\mathbf{p}}_r$ of (4.14).

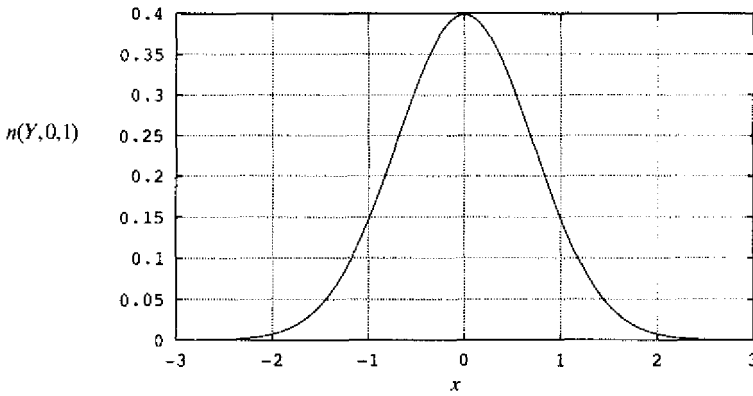


Figure 4.3. The normal probability distribution.

4.3 Direct Test-Point Selection

This section uses the SVD of a linear model of the form (4.8) to select test points. To simplify the discussion, it is assumed that a linear frequency-dependent circuit is tested. However the presented methods are also applicable to nonlinear circuits in the DC and time domain, as explained in Section 4.1.1.

Again the lowpass filter of Figure 3.1 will be used for the purpose of illustration. For that example only measurements of the magnitude of the complex voltage transfer

function are made. The sensitivities in \mathbf{S} are normalized as in (4.8) to take into account the magnitude of the measurement errors. Thus, for this example, the sensitivities are real numbers defined by

$$(\mathbf{S})_{ij} = \frac{p_j}{\sigma_{t_i}} \operatorname{Re} \left\{ \frac{1}{F(\omega_i)} s_{p_j}^{F(\omega_i)} \right\}, \quad (4.21)$$

where $(\mathbf{S})_{ij}$ denotes element (i, j) of \mathbf{S} , F is the complex voltage transfer function, and p_j is the nominal value of the j^{th} circuit parameter. Expression (4.21) is readily obtained from (3.9) by using the sensitivities (4.8). In this example, $\sigma_{t_i} = 1 \text{ mV}$ for all candidate test points. Because the input voltage has a fixed amplitude (1 V, according to Figure 3.1) the selected measurements of $|F|$ directly correspond with measurements of $|V_o|$, normalized on the measurement errors.

A test method uses test-point selection to select a subset of the m rows of \mathbf{S} , corresponding with a set of test points. A direct method for test-point selection is presented in Section 4.3.2. First, the next section discusses a form of column reduction of \mathbf{S} . This is a necessary preliminary step because in the general case \mathbf{S} does not have full column rank.

4.3.1 Dealing with Untestable Parameters

This section uses the SVD of the sensitivity matrix \mathbf{S} to cope with parameters that have a very low testability, or are completely untestable. A parameter is considered untestable if, for example, its influence on the circuit behavior is so small that it cannot be distinguished from the influence of the unavoidable random errors that disturb the measurements. It is also imaginable that, in the presence of measurement errors, it is not possible to detect the individual influence of the members of a group of two or more parameters. Such a group is called an *ambiguity group* [Sten89]. The usual approach is to consider only one parameter in an ambiguity group testable, rendering the other parameters in the group untestable.

A common solution is to fix parameters with a low testability at their nominal values [Dai90], [Hemi90], [Sten87]. Correspondingly, the best estimate of the deviation of untestable parameters is zero. The underlying assumption is that this estimate is more precise than a deviation that is calculated from measurements. This effectively drops the columns that correspond with untestable parameters from the linear model (4.8). Like this the number of degrees of freedom of this model is reduced by excluding parameters that have a too low influence on the circuit behavior.

A column reduction of \mathbf{S} is commonly performed with a routine that resembles a routine for test-point selection [Dai90], [Hemi90], [Sten87]. Such an approach utilizes a criterion that judges the benefit of including a certain column of \mathbf{S} . Alternatively, this section shows how to use the SVD to accomplish a kind of column reduction that is based upon the *optimal* approximation of matrices discussed in Section 3.2.3.

Consider the SVD of the $m \times n$ sensitivity matrix \mathbf{S} , given by

$$\mathbf{S} = \mathbf{U} \mathbf{W} \mathbf{V}' , \quad (4.22)$$

where \mathbf{U} , \mathbf{W} and \mathbf{V} are defined in Theorem 3.2. The SVD of \mathbf{S} may of course be written as

$$\mathbf{S}' \triangleq \mathbf{S} \mathbf{V} = \mathbf{U} \mathbf{W} , \quad (4.23)$$

because \mathbf{V} is unitary. The i^{th} column of \mathbf{V} contains the coordinates of the i^{th} eigenvector of $\mathbf{S}' \mathbf{S}$ with respect to the unit base of \mathbf{R}^n . According to (4.23), \mathbf{S} is transformed into another $m \times n$ sensitivity matrix \mathbf{S}' by right multiplication with the unitary \mathbf{V} . By substituting into (4.22) the definition of \mathbf{S}' it follows that $\mathbf{S} = \mathbf{S}' \mathbf{V}'$. Substitution of this last expression into the linear system (4.8) results in

$$\mathbf{S}' \Delta \mathbf{p}' = \Delta \mathbf{F}, \quad \text{with } \mathbf{p}' \triangleq \mathbf{V}' \mathbf{p} . \quad (4.24)$$

Because \mathbf{V} has full rank, (4.24) is completely analogous to the original system (4.8). It describes the circuit outputs in terms of a transformed sensitivity matrix \mathbf{S}' and

deviations $\Delta \mathbf{p}'$ of new parameters $\mathbf{p}' = (p_1', \dots, p_n')^T$. Note that there is a one-to-one correspondence between the rows of \mathbf{S} and the rows of \mathbf{S}' , so test-point selection can be accomplished by selecting a subset of the rows of \mathbf{S}' . This will be discussed in a later stage.

Note that the transformation (4.8) \rightarrow (4.24) is analogous to the transformation that was used to obtain (3.30) in the data set example presented in Section 3.2.1. There the transformation corresponded with a rotation and mirroring of the data set of Figure 3.4, resulting in the data set of Figure 3.6. Similarly, the transformation with \mathbf{V} that obtains (4.24) corresponds with a rotation and mirroring in three dimensions of the circuit sensitivities depicted in Figure 4.2. The result is depicted in Figure 4.4. In this example, \mathbf{S} and \mathbf{S}' both have three columns, corresponding with three parameters. The coordinate axes of Figure 4.4 correspond with the first two new parameters, p_1' and p_2' .

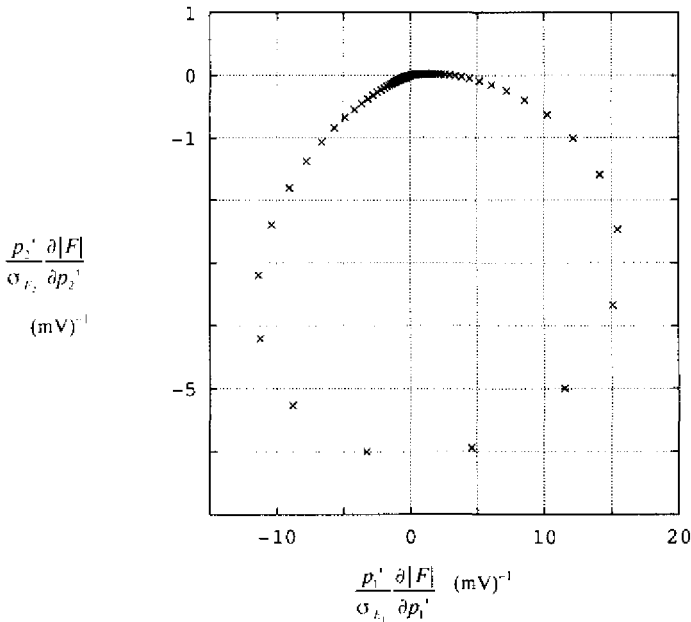


Figure 4.4. Sensitivities to p_1' and p_2' as a function of frequency, for the circuit of Figure 3.1.

In contrast to the original circuit parameters \mathbf{p} , the new parameters \mathbf{p}' are uncorrelated. This follows from the covariance matrix:

$$\begin{aligned} \mathbf{C}_{p'} &= [(\mathbf{S}')' \mathbf{S}']^{-1} \\ &= [(\mathbf{U}\mathbf{W})' \mathbf{U}\mathbf{W}]^{-1} \\ &= \mathbf{W}^{-2} \\ &= \text{diag}(w_1^{-2}, \dots, w_n^{-2}) \end{aligned} \quad (4.25)$$

which has zero off-diagonal elements, and thus the covariances of the new parameters are zero. Furthermore, the diagonal elements of (4.25) are sorted to increasing magnitude because it are the inverse squares of the singular values⁵ of \mathbf{S} . Thus, for the example of Figure 3.1, p_1' has the highest testability, p_2' has second highest testability, and p_3' has lowest testability. This shows that Figure 4.4 depicts the sensitivities to the two new parameters with the highest testability. For the example of Figure 3.1, the three singular values of \mathbf{S} are calculated as $w_1 = 92.33 \cdot 10^3$, $w_2 = 27.56 \cdot 10^3$, and $w_3 = 0$ within the precision with which the singular values are calculated. Thus $\text{rank}(\mathbf{S}) = 2$, corresponding with two testable parameters. This is not surprising because the sensitivities in Figure 4.2 are in a two-dimensional plane. Figure 4.4 depicts just this two-dimensional cross section from \mathbf{R}^3 .

According to Theorem 3.3, the best⁶ rank k approximation of \mathbf{S} is given by

$$\mathbf{S}_k = \sum_{i=1}^k w_i \mathbf{u}_i \mathbf{v}_i', \quad (4.26)$$

where \mathbf{S}_k is an $m \times n$ matrix. For the transformed sensitivity matrix (4.23) the best rank k approximation is

$$\mathbf{S}'_k = \mathbf{U} \mathbf{W}_k, \text{ with } \mathbf{W}_k = \left[\begin{array}{c|c} \mathbf{W}_\pi & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right] \text{ and } \mathbf{W}_\pi \triangleq \text{diag}(w_1, \dots, w_k), \quad (4.27)$$

⁵ The singular values of \mathbf{S} are sorted to decreasing magnitude, according to Theorem 3.4.

⁶ With regard to 2-norm, as explained in Theorem 3.5.

where both \mathbf{S}'_k and \mathbf{W}_k are $m \times n$ matrices. Now define the $m \times k$ matrix

$$\mathbf{S}'_{\cdot r} \triangleq \mathbf{U}_{\cdot r} \mathbf{W}_r, \quad (4.28)$$

where $\mathbf{U}_{\cdot r}$ is an $m \times k$ matrix containing the first k columns of \mathbf{U} . Then it follows that, if $\text{rank}(\mathbf{S}) = k$, the output deviations are given by

$$\mathbf{S}'_{\cdot r} \Delta \mathbf{p}'_r = \Delta \mathbf{F}, \quad (4.29)$$

where $\Delta \mathbf{p}'_r \triangleq (\Delta p'_1, \dots, \Delta p'_k)^T$. Note that the linear system (4.29) is directly obtained from the SVD of \mathbf{S} , as defined by Theorem 3.2. The matrix $\mathbf{S}'_{\cdot r}$ is obtained as (4.28), and $\Delta \mathbf{p}'_r$ contains the first k new variables. If $\text{rank}(\mathbf{S}) > k$ (4.29) gives the output deviations for the best rank k approximation of \mathbf{S} . Therefore it has now been shown that the mentioned column reduction may be obtained directly from the SVD of \mathbf{S} , without further computational effort. Note that a rank k approximation of \mathbf{S} that is obtained by selecting a subset of the columns of \mathbf{S} is never better than the approximation (4.26). Therefore, (4.29) is the most accurate rank k description of the output deviations, in the general case of a matrix \mathbf{S} with $\text{rank}(\mathbf{S}) \leq n$.

The remaining issue is how to determine k , the number of testable parameters. For higher k , (4.29) more accurately describes circuit behavior, because \mathbf{S}'_k more precisely approximates \mathbf{S}_k . However the influence of random measurement errors on the parameters increases, because the last parameter(s) will have low testability. In this respect, a trade-off will have to be made between the precision of the linear model and the influence of random measurement errors. Section 4.3.3 further discusses this issue.

For the example of Figure 3.1, $\text{rank}(\mathbf{S}) = 2$, so $\mathbf{S} = \mathbf{S}_2$, and the deviation of the outputs are exactly described with the column reduced system⁷

⁷ Here $\Delta \mathbf{y}$ denotes the exact output deviations, not taking into account the measurement errors.

$$\mathbf{S}'_2 \begin{bmatrix} p_1' \\ p_2' \end{bmatrix} = \Delta \mathbf{y}.$$

As discussed above, this means that Figure 4.4 depicts all the sensitivity information for the circuit of Figure 3.1 in just two dimensions.

4.3.2 A Direct Method for Test-Point Selection

Consider *functional testing*, where one is only interested in verifying whether or not a circuit's behavior is within agreed bounds. It is not necessary to determine the testability of the circuit parameters \mathbf{p} , or their deviations $\Delta \mathbf{p}$, as long as an accurate prediction of the complete circuit behavior is obtained from the selected measurements. Of course the *effect* of non testable parameters must still be incorporated. This may be accomplished by using (4.29) to relate the circuit outputs to k new parameters $\Delta \mathbf{p}'$, in the case of a rank k approximation to the sensitivity matrix. For functional testing it is not important what subset of the original circuit parameters contains testable parameters. Only the observed circuit behavior is of importance.

Test points are selected from the set of candidate test points that correspond with the rows of \mathbf{S}' . For the moment, the goal of test-point selection is to select a minimum number of measurements with which the functionality of the circuit may be verified. A few observations follow directly from the discussed plots of the circuit sensitivities.

First it seems logical to measure a circuit output at a frequency where a possible deviation (caused by non nominal circuit components) is distinctly noticeable. In other words: A measurement should be made where the sensitivity of a circuit output to component deviations is high. In that case it is more likely that the output deviation can be distinguished from fluctuations caused by random measurement errors.

Second, the measurements should be as independent as possible. Loosely speaking, this minimizes the number of measurements by selecting one measurement per

independently measurable circuit property. Next a straightforward method for test-point selection is derived on the basis of these arguments.

Consider the sensitivities of Figure 4.4. For clarity this plot is redrawn in Figure 4.5, where both coordinate axes are now drawn to the same scale. A test-point is denoted by a symbol 'o'. It was shown in the previous section that p_1' has the smallest variance, so $\Delta p_1'$ may be determined with the highest precision. This is confirmed by Figure 4.5, where the main axis of the sensitivity ellipsoid is almost parallel to the coordinate axis that corresponds with the sensitivity to p_1' . Similarly, the sensitivity to the (uncorrelated) parameter p_2' is second highest.

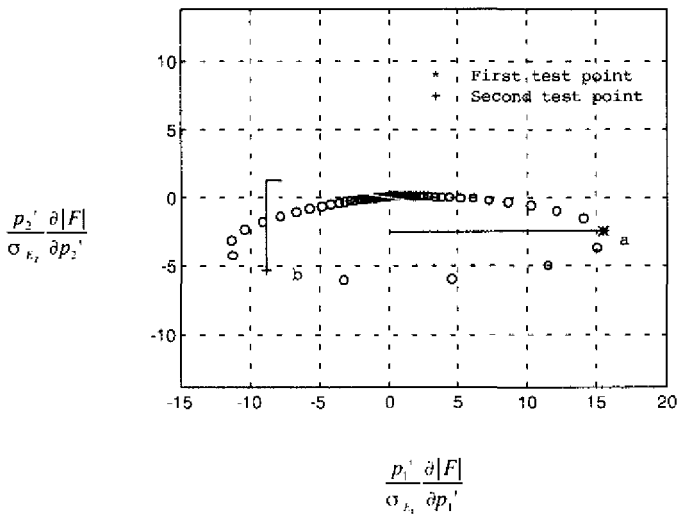


Figure 4.5. Direct test-point selection for the circuit of Figure 3.1.

Test-point selection corresponds with selecting dots in Figure 4.5. The first test-point is selected where the sensitivity to p_1' is at a maximum. Thus the first test-point is selected to measure p_1' with the highest possible precision. In principle, this procedure is repeated for the second test-point. However there is an additional criterion that influences the selection of the second test-point: it should be independent from the first test-point.

This is accomplished as follows. The sensitivities may be seen as endpoints of vectors that start from the origin. The direction of the vector associated with the first test-point is indicated by a grayed line **a** in Figure 4.5. For the second test-point only the components of the sensitivity vectors are considered that are orthogonal to line **a**. For the second selected test-point this orthogonal distance is indicated by a grayed line **b**. This particular test-point is selected because line **b** has the largest component in the direction of p_2' (indicated by the vertical black line). Thus the second test-point is selected to measure deviations in p_2' , while ensuring its independence from the first test-point.

The algorithmic counterpart of the discussed graphical method is derived easily. The first test-point is selected where the sensitivity to p_1' is at a maximum. Thus the first test-point is the row index of the largest element in the first column of \mathbf{S}' . Next the remaining $m - 1$ rows of \mathbf{S}' are orthogonalized to the selected row to remove redundant information. The second test-point is selected as the row index of the largest element in the second column of \mathbf{S}' . Then the remaining $m - 2$ rows of \mathbf{S}' are orthogonalized to the second selected row. The process continues until k test points are selected, where k is the number of testable parameters. The numerical stability of this method is sufficient because the number of test points that is selected is in general much smaller than n .

The result of the discussed method is a $k \times k$ matrix \mathbf{S}'_{rr} . The parameter deviations are estimated as

$$\Delta \hat{\mathbf{p}}'_r = (\mathbf{S}'_{rr})^{-1} \mathbf{S}'_{rn} \Delta \mathbf{y}_r, \quad (4.30)$$

in accordance with (4.14). The SVD of \mathbf{S}'_{rr} may be used to compute (4.30), according to (3.43). After the parameter deviations are estimated, the predicted circuit output is obtained as

$$\Delta \hat{y} = \mathbf{S}'_{r,} \Delta \hat{\mathbf{p}}'_{r,}, \quad (4.31)$$

in accordance with (4.17).

For the example of Figure 3.1, $k = \text{rank}(\mathbf{S}') = 2$ and therefore two test points were selected. They correspond with measurements of the magnitude of the voltage transfer function at frequencies of 957 Hz and 1.07 kHz. To check the quality of the selected points, the complete circuit response was predicted from the two corresponding measurements. A deviating circuit was simulated by randomly setting the component deviations at $\Delta L = -1.04\%$, $\Delta R = -4.40\%$ and $\Delta C = -1.76\%$. The behavior of the deviating circuit was then obtained with a circuit simulator. Measurements of the output voltage were simulated by adding to the output voltage a Gaussian distributed random variable with a standard deviation of 1 mV.

The prediction (4.31) is depicted in Figure 4.6, together with the error in the prediction. Each continuous curve was obtained by joining the 140 calculated points with line segments.

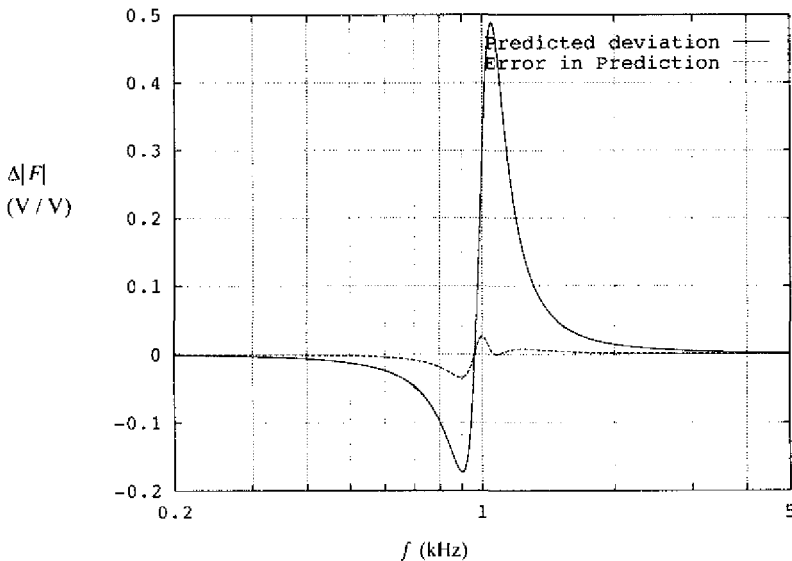


Figure 4.6. Predicted response deviation for the circuit of Figure 3.1.

It appears that the deviating output was predicted with good accuracy. At all frequencies the error is small, relative to the predicted deviation. Also for other examples (presented in Chapter 7) the discussed direct method provides accurate predictions. Naturally, such examples do *not* prove that no better test points are possible. To be able to discuss the “quality” of a set of test points some criterion is necessary. This issue is further discussed in Chapter 5. It has been confirmed that, according to the D -optimality criterion that is discussed there, the two test-points of Figure 4.5 are optimal.

4.3.3 Determination of the Number of Test Points

The direct method that was discussed in the previous section selects test-point i to measure the deviation $\Delta p_i'$. Thus it selects one test-point per testable parameter. According to (4.25) the testability of p_i' decreases with increasing i . Associate a random variable \hat{P}_i' with each estimate \hat{p}_i' . Then a measure for the testability of p_i' is the covariance

$$\sigma_{\hat{p}_i'}^2 = \left((\mathbf{S}'_{\pi} \mathbf{S}'_{\pi})^{-1} \right)_{ii}, \quad (4.32)$$

where \mathbf{S}'_{π} is a $k \times k$ matrix, assuming that k test points are selected. Equation (4.32) is directly obtained from (4.16) when \mathbf{S}_{π} is substituted by \mathbf{S}'_{π} . Because the relation between the deviations $\Delta \mathbf{p}'$, and the circuit output deviations is linear, the estimated parameter deviations $\Delta \hat{\mathbf{p}}' = (\Delta \hat{p}_1', \dots, \Delta \hat{p}_k')^T$ are normally distributed.

The manufacturing process that is used to produce discrete parameters or integrated circuits will show certain fluctuations. This is the result of imperfect control of the process parameters, and is unavoidable. The process fluctuations cause deviations of the circuit parameters. These deviations are modeled as normally distributed errors that are added to the circuit parameters \mathbf{p} . Thus the circuit parameters are modeled as a vector \mathbf{P} of random variables, with $\mu_{p_i} = p_i$ and a covariance matrix \mathbf{C}_p . Because

$\mathbf{p}' \triangleq \mathbf{V}' \mathbf{p}$ is a linear relation, (B.4) may be used to calculate the variance of the new

parameters. Associate a vector P' of random variables with the parameters p' . Then the covariance matrix of the new parameters is

$$C_{P'} = V' C_P V, \quad (4.33)$$

and $\sigma_{P'_i}^2 = (C_{P'})_{ii}$.

It seems reasonable to consider p_i untestable if

$$\frac{\sigma_{\hat{p}'_i}^2}{\sigma_{P'_i}^2} > \eta, \text{ with } \eta \approx 1. \quad (4.34)$$

For $\eta = 1$, (4.34) means that the precision with which the testable parameters are estimated is at least equal to the expected precision, keeping into account the deviations of the manufacturing precision.

It is not always possible to estimate the variances and covariances of all circuit parameters from available information about the production process. However, suppose that the maximum variance that is caused by the production process is estimated as σ_{man}^2 for all circuit parameters. In that case, because V is unitary, $\sigma_{P'_i}^2 = \sigma_{\text{man}}^2$, for all new parameters. In this case, a parameter is untestable if

$$\frac{\sigma_{\hat{p}'_i}^2}{\sigma_{\text{man}}^2} > \eta, \text{ with } \eta \approx 1. \quad (4.35)$$

Criterion (4.35) is used to determine the number of test points. Suppose that $k + 1$ test points are selected. Then the variances (4.32) are calculated. Note that p'_{k+1} has the

lowest testability. Therefore, if $\frac{\sigma_{\hat{p}'_{k+1}}^2}{\sigma_{\text{man}}^2} > \eta$, then the number of test points is determined

as k and the test-point selection is completed. The effectiveness of criterion (4.34) will be demonstrated with examples in Chapter 7.

4.3.4 Selecting a Subset of Testable Parameters

In some applications it is necessary to determine the testability of the parameters \mathbf{p} , as discussed in Section 3.1.2. For example, it may be necessary to evaluate the influence of the parameters on the circuit behavior to obtain a robust circuit design. In this context it is useful to determine the testability of a parameter, because it is directly related to its influence on the circuit behavior. This is the goal of a *testability analysis*.

A subset of the testable circuit parameters may be found with the direct method for test-point selection of Section 4.3.2. Since each parameter corresponds with a column of \mathbf{S} , selecting a subset of parameters corresponds with selecting a subset of the columns of \mathbf{S} . This may be accomplished by using the direct test-point selection method to select a subset of the rows of \mathbf{S}^T . Note that the SVD of the $n \times m$ matrix \mathbf{S}^T consists of the same matrices as the SVD of \mathbf{S} :

$$\begin{aligned}\mathbf{S}^T &= (\mathbf{U}\mathbf{W}\mathbf{V}^*)^T \\ &= (\mathbf{V}^*)^T \mathbf{W}\mathbf{U}^T \\ \Leftrightarrow \mathbf{S}^T \mathbf{U} &= \bar{\mathbf{V}}\mathbf{W}\end{aligned}\quad (4.36)$$

Here $\bar{\mathbf{V}}$ denotes the matrix that is obtained from \mathbf{V} by replacing each element of that matrix by its complex conjugate. Thus, analogous to (4.23), a new $n \times m$ matrix

$$\mathbf{S}'' = \bar{\mathbf{V}}\mathbf{W} \quad (4.37)$$

is defined. A subset of testable parameters may be determined by performing test-point selection on \mathbf{S}'' . According to (4.37) this matrix is immediately obtained from the SVD of \mathbf{S} . Therefore the computational effort of parameter subset selection is small. After the test points and parameters have been determined, the parameter variances (4.16) may be calculated.

A testability analysis was performed for the circuit of Figure 3.1. The measurement errors have a standard deviation of 1 mV. Parameters C and L are selected as testable parameters. Their standard deviations are computed as $7.06 \cdot 10^{-3} \%$ and $7.12 \cdot 10^{-3} \%$,

respectively. It follows that the influence of these parameters on the circuit behavior is large. The high precision with which the parameters are determined is caused by the relatively small measurement errors.

4.3.5 Dealing with Large Parameter Deviations

This section discusses test-point selection and response prediction when the deviations of the circuit parameters are large. In that case the linear model (4.8) no longer accurately describes the (non linear) influence of the parameter deviations on the circuit output. However it is possible to repeatedly apply test-point selection to obtain an accurate response prediction. This may be implemented as follows.

First the linear model (4.8) is calculated by computing the circuit sensitivities, on the basis of the nominal parameter values. Then test-point selection is performed and a group of testable circuit parameters is identified. After the measurements are made the estimates (4.14) are used to update the values of the circuit parameters. The resulting values are a first order estimate of their actual values. On the basis of this first order estimate an updated linear model is calculated, and test-point selection is again performed. This procedure may be repeated several times. It stops when the parameter deviations do not change in the next iteration.

This approach is used in the literature with good results [Dai90]. Still this technique offers some problems. For example, it is not clear what are the convergence properties. Furthermore the efficiency is not very high. For example, there exist techniques that can efficiently update the sensitivities [Bray80] after new measurements have been made. Thus they do not have to be calculated from scratch in each iteration. Finally, it is problematic that a complete set of measurements has to be made in each iteration. Thus the total number of measurements can grow quite large. A possible alternative approach that uses the SVD is suggested in Section 7.4.

4.4 Discussion

This chapter discussed the application of the SVD to analog circuits testing, on the basis of a linear circuit model, defined by a sensitivity matrix \mathbf{S} . Section 4.1.1 showed that the expected magnitude of the measurement errors must be taken into account to obtain a useful linear model. The resulting model uses a particular normalized form of differential sensitivities. Section 4.2 derived practical formulas that estimate the influence of random measurement errors on the precision of the predicted circuit response and parameter deviations.

The examples in Chapter 7 will show that these formulas are accurate when the parameter deviations are less than approximately 5 %. Section 4.3.1 used the SVD of \mathbf{S} to find a set of orthogonal parameters that are optimally suited to predict the circuit response for functional testing. This approach quite naturally lead to a new column reduced linear model of full rank. The obtained parameters are optimal, as was shown in Chapter 3. Section 4.3.2 presented a direct method of test-point selection that selects test points to determine one by one the deviations of the new, optimal parameters. A criterion was introduced in Section 4.3.3 that determines the necessary number of test points, based on the magnitude of the measurement errors. It appeared that, the number of test points is determined by the testability of the last added parameter, because the test points are selected one by one to determine decreasingly measurable parameters.

Summarizing, the SVD is a versatile tool, with applications in several aspects of analog testing. It is used to obtain an optimal, new circuit model of full rank. Column reduction of the old model is accomplished in a simple and optimal manner. The SVD also leads to an effective, direct method of test-point selection. Finally, the SVD may be used to find the least-squares solution of the system that results from test-point selection and column reduction.

The result is a powerful and complete test method, that tackles the various problems encountered in analog testing, particularly the low testability of some circuit parameters and the influence of measurement errors. Furthermore a novel criterion was given that determines the optimal/necessary number of measurements, taking into account the magnitude of the measurement errors.

5

Iterative Test-Point Selection with the SVD

The direct method of Chapter 4 provides accurate response predictions for many examples. However there are still some remaining issues. First, it remains unclear if the obtained test points are optimal. To be able to determine this, Section 5.1 introduces a criterion that can judge the quality of a particular set of test points. Then Section 5.2 presents an iterative method for test-point selection that can obtain better test points than the straightforward direct method of Section 4.3.2. Note that the latter method always chooses the number of test points equal to the number of testable parameters. However, in the presence of relatively large measurement errors it may be advantageous to select more measurements than testable parameters. This compensates for the imprecise measurements by creating an overdetermined system from which the parameter deviations may be solved. This is discussed in Section 5.3, which shows how the iterative method may be used to optimize the number of test points as well as the number of testable parameters.

5.1 The *D*-Optimality Criterion

The selection of test points from a set of candidate test points is closely related to the *optimal design of experiments* [Fedo72]. In this field the goal is to determine the parameters of a linear or non linear system with a minimal set of experiments. In the

case of test-point selection an *experiment* corresponds to a test-point, specifying one measurement. Each test-point may be denoted by an integer number: the index of the corresponding row of the sensitivity matrix. A *design* is a set of k test points, denoted by a set of integers. An often-used criterion that judges the quality of a design is the D -optimality criterion [Box71], [Mitt74], [Fedo72]. This criterion will be used for iterative test-point selection.

Definition 5.1 Consider a set Ξ_j of k test points. Then Ξ_j corresponds with a $k \times r$ matrix \mathbf{S}'_{π} , where $r = \text{rank}(\mathbf{S}'_{\pi})$. Ξ_j is D -optimal if it maximizes the determinant¹ $D \triangleq |(\mathbf{S}'_{\pi})^* \mathbf{S}'_{\pi}|$ over all sets of k test points.

Because

$$D = |(\mathbf{S}'_{\pi})^* \mathbf{S}'_{\pi}| = |(\mathbf{S}'_{\pi})^*| \cdot |\mathbf{S}'_{\pi}| = |\mathbf{S}'_{\pi}|^2 \quad (5.1)$$

a D -optimal design maximizes $|\mathbf{S}'_{\pi}|$. Geometrically, this determinant is the volume of the hyper-parallelepiped defined by the columns of \mathbf{S}'_{π} . Maximization of this volume creates a trade-off between the norm of each column, and the columns' mutual independence. This is similar (but not equal) to the kind of trade-off discussed in Section 4.3.2 (which introduced the direct method for test-point selection).

Let the SVD of the $k \times r$ matrix \mathbf{S}'_{π} be given by

$$\mathbf{S}'_{\pi} = \mathbf{U}_{\pi} \mathbf{W}_{\pi} \mathbf{V}_{\pi}^*, \quad (5.2)$$

where \mathbf{U}_{π} is a $k \times k$ matrix, \mathbf{W}_{π} is a $k \times r$ extended diagonal matrix, and \mathbf{V}_{π} is an $r \times r$ matrix. Substitution of (5.2) into (5.1) acquires

$$D = |\mathbf{W}_{\pi}^2| = |\mathbf{W}_{\pi}| \|\mathbf{W}_{\pi}\|. \quad (5.3)$$

Thus a D -optimal design maximizes

¹ The determinant of a matrix \mathbf{A} is denoted by $|\mathbf{A}|$.

$$|\mathbf{W}_r| = \prod_{i=1}^r w_i, \quad (5.4)$$

the product of the singular values of \mathbf{S}'_r .

According to (4.16) the covariance matrix of the testable parameters is given by

$$\mathbf{C}_{\hat{\rho}_r} = \mathbf{V}_r \mathbf{W}_r^{-2} \mathbf{V}_r^*, \quad (5.5)$$

and therefore

$$|\mathbf{C}_{\hat{\rho}_r}| = |\mathbf{W}_r^{-2}|. \quad (5.6)$$

Comparison of (5.6) with (5.4) shows that the D -optimality criterion minimizes the determinant of the parameter covariance matrix. Thus, by (4.19), it also minimizes the variance of the response prediction.

For example, Figure 5.1 displays $|\mathbf{W}_r|$ for all possible sets of two test points for the circuit of Figure 3.1. Actually it provides a zoomed view, showing in the z direction $|\mathbf{W}_r|$ for all combinations of two test points in the range [50, 90] (outside this range $|\mathbf{W}_r|$ is negligibly small). The height of the surface is also indicated with contour lines that are plotted in the plane below the surface. The smoothness of the surface is caused by the fact that the parameter sensitivities vary only slightly from one frequency to the next. The resulting surface topology is used by the test-point selection algorithm that is presented later.

For each set of two identical test points the two corresponding rows of \mathbf{S}'_r are equal, causing $|\mathbf{W}_r| = 0$. This is visible in the plot: the height of the surface in the plane $x = y$ is zero. Furthermore the determinant of a two-row matrix is identical to the determinant of that matrix with its rows swapped. Therefore the sets of test points (i, j) and (j, i) have the same value of $|\mathbf{W}_r|$. This causes the surface to be symmetrical around the line

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 70 \\ 70 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

since 70 is the midpoint of the complete range $[0, \dots, 139]$ of candidate test points. The test points selected by the direct method of Section 4.3.2 are labeled in Figure 5.1, with the accompanying value for $|\mathbf{W}_\pi|$. The obtained set is D -optimal² because it maximizes $|\mathbf{W}_\pi|$. For this example the direct method obtained a D -optimal design because of the obvious simplicity of the displayed surface. In fact, not taking into account the mentioned symmetry, a D -optimal design is found by locating the surface's single maximum.

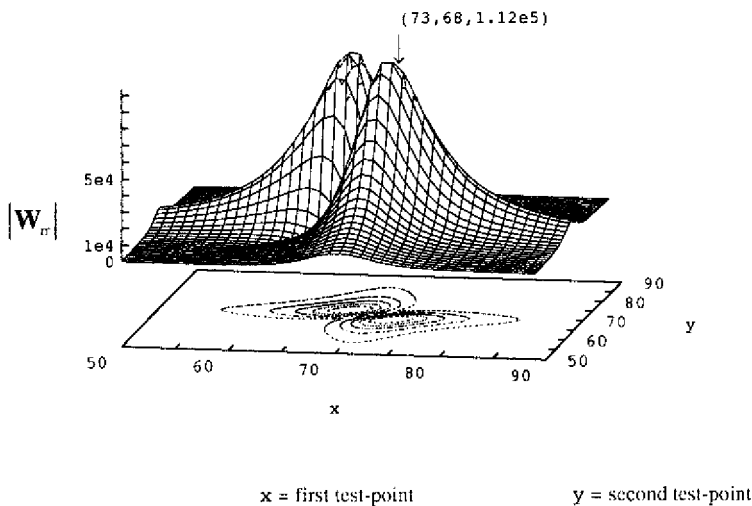


Figure 5.1. $|\mathbf{W}_\pi|$ for the interesting combinations of test points, for the circuit of Figure 3.1.

In the general case it is less straightforward to obtain a D -optimal set of test points. For example, there may be many local maxima in $|\mathbf{W}_\pi|$ that give a sub-optimal design, resulting in a response prediction that is less precise than possible. In this case the

² This is not entirely clear from Figure 5.1, due to slight inaccuracies in the plotting process.

direct method of Section 4.3.2 does not give optimal results, due to its inherent greediness. This may be appreciated by considering direct test-point selection with the help of Figure 5.1. The first test-point defines the plane $x = 73$. Because this test-point is never again changed, in the best case the direct method finds the highest value of $|\mathbf{W}_\pi|$ in the plane $x = 73$. This is not necessarily the global maximum of $|\mathbf{W}_\pi|$. Continuing this argument, the first and second test-point mutually define a line. Suppose that the direct method would have to select a third test-point. It follows that at best it obtains the maximal value of $|\mathbf{W}_\pi|$ on the mentioned line. Again, this is not necessarily the global maximum. This illustrates that, for more complicated surfaces than Figure 5.1, a direct method may easily obtain a less than optimal set of test points. The next section therefore discusses an iterative approach that may obtain a better set of test points in the general case of a more intricate function $|\mathbf{W}_\pi|$.

5.2 An Iterative Approach

The previous section explained that a method for test-point selection should try to find the global maximum of $|\mathbf{W}_\pi|$, in order to obtain a D -optimal design. This is in fact a *combinatorial optimization* problem, where the minimum or maximum value of a cost function of potentially many independent variables must be found. The cost function gives a quantitative measure of the goodness, or quality of a certain choice of the variables. The optimization process changes the variables in a way that optimizes the cost function. In this context the test-point selection problem may be formulated as follows.

The m rows of \mathbf{S}' have indices $0, \dots, m-1$. Consider a set Ξ_i of k test points. Then each element of Ξ_i is an integer number in the range $0, \dots, m-1$. The D -value (according to Definition 5.1) that corresponds with Ξ_i is denoted by D_i . If a row index of \mathbf{S}' appears more than once in Ξ_i , then $D_i = 0$. This renders the set Ξ_i useless, since D should be maximized.

Therefore the members in Ξ_i are chosen to be mutually different. Furthermore, as has been pointed out in the previous section, if Ξ_i consists of a permutation of the test points in Ξ_j then $D_i = D_j$. In that case Ξ_i and Ξ_j are considered to be identical. Consequently there are $\binom{m}{k}$ different sets of test points, constituting the complete *search space*. The maximum value (hence optimal) of D over the search space is denoted by D_{opt} .

An iterative approach often starts with a randomly selected set Ξ_0 . In iteration i , Ξ_{i-1} is changed into Ξ_i . At the end of iteration i a set Ξ_i is selected for which $D = D_i$. Perhaps several modifications of Ξ_{i-1} were tried in iteration i . The highest value of D that has been obtained in any stage of the iterative process is denoted by D_{max} . It is obtained for the set Ξ_{max} . Ultimately the iterative process is aborted when it is not possible to find a set for which $D > D_{\text{max}}$. Note that D_{max} is not necessarily equal to D_{opt} because the method may get stuck in a local maximum. There are several combinatorial optimization methods known in the literature that adhere to this general scheme. They differ mainly in the way Ξ_{i-1} is changed in iteration i , and in how the change in D_i is incorporated.

One possible approach tries a large number l of randomly composed sets and then selects the set with highest D . The problem with this method is that it is not clear how many sets should be tried to ensure that D_{max} is close to D_{opt} . If l is too small then the global maximum is not reached and it is very likely that D_{max} is less than D_{opt} . Choosing l very large is more likely to give a good solution but results in a large computational overhead.

A more effective approach is iterative refinement [Kirk83]. In iteration i several modifications of Ξ_{i-1} are tried until a new set Ξ_i is found with $D_i > D_{i-1}$. Then the procedure proceeds with the next iteration. The advantage of this technique is that, in contrast to the random method, the local behavior of D is taken into account. A

disadvantage of this approach is its potential greedy behavior, causing it to get stuck in a local minimum, like the direct approach of Section 4.3.2. A well-known remedy is to reiterate the procedure several times, each time starting with a different set Ξ_0 . Unfortunately this approach may quickly become very time-consuming.

To circumvent the problems with iterative refinement, hill-climbing algorithms were developed which can escape from a local maximum. An example of such an approach is simulated annealing [Kirk83]. Hill-climbing methods every now and then accept a set with a *lower* value of the cost function, in contrast to the previously discussed methods. The number of down-hill steps is made to decrease when the algorithm proceeds. Therefore at the onset the methods behave like a random search and gradually turn more greedy.

Of the mentioned methods a specific form of iterative refinement was found to have the most satisfactory performance. The algorithm is designed to make a trade-off between too greedy behavior on the one hand, and a tendency to randomly jump through the complete search space on the other hand. This corresponds with a trade-off between a fast convergence and an increased possibility that D_{\max} is close to D_{opt} .

As mentioned, iteration i tries several modifications of Ξ_{i-1} , which are implemented as follows. To modify a set Ξ_{old} into a new set Ξ_{new} , the p^{th} member of Ξ_{old} is modified, where p is chosen at random between 0 and $k - 1$. The maximum amount by which the p^{th} test-point changes is δ_{\max} , which is initialized to $m - 1$. Now a number δ is chosen at random, in the range $1, \dots, \delta_{\max}$. The p^{th} test-point is now modified by either increasing or decreasing³ it by an amount δ . This modification is repeated until the changed test points is in the range $1, \dots, m - 1$ and different from all test points in Ξ_{old} . Such a

³ Whether a test-point is modified by increasing it or decreasing it is decided at random.

modification is called a *rearrangement*. By applying enough rearrangements each of the $\binom{m}{k}$ sets of test points may be obtained.

The starting set Ξ_0 contains k randomly chosen unique row indices of \mathbf{S}' . In the i^{th} iteration, Ξ_{i-1} is changed into Ξ_i by making the explained rearrangements of Ξ_{i-1} until $D_i > D_{i-1}$. In the first stages of the algorithm δ_{\max} is large, so the complete search space is quickly traversed. After each iteration δ_{\max} is decreased by a certain amount. Therefore, after many iterations the algorithm searches only the immediate environment of a set. Thus the algorithm exhibits greedy behavior in its final stages. The obtained effect is similar to that of a hill-climbing algorithm. Because in the beginning the rearrangements are rather large, a local maximum may be skipped by “jumping” over it. The speed with which the algorithm starts to exhibit greedy behavior is determined by the speed with which δ_{\max} is decreased. An exponential decrease with the number of iterations gives good results. The algorithm terminates if it cannot increase D with a certain maximum number of rearrangements.

Experiments showed that, if more than one test-point is changed per rearrangement, D_{\max} does not increase for a set of typical examples. Merely the convergence of the algorithm is slowed down. This suggests that by changing more than one test-point per iteration, the algorithm already starts to resemble a random search. Therefore only one test-point is changed in each rearrangement.

For larger problems a major computational task is the evaluation of $|\mathbf{S}'_{rr}|$, due to the large total amount of rearrangements. In this case, repeated computing of the SVD \mathbf{S}'_{rr} to obtain $|\mathbf{W}_r|$ takes a significant amount of the total time spent. The repeated computation of the determinant may be accomplished more efficiently with the QR decomposition [Golu89] of the $k \times r$ matrix $|\mathbf{S}'_{rr}|$, defined as

$$\mathbf{S}'_{rr} = \mathbf{QR}, \quad (5.7)$$

where \mathbf{Q} is a unitary $k \times k$ matrix, and \mathbf{R} is a $k \times r$ upper-right triangular matrix. From (5.7) it follows that

$$|\mathbf{S}'_n| = |\mathbf{QR}| = |\mathbf{R}| = \prod_{i=1}^r r_{ii}, \quad (5.8)$$

because \mathbf{Q} is orthogonal and \mathbf{R} is upper-triangular. Calculation of (5.8) is $O(r^3)$. However, the iterative procedure changes only one test-point per rearrangement. This corresponds with a dyadic (rank-1) update of \mathbf{S}'_n . In this case the QR-decomposition of the new \mathbf{S}'_n may be calculated with an efficient update procedure, which is $O(r^2)$ rather than $O(r^3)$ [Golub89].

For the circuit of Figure 3.1, the iterative method finds the same set of test points as the direct method. This is caused by the obvious simplicity of the surface of Figure 5.1. Chapter 7 provides larger examples for which the iterative method selects better test points than the direct method.

5.3 Selecting the Optimal Number of Test Points and Parameters

A complication with the iterative selection of test points is that it is not so easy as with the direct method (presented in Section 4.3.2) to determine the necessary and/or sufficient number of test points. The reason is that the iterative method does not add test points one by one, but evaluates a full set of k test points in each iteration. Therefore it is not possible to use criterion (4.35) with the iterative algorithm. This section presents an alternative approach.

The number of test points and parameters is adjusted to reduce the error in the prediction (4.17) as much as possible. There are four sources that contribute to this error:

1. The sensitivity matrix (4.8) is calculated from a circuit model composed of lumped passive or active circuit components, according to Sections 3.1.1 and 3.2.4. This

model may not accurately represent the actual circuit. This is often the case when an integrated circuit is modeled. The resulting model error has its influence in any test method.

2. Even if the circuit model is exact, (4.8) is still a first-order approximation of the real (generally non linear) circuit behavior. If the parameter deviations are not infinitesimally small then this approximation introduces a certain error in the predictions.
3. The number of columns that is included in \mathbf{S}'_r corresponds to the number of optimal parameters, as explained in Section 4.3.2. If this number is too low then there are not enough model parameters (i.e., degrees of freedom) to account for the deviating circuit behavior.
4. The measurement errors influence the precision of the predictions, as discussed in Section 4.2.

Once the circuit model is determined it is not possible to reduce the influence of error source 1. Also it appears that the second error source cannot be eliminated without making repeated measurements, as explained in Section 4.3.5. However, items 3 and 4 may be addressed by adjusting the number of test points and parameters as follows.

Before test-point selection starts, a “reasonable” number of columns of \mathbf{S}' is determined. A quantitative estimate follows directly from the discussion about the approximation of matrices in Section 3.2.3. The number of columns is determined as the lowest p for which

$$\frac{\sum_{i=1}^p w_i}{\sum_{i=1}^r w_i} \geq \gamma, \text{ with } 0 < \gamma \leq 1, \quad (5.9)$$

where w_i denotes the i^{th} singular value of \mathbf{S} , and $r = \text{rank}(\mathbf{S})$. According to (5.9), the number of columns is p if \mathbf{S}_p (conform Theorem 3.5) approximates \mathbf{S} with a relative precision of at least γ . This translates to a certain relative precision of the response

predictions, taking into account the magnitude of the measurement errors, according to (4.8). Because criterion (5.9) is scale free with respect to the circuit parameters as well as to the circuit outputs, a reasonable estimate of p is obtained for many different examples, for the same value of γ . Good results were obtained with $\gamma = 0.98$.

Next $k = p$ test points are selected with the iterative approach discussed in the previous section. Next the algorithm starts to iteratively adjust k (the number of test points) and p (the number of columns of \mathbf{S}'_{π} , or model parameters). In each stage of this process, one of the following changes is made.

- *Add a test-point*

This reduces the influence of measurement errors on the prediction, because it will cause system (4.12) to be more overdetermined.

- *Remove a test-point*

This removes redundant test points that do not contribute to the precision of the prediction (4.17).

- *Add a parameter*

This increases the precision of the linear model, conform item 3 above.

- *Remove a parameter*

This reduces the accuracy of the linear model. For example, if the measurement errors are large then the model precision does not need to be high.

The changes are restricted by the requirement that the number of rows of \mathbf{S}'_{π} must be larger than or equal to the number of columns⁴. After a change is made, it is evaluated. If the change results in an increased precision of the response prediction then it is incorporated. When the precision cannot be further enhanced the algorithm terminates.

⁴ The reason is explained in Section 4.1.2.

The remaining issue is how to evaluate the effect of a change, without making measurements of an actual circuit. This is accomplished by simulated measurements of several deviating circuits. It is assumed that estimates of the expected parameter deviations and measurement errors are available. The parameter estimates are used to simulate the complete response of a deviating circuit. Measurements of this circuit are simulated by adding random measurement errors to the circuit response at the k selected test points. Then the response prediction is calculated from (4.14) and (4.17). The error in the prediction is computed by subtracting from it the simulated response of the deviating circuit. The RMS value of the error is a measure of the precision of the prediction. This procedure is repeated several times, for a number of deviating circuits. The mean of the RMS errors is used as a criterion for the quality of S'_{α} , given a certain number of test points and parameters.

The procedure has been successfully used to refine the number of test points and parameters for a number of circuits. An example in Chapter 7 applies the procedure to a D/A converter to reduce the influence of large measurement errors.

The computational expense of the presented refinement method may be high. However this is not important, because the computations are performed off-line. Once the optimal number of test points and parameters is determined, the complete response of a device may be predicted very efficiently from measurements, by calculating (4.14) and (4.17), as explained in section 4.2. This concludes the discussion of iterative test-point selection.

5.4 Discussion

This section presented an iterative approach to optimal test-point selection. The method judges the optimality of a particular set of test points with the D -optimality criterion, which is widely used in the design of experiments. According to this criterion the determinant $|S'_{\alpha}|$ is maximized. It was shown that this minimizes the influence of random measurement errors. The discussed iterative refinement can escape from a

local maximum of $|\mathbf{S}'_{\pi}|$ while retaining its convergence properties. The result is a method that selects truly D -optimal test points for many examples.

It should be mentioned that it is not *guaranteed* that the discussed method obtains a D -optimal set of test points. Unfortunately, it is in most cases not feasible to verify whether or not a selected set of test points *is* in fact D -optimal: the number of possible sets of test points is in general so large that it is not possible to evaluate the quality of each set. However, the examples demonstrate that the iterative method selects test points of higher quality than the direct approach discussed in Section 4.3.2.

An extension to the iterative approach was discussed. It allows a refinement of the number of test points and parameters, minimizing the error in the response prediction that is due to some of the error sources that were identified.

The discussed approach demonstrates that the SVD may be fruitfully combined with a powerful iterative algorithm to select test points. The result is a versatile test method that offers some distinct advantages over existing methods, which may be summarized as follows.

- As discussed in Section 4.4, the SVD deals with rank-deficiency of the linear model in an optimal manner.
- The SVD exists for a larger class of matrices than any other matrix decomposition [Golu89].
- The quality of the test points selected by the *iterative* algorithm will be higher⁵ (in the D -optimal sense) than can be obtained by a *direct* method. The main reason is that the direct approaches (like the method of Section 4.3.2, [Hemi90] and [Sten87]) suffers from a kind of greedy behavior.

⁵ This will be demonstrated with an example in Chapter 7.

The algorithm of Section 5.2 allows the number of test points and testable parameters to be determined independently. This is not possible with existing approaches as [Hemi90] and [Sten87], where the number of test points always equals the number of parameters.

6

Application of PL Techniques to Analog Testing

This chapter describes an approach to analog testing that uses a combination of piecewise-linear (PL) modeling and interval analysis [Leen93a], [Leen93b]. Section 6.1 gives an outline of the discussed approach. Section 6.2 introduces the closed-form PL model that is used to approximate the behavior of non linear circuit components. Section 6.3 describes interval analysis, illustrating with a geometrical example how the complete solution of a set of linear inequalities may be obtained. Also the corresponding matrix method is introduced there, with an outline of a very efficient implementation. Then Section 6.4 applies the discussed techniques to analog testing. The approach that is presented there is based on a solution method for non linear equalities with bounded parameters, discussed in [Leen91]. It will become clear that this approach may form the basis of a more complete method for analog testing.

6.1 Introduction

This chapter presents an application of PL techniques and interval analysis to the testing of analog integrated circuits. The combination of these techniques has been used already in the design of analog circuits [Leen90], [Leen91]. A major characteristic of that approach is that it deals with solution spaces, instead of solution points. The end result is a solution space that defines all allowable combinations of design parameters, taking into account user-specified constraints. An optimal choice for the

design parameters may then be made by applying a certain optimization criterion on this solution space.

The mentioned techniques may also be applied to analog testing. PL models are used to model the non linear behavior of circuit components. A PL model description by van Bokhoven is used, as explained in Section 6.2. Loosely speaking, this is an implicit PL model description, containing a collection of linear descriptions that collectively define the non linear component behavior. Together with certain constraints (which define, for example, allowable component or circuit behavior) these linear descriptions form a set of linear equalities with bounded parameters. One technique for solving equalities of this type is used in an algorithm by Tschernikowa, which is introduced in Section 6.3. Section 6.4 shows that by using PL modeling and interval analysis a multi-dimensional region (solution space) is obtained that defines the behavior of a circuit that functions according to specifications. By making measurements, additional linear relations are obtained. When these are taken into account the dimension of the solution space is reduced. The circuit does not work according to specifications if the solution space is empty after the measurements are taken into account as additional constraints.

Note that, in contrast to the methods of Chapters 4 and 5, the principles discussed in this chapter do not constitute a complete test method. For example, the selection of measurements has not been investigated. However, the discussed principles may be used as the basis of a test method.

6.2 Piecewise-Linear Modeling

With piecewise-linear (PL) modeling, the non linear behavior of circuit components is approximated by a collection of affine mappings. Each mapping is valid on a certain polytope, a convex region bounded by hyperplanes. In principle the PL approximation may be made arbitrarily precise by increasing the number of polytopes. However the attainable precision depends on the modeling freedom that is allowed by a specific PL model description, as comprehensively discussed in [Keve92]. Thus the choice for a

specific PL model description will be determined, amongst other things, by the desired modeling freedom.

A major advantage of PL modeling is that all models of a certain description are of the same form for all kinds of components, ranging from precisely modeled non linear components to macro models of digital logic. The result is that a single PL solution algorithm is inherently capable of mixed mode simulation, eliminating the need to couple different specialized simulators [Keve91]. A second advantage of a PL simulator is that the convergence behavior is better than that of a classical simulator that uses a particular form of the Newton-Raphson algorithm to solve the non linear circuit equations [Keve91].

A distinction can be made between explicit and implicit PL model descriptions. Explicit descriptions, e.g., those presented in [Chua77] and [Kahl90], store the PL mapping in a canonical form from which the linear mapping in a specific polytope is directly obtained. In contrast, with implicit model descriptions an upgrade needs to be performed on the current mapping to obtain the mapping in a newly entered polytope. An advantage of implicit models is the larger class of functions that can be described, such as one-to-many mappings. However often this is accomplished at the cost of a slow and complex algorithm to calculate the mapping in a certain polytope.

This section briefly discusses an implicit PL model [Bokh86], henceforward referred to as the Bokh2 model. It defines a PL mapping, with input $\mathbf{x} \in \mathbf{R}^n$ and output $\mathbf{y} \in \mathbf{R}^m$, as

$$\begin{cases} \mathbf{ly} + \mathbf{Ax} + \mathbf{Bu} + \mathbf{f} = \mathbf{0} & (6.1) \\ \mathbf{Dy} + \mathbf{Cx} + \mathbf{lu} + \mathbf{g} = \mathbf{j} & (6.2) \\ \mathbf{u}^T \mathbf{j} = 0 \quad \forall_i u_i, j_i \geq 0 & (6.3) \end{cases}$$

where $\mathbf{u}, \mathbf{j} \in \mathbf{R}^k$ contain the so called state variables. For $\mathbf{u} = \mathbf{0}$, (6.1) defines the current linear mapping as $\mathbf{y} = -\mathbf{Ax} - \mathbf{f}$. In this case $\mathbf{j} \geq \mathbf{0}$ because of the complementarity condition (6.3). Therefore (6.2) describes the current polytope as the convex region defined by $\mathbf{Dy} + \mathbf{Cx} + \mathbf{g} \geq \mathbf{0}$. If (due to a change in the independent

variable x) $u_i > 0$ and $j_i = 0$ for $1 \leq i \leq k$, a pivot operation is performed. In effect this operation exchanges u_i and j_i , yielding an adapted $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{j}}$ so that again (6.3) is satisfied with $\tilde{\mathbf{u}} = \mathbf{0}$ and $\tilde{\mathbf{j}} \geq \mathbf{0}$. The pivot operation performs a rank-1 update on the linear mapping (6.1). After this update (6.1) is the linear mapping in the polytope described by (6.2) for $\mathbf{j} \geq \mathbf{0}$. This is the polytope that has been entered because of the change in \mathbf{x} . To find out which u_i has become positive, a modified form of the Katzenelson algorithm may be used [Keve91]; this is not further discussed here.

One way of understanding the mathematical formulation (6.1) - (6.3) of the PL mapping is to regard it as a description of the input-output relation of some static system composed of several linear and PL elements. As depicted in Figure 6.1 a, such a system can be seen as a linear multi-port network with k of its ports connected to ideal diodes. The voltage/current characteristic of an ideal diode is depicted in Figure 6.1 b (notice the sign convention). The state vectors \mathbf{u} and \mathbf{j} contain the voltages and currents of the ports loaded with the diodes. Thus \mathbf{u} and \mathbf{j} satisfy (6.3). The remaining port variables are elements of \mathbf{x} and \mathbf{y} , which may be voltages or currents. The state of the diodes (blocking or conducting) influences the topology and thus the linear relation between \mathbf{x} and \mathbf{y} .

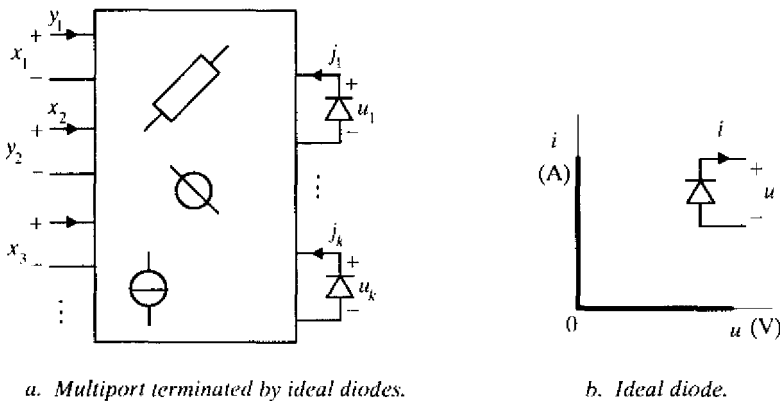


Figure 6.1. The PL network description.

As an example of a PL component model, consider the NMOST transistor depicted in Figure 6.2a. Its behavior may be described¹ by the PL model

$$\mathbf{I} \begin{bmatrix} i_d \\ i_g \\ i_s \end{bmatrix} + \begin{bmatrix} -\alpha & 0 & \alpha \\ 0 & 0 & 0 \\ \alpha & 0 & -\alpha \end{bmatrix} \begin{bmatrix} v_d \\ v_g \\ v_s \end{bmatrix} + \begin{bmatrix} -\beta & \beta \\ 0 & 0 \\ \beta & -\beta \end{bmatrix} \mathbf{u} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \mathbf{0} \quad (6.4)$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_d \\ i_g \\ i_s \end{bmatrix} + \begin{bmatrix} 0 & -1 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} v_d \\ v_g \\ v_s \end{bmatrix} + \mathbf{I} \mathbf{u} + \begin{bmatrix} V_t \\ V_t \end{bmatrix} = \mathbf{j}$$

$$\mathbf{u}^T \mathbf{j} = 0 \quad \forall_i u_i, j_i \geq 0$$

where V_t denotes the threshold voltage. The constants α and β are defined as

$$\alpha = \frac{\mu C_{ox}}{2} \frac{W}{L} \lambda$$

$$\beta = \frac{\mu C_{ox}}{2} \frac{W}{L} (\lambda - 1) \quad (6.5)$$

$$\lambda = 1/V_A$$

where W and L denote the physical dimensions of the transistor, μ is the electron mobility, C_{ox} is the capacitance of the gate oxide, and V_A is the Early voltage. Figure 6.2b depicts a graph of the NMOST behavior.

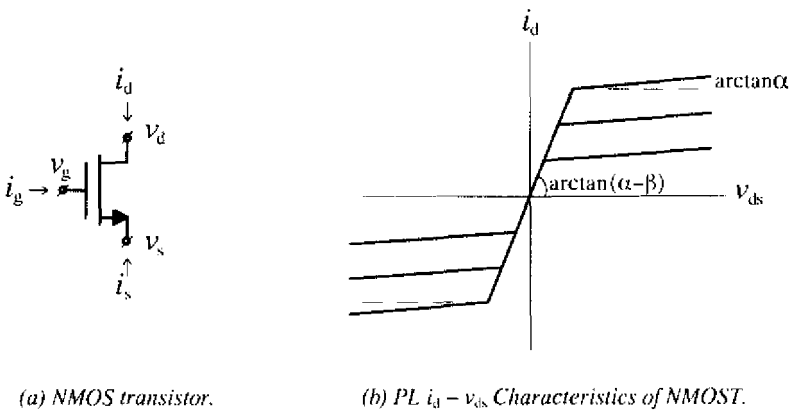


Figure 6.2. Piecewise-linear model of an NMOS transistor.

¹ The presented PL model is kept simple for illustration purposes. More realistic models are available.

The PL model may be made more accurate at the expense of extra segments, in which case there will be more elements in vectors \mathbf{u} and \mathbf{j} .

6.3 Solving Sets of Linear Inequalities

As mentioned in Section 6.1, PL modeling will be combined with a method that solves linear equalities. This method will be discussed in this section. The result is a method that obtains the solution space of a set of non linear equalities with bounded parameters. This method may be applied to analog testing, as will be explained in Section 6.4.

This section presents a method that finds the complete solution space of a set of linear inequalities

$$\begin{cases} \mathbf{Ax} \leq \mathbf{0} & (6.6) \\ \mathbf{x} \geq \mathbf{0} & (6.7) \end{cases}$$

where $\mathbf{A} \in \mathbf{R}^{m \times n}$ and $\mathbf{x} \in \mathbf{R}^n$. These inequalities may be combined into a single set

$$\mathbf{C} \mathbf{x} \leq \mathbf{0}, \text{ with } \mathbf{C} \triangleq \begin{bmatrix} \mathbf{A} \\ -\mathbf{I} \end{bmatrix}, \quad (6.8)$$

where $\mathbf{C} \in \mathbf{R}^{(m+n) \times n}$ and $\mathbf{I} \in \mathbf{R}^{n \times n}$. Note that $\text{rank}(\mathbf{C}) = n$, due to the identity matrix. It will be shown that the method can also deal with linear *equalities*.

Section 6.3.1 provides a geometrical interpretation of the solution process of (6.8). Then Section 6.3.2 presents a matrix method that allows a computer to obtain this solution space. Note that these sections are intended as an outline, and do not provide a rigid exposition of the discussed methods. A more elaborate treatment is found in [Tsch71] and [Solo80].

6.3.1 A Geometrical Solution Method

This section presents a geometrical interpretation of the solution method for sets of linear inequalities that is described in [Tsch71]. Historically, the geometrical approach precedes the matrix algorithm of [Tsch71].

The inequalities (6.6) define a set H_1, \dots, H_m of hyperplanes in \mathbf{R}^n , according to

$$H_i: \mathbf{a}_i^T \mathbf{x} = 0, \quad i \in [1, \dots, m], \quad (6.9)$$

where \mathbf{a}_i^T denotes the i^{th} row of \mathbf{A} . Likewise (6.7) defines a set of hyperplanes H'_1, \dots, H'_n according to

$$H'_i: \mathbf{e}_i^T \mathbf{x} = 0, \quad i \in [1, \dots, n], \quad (6.10)$$

where \mathbf{e}_i denotes the i^{th} unit vector of \mathbf{R}^n .

Each hyperplane divides \mathbf{R}^n into two half spaces. The solution space of the corresponding inequality is one of these half spaces, bounded by (and including) the hyperplane. Thus there are $m + n$ half spaces associated with (6.8). The solution space of (6.8) is found as the intersection of all $m + n$ half spaces. Because the inequalities are homogeneous, the bounding hyperplanes all pass through the origin. It is well known that in this case the intersection of the $m + n$ half spaces is a convex polyhedral cone [Solo80]. The vertex of the cone is the origin of \mathbf{R}^n . The edges, or rays, of the cone are called fundamental solutions, formally defined as follows.

Definition 6.1. Let a set of inequalities (6.8) be given. For a *fundamental solution* $\mathbf{v} \in \mathbf{R}^n$ of this set there are $n - 1$ linearly independent inequalities that are transformed into equalities. Thus each fundamental solution is in a one-dimensional subspace of \mathbf{R}^n .

The fundamental solutions are independent in the sense that one such solution cannot be formed as a *positive* linear combination of other fundamental solutions. Because the

vertex of the solution space is at the origin, a fundamental solution is completely defined by a vector of arbitrary length. Because the solution space of (6.6) and (6.7) is convex, and the fundamental solutions are at the edges of the solution space, each solution can be formed as a positive linear combination of fundamental solutions. This leads to the following definition.

Definition 6.2. Let $\mathbf{v}_1, \dots, \mathbf{v}_k$ be the fundamental solutions of (6.8). Then the *fundamental solution space* \aleph of (6.8) is

$$\aleph: \left\{ \mathbf{x} \mid \mathbf{x} = \sum_{i=1}^k p_i \mathbf{v}_i, \text{ with } \forall_{i \in \{1, \dots, k\}} p_i \in \mathbf{R} \wedge p_i \geq 0 \right\}.$$

This definition describes the interior of a convex polyhedral cone in terms of its rays. An example, in \mathbf{R}^3 with 5 rays, is depicted in Figure 6.3. The solution space consists of all points on the surface or in the interior of the cone.

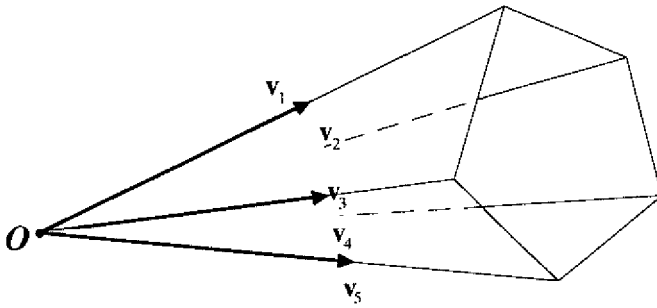


Figure 6.3. A convex polyhedral cone in \mathbf{R}^3 .

From the preceding discussion it follows that once the fundamental solutions of (6.8) are computed, its solution space is obtained directly from Definition 6.2. So the problem is to find the fundamental solutions of (6.8), which may be accomplished as follows. The method generates a set of matrices $\mathbf{C}^{(i)}$ starting with $\mathbf{C}^{(0)}$, satisfying

$$\mathbf{C}^{(i)} \mathbf{x} \leq 0, \text{ with } \mathbf{C}^{(i)} \stackrel{\Delta}{=} [-\mathbf{I}], \quad (6.11)$$

that has fundamental solutions $\mathbf{V}^{(0)} \triangleq \{\mathbf{v}_1^{(0)}, \dots, \mathbf{v}_n^{(0)}\}$. Because of the identity matrix it follows directly that $\mathbf{V}^{(0)} = \{\mathbf{e}_1, \dots, \mathbf{e}_n\}$. Therefore, according to Definition 6.2 $\mathbf{V}^{(0)}$ describes the complete solution space of (6.11) as the first orthant of \mathbf{R}^n . Next the inequalities (6.6) are added one by one to (6.11)². Each time the set of fundamental solutions is changed, as described below. After a maximum of m iterations, the complete solution space of (6.8) is obtained.

To understand how a set of fundamental solutions is changed when an inequality is added, suppose that iteration $j - 1$ already obtained $\mathbf{V}^{(j-1)} = \{\mathbf{v}_1^{(j-1)}, \dots, \mathbf{v}_r^{(j-1)}\}$, the solution space of

$$\mathbf{C}^{(j-1)} \mathbf{x} \leq \mathbf{0}. \quad (6.12)$$

Here $\mathbf{C}^{(j-1)}$ was created by adding $j - 1$ inequalities of (6.6) to (6.11). The j^{th} iteration adds the j^{th} inequality of (6.6) to (6.12), giving a total of $n + j$ inequalities that must be simultaneously satisfied. Geometrically, the cone associated with $\mathbf{V}^{(j-1)}$ is intersected with hyperplane $H_j : \mathbf{a}_j^T \mathbf{x} = 0$. This is shown in Figure 6.4 for the example cone of Figure 6.3 with 5 fundamental solutions ($r = 5$). The new set $\mathbf{V}^{(j)}$ is computed from $\mathbf{V}^{(j-1)}$ in two steps:

Step 1 Each $\mathbf{v}_i^{(j-1)}$ for which $\mathbf{a}_j^T \mathbf{v}_i^{(j-1)} \leq 0$ remains a fundamental solution after the j^{th} iteration. In Figure 6.3 this step selects $\mathbf{v}_4^{(j-1)}$ and $\mathbf{v}_5^{(j-1)}$: the two fundamental solutions under H_j .

Step 2 New fundamental solutions are created at the intersection of H_j and the existing cone. In Figure 6.4 the two new fundamental solutions are indicated with dotted lines.

² The order in which the inequalities are added is arbitrary.

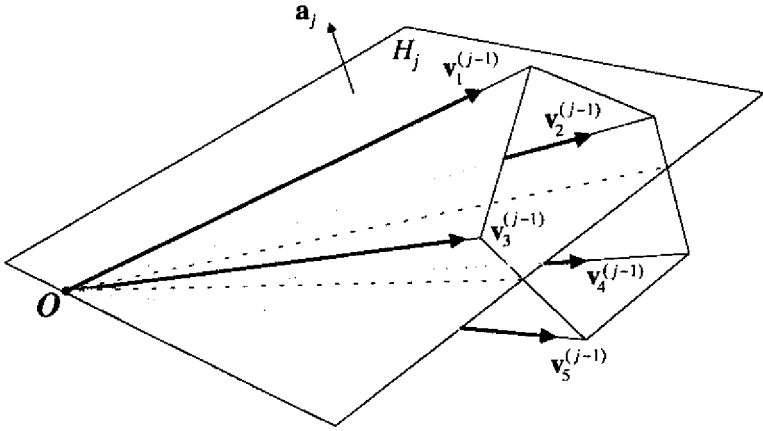


Figure 6.4. Intersecting the cone with hyperplane H_j .

Each new fundamental solution in this step is formed as a linear combination

$$\mathbf{v}^{(j)} = \mathbf{v}_p |\mathbf{a}_j^T \mathbf{v}_q| + \mathbf{v}_q |\mathbf{a}_j^T \mathbf{v}_p|, \quad (6.13)$$

where \mathbf{v}_p and \mathbf{v}_q are selected from $\mathbf{V}^{(j-1)}$ according to the following criteria:

- $\mathbf{a}_j^T \mathbf{v}_p < 0$ and $\mathbf{a}_j^T \mathbf{v}_q > 0$. This ensures that \mathbf{v}_p and \mathbf{v}_q are on opposite sides of H_j .
- For \mathbf{v}_p as well as for \mathbf{v}_q the same $n - 2$ linearly independent inequalities of (6.12) are transformed into equalities. This ensures that \mathbf{v}_p and \mathbf{v}_q are in the same two-dimensional plane.

The choice of the multiplication factors in (6.13) ensures that the new solution $\mathbf{v}^{(j)}$ is in H_j . This follows immediately from

$$\begin{aligned} \mathbf{a}_j^T \mathbf{v}^{(j)} &= \mathbf{a}_j^T \mathbf{v}_p |\mathbf{a}_j^T \mathbf{v}_q| + \mathbf{a}_j^T \mathbf{v}_q |\mathbf{a}_j^T \mathbf{v}_p| \\ &= 0 \end{aligned}$$

since $\mathbf{a}_j^T \mathbf{v}_p / \mathbf{a}_j^T \mathbf{v}_q < 0$ (because \mathbf{v}_p and \mathbf{v}_q are selected at different sides of H_j). Thus the added inequality is turned into an equality for $\mathbf{v}^{(j)}$. Because for \mathbf{v}_p and \mathbf{v}_q the same $n - 2$ inequalities of (6.12) are transformed into equalities

this is also true for $\mathbf{v}^{(j)}$. Furthermore the added inequality is linearly independent of the mentioned $n - 2$ inequalities³. Thus for $\mathbf{v}^{(j)}$ a total of $n - 1$ linearly independent inequalities are transformed into equalities, and according to Definition 6.1 this makes $\mathbf{v}^{(j)}$ a fundamental solution. With this approach fundamental solutions are created for all pairs \mathbf{v}_p and \mathbf{v}_q that satisfy the specified conditions.

In Figure 6.4 the pairs $\{\mathbf{v}_3^{(j-1)}, \mathbf{v}_5^{(j-1)}\}$ and $\{\mathbf{v}_2^{(j-1)}, \mathbf{v}_4^{(j-1)}\}$ are selected according to the two criteria. The pairs form new fundamental solutions $\mathbf{v}_3^{(j)}$ and $\mathbf{v}_4^{(j)}$, respectively, as depicted in Figure 6.5. Furthermore, according to step 1, $\mathbf{v}_1^{(j)} = \mathbf{v}_5^{(j-1)}$ and $\mathbf{v}_2^{(j)} = \mathbf{v}_4^{(j-1)}$. Geometrically, the method omits the part of the cone that is on the wrong side of H_j , as shown in Figure 6.5.

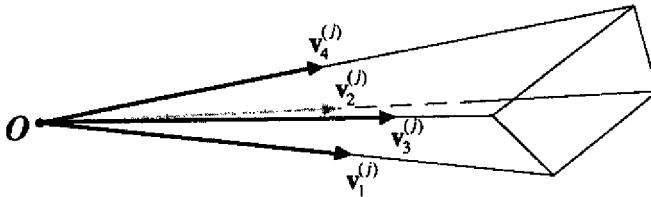


Figure 6.5. Solution space after iteration j .

The next section presents a matrix method for the discussed approach.

6.3.2 The Matrix Method of Tschernikowa

The algorithm explained in the previous section is very inefficient in terms of the number of calculations that must be performed. The cause is the need to evaluate the inner product $\mathbf{a}_j^T \mathbf{v}$ to check if \mathbf{v}_j is above, in or under hyperplane H_j . This gives rise to

³ Otherwise it would turn into an equality for \mathbf{v}_p and \mathbf{v}_q which contradicts criterion a.

many floating point multiplications in the course of the algorithm. In 1965, Tschernikowa presented an efficient matrix manipulation algorithm that has a higher computational efficiency [Tsch71].

This approach expresses system (6.8) in a more efficient matrix notation. At the start of the algorithm a start tableau is set up as

$$\mathbf{T}^{(0)} = \left[\mathbf{T}_1^{(0)} \mid \mathbf{T}_2^{(0)} \right] = \left[\mathbf{I} \mid \mathbf{A}^T \right], \quad (6.14)$$

where the superscript (\cdot) denotes the iteration number. The rows of $\mathbf{T}_1^{(0)}$ are the coordinates of the fundamental solutions of (6.11), with respect to the orthonormal base of \mathbf{R}^n . The coordinates of the normals of the hyperplanes (6.9) are entered as columns in $\mathbf{T}_2^{(0)}$. The central aspect of the matrix algorithm may be loosely described as follows:

After the j^{th} iteration the columns of $\mathbf{T}_2^{(j)}$ contain the coordinates of the normals of the m hyperplanes (6.9). These normals are expressed as linear combinations of the fundamental solutions contained in the rows of $\mathbf{T}_1^{(j)}$:

$$\forall_{i \in [1, \dots, m]} \mathbf{a}_i = \sum_{k=1}^n c_{ik} \mathbf{v}_i^{(j)}, \text{ with } c_{ik} = \left(\mathbf{T}_2^{(j)} \right)_{ik}, \quad (6.15)$$

where $\mathbf{v}_i^{(j)}$ is the k^{th} row of $\mathbf{T}_1^{(j)}$. Thus the coordinates of the hyperplanes are not given with respect to the orthonormal base of \mathbf{R}^n . The advantage of storing the hyperplanes in this specific format is that it is now immediately clear if \mathbf{v}_i is above, in or under hyperplane H_j by evaluating entry c_{ij} of $\mathbf{T}_2^{(j)}$. This entry will be positive, zero or negative respectively. Consequently the tableau method saves many calculations.

Each iteration of the algorithm consists of the two steps explained in the previous section. Iteration j adds a new hyperplane by selecting an arbitrary (but not previously

selected) column from $\mathbf{T}_2^{(j-1)}$, say column l . This column is denoted as \mathbf{a}_l , because it contains the normal of hyperplane H_l . In addition, the following notation is used:

- $\mathbf{K}^{(j)} \triangleq [\mathbf{T}_1^{(j-1)} \mid \mathbf{T}_2^{(j-1)}]$, where $\mathbf{T}_2^{(j-1)}$ contains the $j-1$ already treated columns of $\mathbf{T}_2^{(j-1)}$.
- $S(i)$ denotes the set of columns of $\mathbf{K}^{(j)}$ with a zero in row i .
- $S(i_1, i_2)$ denotes the collection of columns for which $S(i_1, i_2) = S(i_1) \cap S(i_2)$, where the i_1^{th} and i_2^{th} entry in column \mathbf{a}_l have opposite signs.

$\mathbf{T}_2^{(j)}$ is derived from $\mathbf{T}_2^{(j-1)}$ in two steps:

Step 1 Copy into $\mathbf{T}^{(j)}$ all rows from $\mathbf{T}^{(j-1)}$ that have a non-positive entry in \mathbf{a}_l . This selects the fundamental solutions that are in H_l , or on the correct side of H_l .

Step 2 If, for each row i , $S(i_1, i_2) \not\subseteq S(i)$ ($i_1 \neq i, i_2 \neq i$), then a positive linear combination of rows i and j is placed in $\mathbf{T}^{(j)}$, such that the entry in \mathbf{a}_l becomes zero. In this case the pair (i, j) is called a valid row pair. Geometrically, this step selects two fundamental solutions that lie on opposite sides of H_l and are in a single plane. In this case a new fundamental solution may be formed as a linear combination that lies in H_l (causing a zero in column l of $\mathbf{T}_2^{(j)}$), as explained in the previous section.

When a hyperplane is added for which \mathbf{a}_l is strictly positive then the solution space is empty and the algorithm stops. Furthermore, if there are no positive entries in \mathbf{a}_l then the algorithm proceeds with the next iteration. In this case the solution space already is on the correct side of the added hyperplane.

In the course of the algorithm the number of fundamental solutions may become quite large. This causes a large amount of time to be spent in the search for valid pairs in

step 2. Therefore this step should be implemented as efficiently as possible. An efficient implementation is based on the following observation: With regard to the search for valid row pairs each element of $\mathbf{T}^{(j-1)}$ represents just a single bit of information. To explore this fact $\mathbf{T}^{(j-1)}$ is translated into a binary form at the start of the j^{th} iteration. Non-zero elements are represented by "1" and zero elements by "0". Suppose that on a particular computer an integer is stored in 64 bits. Then up to 64 elements of a row of $\mathbf{T}^{(j-1)}$ may be stored in a *single* integer. It follows that if the number of columns of $\mathbf{T}^{(j-1)}$ is p then the number of integers that is needed to store a row is just $\lfloor p \rfloor + 1$. In iteration j the two steps of the algorithm use the binary representation of $\mathbf{T}^{(j-1)}$ in the following manner.

In step 1 the rows with a non-positive entry in \mathbf{a}_i are copied from $\mathbf{T}^{(j-1)}$ to $\mathbf{T}^{(j)}$. Furthermore sets P and N of rows are formed that have a positive or negative entry in \mathbf{a}_i of $\mathbf{T}^{(j-1)}$, respectively. Thus the implementation of step 1 is straightforward.

In step 2 the binary form of $\mathbf{T}^{(j-1)}$ is used to find all valid row pairs with two sub steps in a computationally efficient manner:

- a. First $S(i_1, i_2)$ is determined by a bit wise OR of rows i_1 and i_2 , for all combinations of i_1 and i_2 for which $i_1 \in P$ and $i_2 \in N$. The result of this logical operation is a binary representation of $S(i_1, i_2)$, where its elements are denoted by "0" bits.
- b. This step implements the check $S(i_1, i_2) \not\subset S(i)$ ($i_1 \neq i, i_2 \neq i$). At this stage $S(i_1, i_2)$ is available in binary format. A bit wise OR is performed with $S(i_1, i_2)$ and each row of $\mathbf{T}^{(j-1)}$. If for each row the result of the OR is identical to $S(i_1, i_2)$ then $S(i_1, i_2) \not\subset S(i)$ ($i_1 \neq i, i_2 \neq i$). This selects a pair of rows with the maximum number of common zeros, ensuring that the two fundamental solutions that correspond with these rows are in a two-dimensional plane [Tsch71].

After the row pairs are selected, the “full” representation of $\mathbf{T}^{(j-1)}$ is used to form new fundamental solutions in $\mathbf{T}^{(j)}$. After $\mathbf{T}^{(j)}$ is derived from $\mathbf{T}^{(j-1)}$ it is translated into binary format, and the procedure is repeated for iteration $j + 1$.

The use of the binary version of the tableau results in reduction of used memory as well as an improvement in the execution speed. This is important, since the Tschernikow algorithm is the limiting factor in the speed of the approach of Section 6.4.

It has been shown how the Tschernikowa algorithm solves homogeneous sets of linear inequalities of the form (6.8). However by using slack variables, a technique known from linear programming, any set of linear inequalities may be translated into this form. This also includes inequalities with bounded parameters, where a specific interval is defined for each of the variables \mathbf{x} . The use of slack variables with the Tschernikowa algorithm is treated with an example in [Leen90] and [Leen91]. Finally, linear equalities are treated in much the same manner as linear inequalities. The only difference is in step 1 of the algorithm, where only the rows of $\mathbf{T}^{(j-1)}$ that have a zero in the new column are transferred to $\mathbf{T}^{(j)}$.

6.4 Analog Testing with PL Modeling and Interval Analysis

Here the discussed principles are applied to analog testing. First Section 6.4.1 shows how PL modeling and interval analysis may be combined to solve non linear equalities with bounded parameters [Leen91]. Then Section 6.4.2 applies this technique to analog testing.

6.4.1 Solving Sets of Non Linear Equalities with Bounded Parameters

This section explains how PL modeling may be used to obtain the complete solution space of the set of non linear equations defined by

$$\mathbf{y} = \mathbf{f}(\mathbf{x}), \quad (6.16)$$

where $\mathbf{x} \in \mathbf{R}^n$ and $\mathbf{y} \in \mathbf{R}^m$. The non linear mapping $\mathbf{f}: \mathbf{R}^n \rightarrow \mathbf{R}^m$ is approximated by a PL model of the form (6.1)-(6.3). Furthermore, each of the elements of \mathbf{x} and \mathbf{y} is bounded by an interval. Each input x_i is bounded as

$$x_i \in [a_i, b_i], \text{ with } a_i, b_i \in \langle -\infty, \infty \rangle \text{ and } a_i < b_i, \quad (6.17)$$

and these bounds collectively define a region $S_x \in \mathbf{R}^n$. Likewise, bounds for \mathbf{y} define a region $S_y \in \mathbf{R}^m$. In the Euclidean space S_x and S_y define a hypercube C , possibly unbounded at one or more sides. Only the part of the solution space of (6.16) that is inside this hypercube is valid. This part may be obtained in essentially two steps.

Step 1 The intersection \aleph_{dom} of the variables' domains (S_x and S_y) and the model polytopes (6.2) is determined as a collection of convex regions. This collection describes the domain on which linear mappings given by (6.1) are valid, taking into account the bounds on the variables.

In principle, \aleph_{dom} may be obtained by intersecting all model polytopes with S_x and S_y . However, a PL model with k hyperplanes has a total number of 2^k model polytopes. (Thus \aleph_{dom} consists of a maximum of 2^k convex regions.) Because often many of these polytopes are inaccessible, this approach would be inefficient. Therefore, the following approach is used [Leen91].

A hyperplane

$$H_i : \mathbf{d}_i^T \mathbf{y} + \mathbf{c}_i^T \mathbf{x} + g_i = 0 \quad (6.18)$$

is associated with the i^{th} inequality of (6.2). H_i is an edge of \aleph_{dom} if and only if there exists a set (\mathbf{x}, \mathbf{y}) that satisfies (6.18) within the domains S_x and S_y . This may be checked with the technique of Section 6.2. If no hyperplanes intersect with S_x and S_y then \aleph_{dom} is situated inside a single polytope. However, if there are l hyperplanes that are a boundary of \aleph_{dom} , then just the 2^l corresponding polytopes must be intersected with S_x and S_y . This obtains the domain \aleph_{dom} .

on which each mapping (6.1) is valid. Because often $l \ll k$, this technique is faster than computing the intersection of each of the 2^k model polytopes with S_x and S_y .

To illustrate, Figure 6.6 depicts the results of this step for the NMOS PL model (6.4). The four polytopes are bounded by hyperplanes H_1 and H_2 . In each polytope the operating mode of the NMOST is indicated. Because according to (6.4) H_1 and H_2 are defined in the x -region, \mathcal{K}_{dom} is obtained as the intersection of S_x and the four polytopes. In this case, only three polytopes intersect S_x . Inside \mathcal{K}_{dom} (which is marked as a shaded area) $v_{\text{ds}} \geq 0.5V_t$. Figure 6.6 shows that in this case \mathcal{K}_{dom} is a collection of three convex areas, labeled I, II and III.

Step 2 The linear mapping that is valid in each region of \mathcal{K}_{dom} is combined with the description of that region, as obtained in Step 1. Again this is accomplished with the Tschernikowa algorithm.

The result of the two steps is a collection of l convex spaces

$$\mathcal{K}_{\text{tot}} = \bigcup_{i=1}^l \mathcal{K}_i,$$

that describes the total solution space of (6.16), given certain bounds on x and y . Each space \mathcal{K}_i is a fundamental solution space according to Definition 6.2.

The discussed method finds the complete solution space of a single PL model, where certain bounds are defined for the model variables. The next section applies the discussed combination of PL modeling and interval analysis to analog testing, and presents an example.

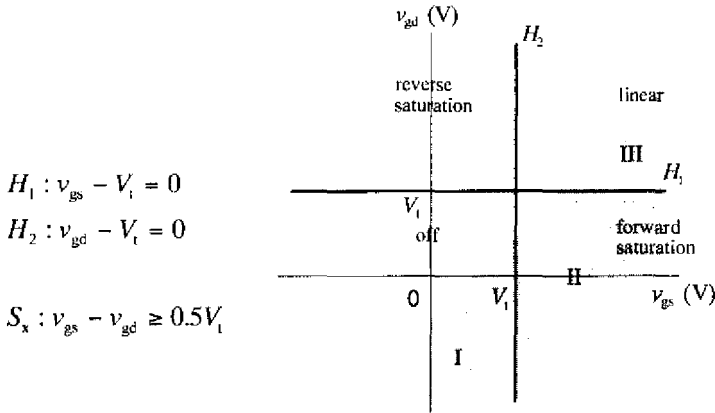


Figure 6.6. Intersection of the polytopes and the variables' domains, for the NMOST of Figure 6.2, defined by (6.4).

6.4.2 Application to Analog Testing

This section applies the principles of Section 6.4.1 to analog testing. First, the approach is briefly outlined.

An analog circuit can be characterized by the values of certain parameters. These may be component parameters (such as the threshold voltages of the transistors in an opamp) but also higher level parameters. It is possible to specify the nominal behavior of a circuit in terms of its parameters, if a sufficient set of parameters is available. The circuit may then be tested by determining, for example, whether the value of each parameter is inside a predefined interval.

The following approach uses preselected parameters to judge whether or not a circuit is at fault. To be able to apply the described combination of PL modeling and interval calculus, these parameters are first added to the PL models as elements of the input vector \mathbf{x} or the output vector \mathbf{y} . Also bounds on (or relations between) the elements of \mathbf{x} and \mathbf{y} are defined. The previously discussed combination of PL modeling and interval analysis computes the complete allowed region that defines proper circuit behavior.

After this region is computed, measurements are made. These measurements correspond with additional relations between some elements of \mathbf{x} and \mathbf{y} . Thus the measurements reduce the dimensionality of the allowed region. If, after sufficient measurements are made, the allowed region is empty then the circuit is at fault.

For example, consider as a parameter the threshold voltage V_t of an NMOST transistor. It is possible to add V_t as an extra input variable to the transistor's PL model (6.4), obtaining the model⁴

$$\mathbf{I} \begin{bmatrix} i_d \\ i_g \\ i_s \end{bmatrix} + \begin{bmatrix} -\alpha & 0 & \alpha & 0 \\ 0 & 0 & 0 & 0 \\ \alpha & 0 & -\alpha & 0 \end{bmatrix} \begin{bmatrix} v_d \\ v_g \\ v_s \\ v_t \end{bmatrix} + \begin{bmatrix} -\beta & \beta \\ 0 & 0 \\ \beta & -\beta \end{bmatrix} \mathbf{u} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \mathbf{0} \quad (6.19)$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_d \\ i_g \\ i_s \end{bmatrix} + \begin{bmatrix} 0 & -1 & 1 & 1 \\ 1 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_d \\ v_g \\ v_s \\ v_t \end{bmatrix} + \mathbf{I} \mathbf{u} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{j}$$

$$\mathbf{u}^\top \mathbf{j} = 0 \quad \forall_i u_i, j_i \geq 0$$

where α and β are again defined by (6.5). Model (6.19) offers exactly the same PL description as (6.4), but with v_t as a *variable*, instead of a constant. Because of this difference, the method of Section 6.4.1 can now take into account bounds that are defined for v_t . Even inequalities in terms of v_t and/or other variables (i.e., other parameters, or certain voltages or currents) may be taken into account. Such bounds and inequalities provide powerful criteria that define allowed circuit behavior. After the PL models are extended by inclusion of the parameters as elements of \mathbf{x} and \mathbf{y} , the discussed solution method for non linear equations with bounded parameters is applied. This will be explained by a simple example.

⁴ The lower case notation for the threshold voltages signifies that they are now variables and not constants.

Figure 6.7 depicts a CMOS inverter that was designed for a 2.4 μm process. The NMOS transistor (T_N) is modeled according to (6.19), and the PMOS transistor (T_P) is modeled similarly. The inverter is tested by determining $v_{i,p}$ and $v_{i,n}$: the threshold voltages of the PMOS and the NMOS transistor, respectively. For a specific application the inverter operates according to specifications if $v_{i,p}$ and $v_{i,n}$ are in the intervals

$$v_{i,n} \in [0.85, 0.95] \text{ V and } v_{i,p} \in [-0.95, -0.85] \text{ V.} \quad (6.20)$$

In addition, $v_i \in [0, 5] \text{ V}$ in the inverter's normal operating region.

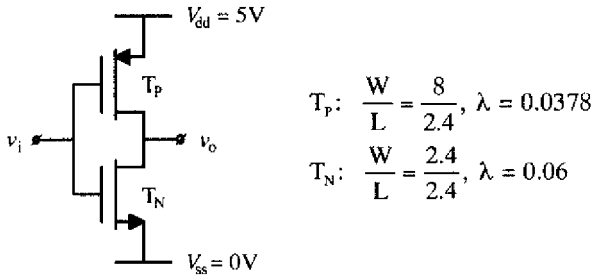


Figure 6.7. CMOS inverter.

It is possible to obtain a description of the inverter polytopes and linear mappings in the $(v_i, v_o, v_{i,n}, v_{i,p})$ space by using the technique outlined in [Spaa94]. This approach uses a hierarchical circuit description. The polytopes of the PL models at the lowest hierarchical level are translated to a higher level by combining PL models and using the topological relations (i.e., the model interconnections). This process continues until a behavioral description at the highest level is obtained. Each polytope at the highest level corresponds with a combinations of model polytopes at the lowest level. Only certain combinations occur, due to the circuit topology. For the inverter 5 combinations occur out of a possible number of $3 \times 3 = 9$ combinations. Each such combination corresponds with a single polytope in the $(v_i, v_o, v_{i,n}, v_{i,p})$ space.

After the inverter polytopes are determined at the highest level they are combined with the bounds on v_i , $v_{i,p}$ and $v_{i,n}$ to obtain \mathcal{K}_{dom} for the inverter. In this case, \mathcal{K}_{dom} is a collection of five convex regions. It defines the domain on which each high-level linear mapping is valid, analogous to step 1 of Section 6.4.1. Combining the linear mappings with these domains (conform step 2 of Section 6.4.2) results in a collection of five convex regions:

$$\begin{bmatrix} v_i \\ v_o \\ v_{i,n} \\ v_{i,p} \end{bmatrix} = \bigcup_{j=1}^{j=5} \mathbf{f}_j, \text{ where } \begin{cases} \mathbf{f}_j = \sum_{i=1}^{i=\ell} r_{ij} \mathbf{a}_{ji}, \sum_{i=1}^{i=\ell} r_{ij} = 1, r_{ij} \geq 0 \\ \ell = 2 \text{ for } j \in \{1, 5\} \\ \ell = 8 \text{ for } j \in \{2, 3, 4\} \end{cases} \quad (6.21)$$

The allowed circuit behavior is completely described by (6.21). Apparently, three of the five convex regions have 8 edges ($\ell = 8$) and two regions have 2 edges ($\ell = 2$). For example, for the third convex region ($j = 3$):

$$\begin{aligned} \mathbf{a}_{31} &= (2.48, 1.64, 0.85, -0.95)^T & \mathbf{a}_{32} &= (2.39, 3.35, 0.85, -0.95)^T \\ \mathbf{a}_{33} &= (2.53, 1.69, 0.95, -0.95)^T & \mathbf{a}_{34} &= (2.45, 3.38, 0.95, -0.95)^T \\ \mathbf{a}_{35} &= (2.54, 1.64, 0.85, -0.85)^T & \mathbf{a}_{36} &= (2.44, 3.36, 0.85, -0.85)^T \\ \mathbf{a}_{37} &= (2.58, 1.64, 0.95, -0.85)^T & \mathbf{a}_{38} &= (2.49, 3.35, 0.95, -0.85)^T \end{aligned} \quad (6.22)$$

To illustrate, Figure 6.8 plots the outlines of (6.21) in the (v_i, v_o) space. The plot was obtained by considering only the first two coordinates of each 4-dimensional vector \mathbf{a}_{ji} . For each of the 5 convex regions of (6.21) this produced a number of vectors in \mathbf{R}^2 . Only those vectors were preserved that could not be formed as a positive linear combination of the other vectors. The third region ($j = 3$) corresponds with the third plotted segment. Note that, according to Figure 6.8, the variations in the threshold voltages do not influence the transfer behavior in the inverter's "high" or "low" state. This is logical since in those cases one of the two transistors has a low drain source impedance, effectively shorting the output to V_{dd} or V_{ss} . For the corresponding regions in Figure 6.8, $\ell = 2$.

An input-output measurement $(v_i, v_o) = (a, b)$ adds the equalities $v_i = a$, $v_o = b$ to (6.21). A recomputation of the solution space results in a new, reduced solution space in the $(v_{i,p}, v_{i,n})$ space. At this stage there are several possibilities:

- The new solution space is empty. Apparently the solution space is on the wrong side of one or both of the two hyperplanes that correspond with the (v_i, v_o) measurement. Note that this implies that a *single* measurement may be enough to decide that the inverter is faulty, although its behavior is characterized by *two* parameters.
- A reduced, non empty solution space is calculated. In this case, more measurements must be made, until either the solution space is empty or a value can be computed for each tested parameter. In the latter case the tested circuit is functioning within specifications.

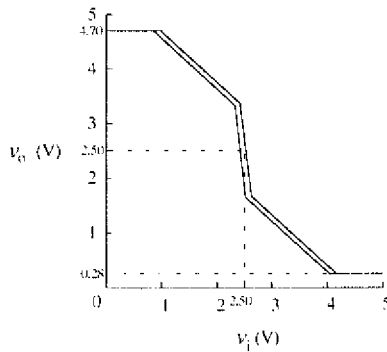


Figure 6.8. Allowed behavior in the (v_i, v_o) space, for the inverter of Figure 6.7.

For example, suppose that the measurement $(v_i, v_o) = (2.50, 2.50)$ V is made (depicted with dotted lines in Figure 6.8). In effect this measurement adds the equalities $v_i = 2.50$ and $v_o = 2.50$ to the description of the third convex region in (6.21). Recomputation of the convex region for $j = 3$ obtains the relation

$$9.3v_{1,n} + 9.5v_{1,p} = 0. \quad (6.23)$$

Thus after the first measurement the solution space is not empty, but the parameter values are not completely determined. Therefore another measurement is necessary. Now the measurement $(v_i, v_o) = (3.00, 1.25)$ V is made. This measurement obtains the relation

$$v_{i,p} = -0.92. \quad (6.24)$$

From (6.23) and (6.24) the threshold voltages are calculated as $v_{i,p} = -0.92$, $v_{i,n} = 0.94$. Thus, from the measurements it follows that the inverter is functioning within specifications.

Defining bounds on parameters to decide whether or not a circuit is functioning according to specifications is not new. However, the technique that has been presented in this section has an important advantage. Instead of just allowing parameter bounds, any linear relation (including inequalities) between the circuit inputs, outputs and parameters is useable. This was illustrated by the definition of S_x in Figure 6.6. This results in a more powerful criterion on which to accept or reject a specific circuit.

The issue of test point selection has not been addressed in this section, because it has not yet been fully researched. However some remarks can be made. Consider a single input-single output circuit. Then a maximum of one measurement per polytope should be made. The reason is that, because the PL model is linear in each polytope, one measurement obtains a single linear relation per polytope. Furthermore, measurements should be made in the polytopes where the influence of parameter deviations is highly measurable. In principle, the entries for the parameters in the \mathbf{A} matrix of the PL model are the parameter sensitivities. For example, in the first and fifth polytopes of Figure 6.8 the sensitivity to the threshold voltages is zero. This is indicated by a zero sensitivity to these parameters in the high-level PL model that describes the inverter operation. A zero sensitivity means that the influence of the corresponding parameter is zero, so it cannot be measured in the polytope in which the zero sensitivity occurs.

6.5 Discussion

An approach to analog testing has been discussed that uses a combination of PL modeling and interval analysis. First the complete circuit behavior is calculated as a collection \mathcal{N}_{tot} of convex spaces, taking into account user-specified linear relations between the circuit's inputs, outputs and parameters. These relations define the circuit's valid operating region. Then input-output measurements are made to reduce the dimensionality of \mathcal{N}_{tot} . If it is possible to compute a value for the circuit parameters from these measurements then the circuit is testable.

The advantage of the discussed technique is that the allowed circuit behavior may be specified with linear relations between the circuit's inputs, outputs and parameters. Thus more powerful criteria may be used than simple parameter bounds. A second advantage is that the method can cope with strongly non linear circuits, as demonstrated in the example with the CMOS inverter. Finally the method is inherently capable of macro-modelling, due to the use of PL models. This enables circuit testing at a high, behavioral level, in stead of a component level.

7

Examples

This section presents examples that demonstrate the applicability of the test methods of Chapters 4 and 5. Section 7.1 briefly discusses the practical implementation of the techniques discussed in Chapters 3 - 5. Then Section 7.2 examines a second-order bandpass filter. The testability of its components is analyzed, and its complete response is predicted from a few measurements. Section 7.3 offers a more complicated example, predicting the complete response of a 10 bits D/A converter from a few selected measurements. The examples compare the direct and iterative methods of test-point selection. The quality of a set of test points is judged with the D-optimality criterion (discussed in Section 5.1). It will be shown that the iterative method selects test points of higher quality, which may considerably reduce the influence of random measurement errors. Finally, Section 7.4 applies the SVD to a form of testing called black-box testing, where only the nominal circuit response is known.

7.1 Practical Implementation

Sections 7.2 and 7.3 present examples that model a physical circuit as a collection of lumped passive or active components. Due to the influence of random deviations of the manufacturing process, a physical circuit always shows a certain deviation from nominal. This deviation is assumed to be caused by deviations of component

parameters. Parameter deviations are simulated by introducing normally distributed errors in the parameters, with zero mean and standard deviation σ_{man} . The result is a deviating circuit.

The random errors may model a certain *maximum* parameter error in the following manner. Consider a normally distributed random variable X , with mean μ_X and standard deviation σ_X . Then $3 \cdot \sigma_X$ may be considered to be the maximum deviation of x from μ_X . This is motivated by the fact that, according to (4.20), $P(|x - \mu_X| \leq 3 \sigma_X) = 0.998$, where x is an observation of X . Thus $3 \cdot \sigma_{\text{man}}$ may be considered to be the maximum parameter error caused by the manufacturing process.

The examples determine parameter values and/or behavior of a deviating circuit from measurements at selected test points. To generate measurements, the response of a deviating circuit is simulated. Then randomly distributed measurement errors are introduced in this response, with zero mean and a standard deviation σ_{meas} (corresponding with a maximum measurement error of $3 \cdot \sigma_{\text{meas}}$). The resulting disturbed response at a selected test-point is used as a measurement.

Next a brief description is given of the practical implementation of the various techniques that were discussed in this thesis. Figure 7.1 provides a global outline of the test program, which has been implemented in the C language. The five numbered blocks are globally described as follows.

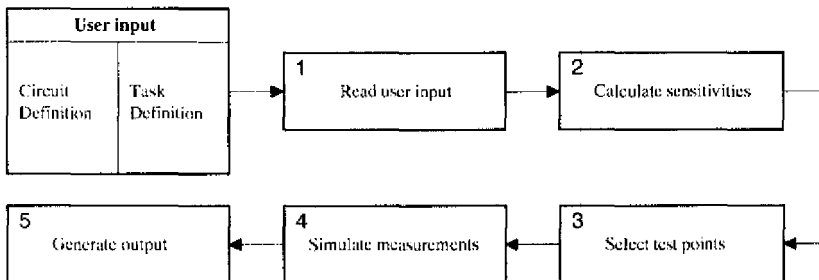


Figure 7.1. Program outline.

1. *Read user input.*

The user input is read from file into an internal data structure. According to Figure 7.1, the user input consists of two parts: a *circuit definition* and a *task definition*. In the circuit definition the circuit elements and topology of the circuit are specified in a SPICE-like manner. Most basic circuit elements are supported, including passive components, switches, and voltage and current controlled sources. With the task definition the user controls the various tasks of the program. For example, the user specifies the method of test-point selection (direct or iterative), the type of analysis (currently, DC and frequency analysis are supported), the circuit outputs that are accessible for measurements (e.g., nodal voltages), and the elements to which no sensitivities should be calculated (input sources, for example). Also some other aspects, like generation of deviating circuits and measurements are controlled through the task definition. The complete user input is parsed with the well-known lexical analyzer LEX and the grammatical analyzer YACC, and the various data are put in internal data and control structures.

2. *Calculate sensitivities.*

After the user input is read, sensitivities are calculated. As explained in Section 3.1.3.3, the adjoint method is used to generate sensitivities in the DC and frequency domain. The calculated sensitivities are put in a sensitivity matrix.

3. *Select test points.*

This part starts by computing the SVD of the sensitivity matrix. Then test points are selected with either the direct method of Section 4.3.2 or the iterative method of Section 5.2. If necessary, a group of testable parameters is selected, as explained in Section 4.3.4. Also the variances (4.16) and (4.19) may be computed in this step.

4. *Simulate measurements.*

This part simulates a deviating circuit by generating parameter deviations, on the basis of a user-specified value of σ_{man} . Also measurements are generated, on the

basis of a user-specified value of σ_{meas} . If necessary, the standard deviations of the parameters that result from random measurement errors may be extracted from repeated measurements at the selected test points. The measurements are used to predict the parameter values from (4.14) or the complete response from (4.17).

5. *Generate output.*

User-specified output is generated and written to a file. Examples are frequencies, outputs of the nominal or deviating circuit, error in the predicted output deviation, sensitivities to particular components and so forth. The output can be used by a program like gnuplot to quickly generate a large variety of plots. In this manner, all the plots in this thesis were produced.

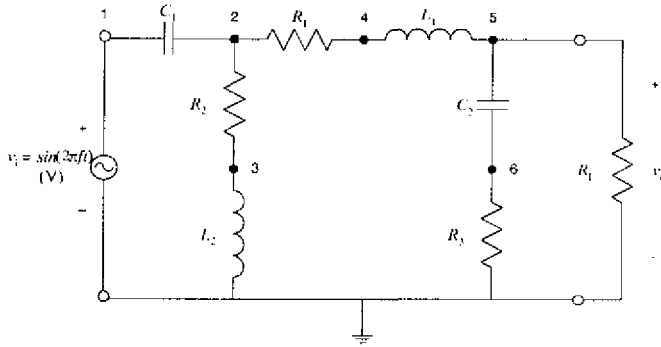
The result is a flexible program that allows a quick evaluation of a wide range of circuits. The size of the program is about 6000 lines, including function headers. The program is fully documented. It is implemented in ANSI C, so it may be compiled for on any platform on which LEXX and YACC are available.

7.2 2nd order Bandpass Filter

This section tests the 2nd order bandpass filter that is depicted in Figure 7.2. It is the mid-range section of a three way cross-over filter, designed by an optimization algorithm. The filter is designed to work with a load consisting of a loudspeaker and an associated impedance correction network. The resulting load is resistive to a good approximation; it is therefore represented by a loading resistor R_L . The amplitude of the sinusoidal input voltage is 1 V. Figure 7.3 depicts the nominal output voltage $|v_{o, \text{nom}}|$ as a function of frequency.

To generate a deviating circuit, maximum component deviations of 10 % are introduced. The resulting component deviations are listed in Table 7.1 B, column 2.

The filter's phase behavior is not considered, so only measurements of voltage amplitudes are made. The maximum measurement error is 1 mV.



$$\begin{array}{lll}
 R_1 = 0.15 \, \Omega & L_1 = 0.56 \, \text{mH} & C_1 = 16.8 \, \mu\text{F} & R_L = 5.8 \, \Omega \\
 R_2 = 1.85 \, \Omega & L_2 = 2.7 \, \text{mH} & C_2 = 1.2 \, \mu\text{F} & \\
 R_3 = 1.00 \, \Omega & & &
 \end{array}$$

Figure 7.2. 2nd order bandpass filter.

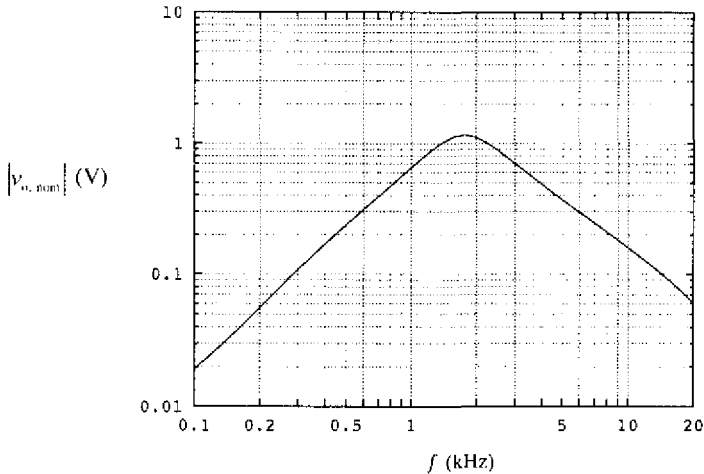


Figure 7.3. Nominal response of the bandpass filter of Figure 7.2.

First, only measurements of $|v_s|$ are allowed. A 47×7 sensitivity matrix \mathbf{S} was generated by evaluating sensitivities of $|v_s|$ at 20 frequencies per decade, equally spaced on a logarithmic scale from 100 Hz – 20 kHz. The direct approach of Section 4.3.2 selected the test points listed in Table 7.1 A. (The test points are listed in order

of selection.) Furthermore this approach identified the testable components, as explained in Section 4.3.4. The testability of component i is determined by computing its predicted standard deviation, denoted¹ by $\sigma_{\hat{p}_i}$. After test points are selected, this standard deviation is derived as the square root of the i^{th} diagonal element of (4.16). As explained in Section 4.2, a large value of $\sigma_{\hat{p}_i}$ signifies a low testability of the i^{th} component. For the 5 testable components, the value of $\sigma_{\hat{p}_i}$ is listed in Table 7.1 B, column 3. The number of testable components was determined with (4.35), using $\eta = 1$. For R_1 and R_3 , $\sigma_{\hat{p}_i} > \sigma_{\text{man}} = 3.3\%$ and therefore these components are considered untestable.

To check the accuracy of the predicted values of $\sigma_{\hat{p}_i}$, these were calculated from 50 repeated measurements of $|v_5|$ at the five selected frequencies, using (B.6). The results are listed in Table 7.1 B, column 4. Comparison with the predicted standard deviations shows that the predictions obtained from (4.16) are rather accurate for this example.

Table 7.1. Results of the testability analysis (node 5 accessible for measurements, direct test-point selection used).

A: Test Points		B: Results of Testability Analysis				
Node	Freq. (kHz)	Comp.	Deviation (%)	$\sigma_{\hat{p}_i}$ (%) (predicted)	$\sigma_{\hat{p}_i}$ (%) (from 50 meas.)	Error (%) (from 50 meas.)
5	0.56	L_1	1.62	0.08	0.09	0.02
5	1.26	C_1	-3.28	0.13	0.16	-0.12
5	1.78	L_2	1.75	0.27	0.26	0.45
5	3.16	C_2	-7.25	0.92	0.87	-0.15
5	14.13	R_2	-1.49	2.40	2.36	2.40
		R_1	3.20			
		R_3	-0.39			

¹ The notation of Section 4.3.3 is used.

The mean of the calculated component deviations was computed from the results of the 50 repeated measurements, using (B.5). This mean was used to calculate the mean error in each predicted component value, relative to the value of the deviating component. The mean errors are listed in the Table 7.1 B, column 5. For R_2 the mean error is unacceptably large: it is even larger than the deviation in that component. The reason for this large error will be investigated later.

Next, nodes 1-5 were considered accessible for measurements in an effort to increase component testability. To simplify interpretation of the results, the same component deviations were used. Again 20 test points per decade were computed in the frequency range 100 Hz - 20 kHz, resulting in a 235×7 sensitivity matrix. The selected test points are listed in Table 7.2 A. According to Table 7.2 B, column 3, all components are now testable. This is due to the increased accessibility for measurements. Comparison of columns 3 and 4 of Table 7.2 B shows that the predicted standard deviations are again close to the standard deviations that were extracted from 50 repeated measurements. Finally the mean errors in the predicted component deviations are listed in Table 7.2 B, column 5. Again there is a large mean error, this time in the predicted value of R_1 .

Table 7.2. Results of the testability analysis (nodes 2-6 accessible for measurements, direct test-point selection used).

A: Test Points		B: Results of Testability Analysis				
Node	Freq. (kHz)	Comp.	Deviation (%)	$\sigma_{\hat{\theta}_i}$ (%) (predicted)	$\sigma_{\hat{\theta}_i}$ (%) (from 50 meas.)	Error (%) (from 50 meas.)
3	1.41	L_1	1.62	0.08	0.08	0.04
4	0.631	C_1	-3.28	0.08	0.11	-0.24
5	1.26	L_2	1.75	0.32	0.32	0.44
5	2.00	C_2	-7.25	1.02	0.94	-0.16
6	2.24	R_2	-1.49	1.33	1.32	-0.43
6	12.59	R_1	3.20	2.15	2.24	1.78
7	2.24	R_3	-0.39	2.43	2.45	-0.03

Next the source of the large mean error in the predicted deviation of R_2 (Table 7.1 B, column 5) and R_1 (Table 7.2 B, column 5) is investigated. To determine if perhaps these errors are due to a low quality of the selected test points, 7 test points were selected with the iterative method of Section 5.2 (Table 7.3 A). The quality of these test points is determined by the determinant D (conform Definition 5.1) of the corresponding row and column reduced sensitivity matrix. For the test points of Table 7.3 A, $D = 1.08 \cdot 10^{-7}$. In contrast, for the test-points of Table 7.2 A, $D = 0.88 \cdot 10^{-7}$. Thus D increased by a factor of 1.23, due to iterative test-point selection. This change is not very large. This may be explained by the fact that the bandpass-filter is a relatively simple example, allowing the direct method to select relatively optimal test points. Thus the standard deviations of Table 7.2 B, columns 3 and 4 are very close to those displayed in Table 7.3 B, columns 3 and 4. Furthermore, from Table 7.3 B, column 5 it is apparent that the large error in R_1 is still present. Therefore it seems that this error is not caused by poorly chosen test points.

Table 7.3. Results of the testability analysis (nodes 2-6 accessible for measurements, iterative test-point selection used).

A: Test Points		B: Results of Testability Analysis				
Node	Freq. (kHz)	Comp.	Deviation (%)	$\sigma_{\hat{p}}$ (%) (predicted)	$\sigma_{\hat{p}}$ (%) (from 50 meas.)	Error (%) (from 50 meas.)
3	1.26	L_1	1.62	0.08	0.09	0.05
4	0.56	C_1	-3.28	0.07	0.11	-0.20
4	2.00	L_2	1.75	0.33	0.33	-0.06
5	1.59	C_2	-7.25	1.02	1.00	-0.35
6	2.24	R_2	-1.49	1.02	1.03	-0.85
6	12.59	R_1	3.20	1.90	1.76	1.80
7	2.24	R_3	-0.39	2.43	2.17	-0.26

Next, the test points of Table 7.3 A were again used for a testability analysis, but now the component deviations of Table 7.3 B, column 2 were decreased by 50%. The resulting component deviations are listed in Table 7.4, column 2. The measurement

errors were kept the same, so the standard deviations $\sigma_{\hat{p}_i}$ are given by Table 7.3 B, columns 3 and 4. The mean errors are listed in Table 7.4, column 3. Interestingly, the maximum mean error in the predicted component deviations was reduced from 1.80 % to 0.51 %: a reduction of 71 %. Thus the relative mean error decreases for decreasing component deviations. This can be explained by the fact that first-order sensitivities are used to model the influence of component deviations on the node voltages: The precision of these predictions that are obtained with these sensitivities decreases for increasing component deviations, as discussed in Section 3.1.

Table 7.4. Testability analysis of Table 7.3, but with a 50 % reduction in component deviations.

Comp.	Deviation (%)	Error (%) (from 50 meas.)
L_1	0.81	0.03
C_1	-1.64	-0.05
L_2	0.87	-0.05
C_2	-3.62	0.28
R_2	-0.74	0.03
R_1	1.60	0.51
R_3	-0.20	-0.14

Finally, Figure 7.4 shows the response deviation $\Delta|v_o| = |v_o| - |v_{o, \text{nom}}|$, predicted from the 5 test points of Table 7.1 A. Also the error in this predicted deviation and the corresponding standard deviation are plotted². The response deviation $\Delta|v_o|$ is caused by the deviating component values listed in Table 7.1 B, column 2. The standard deviation of the predicted $\Delta|v_o|$ at all candidate test points was obtained by taking the square root of the diagonal elements of (4.19)³. The maximum error in the predicted

² In contrast to expression (4.17), the plots in Figure 7.4 are not normalized to σ_{meas} .

³ Not taking into account the normalization mentioned in footnote 2.

$\Delta|v_o|$ that is displayed in Figure 7.4 is 0.38 mV. Furthermore it has been verified that the maximum *mean* error in the predicted $\Delta|v_o|$ is about 0.2 mV. Finally, the maximum standard deviation of the predicted $\Delta|v_o|$ is 0.33 mV.

Thus, the mean error in the predicted $\Delta|v_o|$ is about 50 % of its maximum standard deviation, which is equal to the standard deviation of the measurement errors. Therefore, for this example measurements at just 5 selected frequencies predict the response with about the same precision as when measurements are made at all 47 candidate test points (frequencies). This clearly demonstrates the benefits of careful test-point selection.

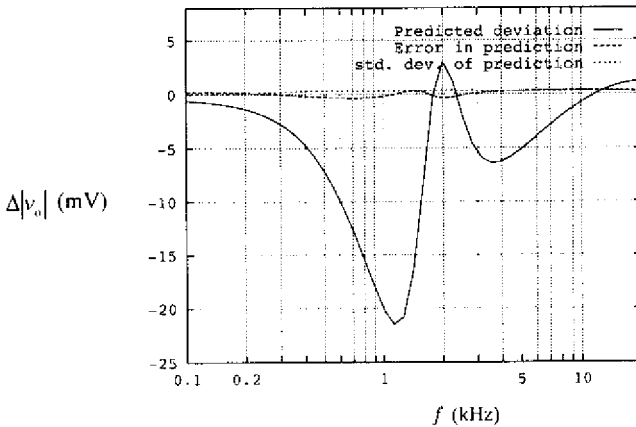


Figure 7.4. Predicted response deviation for the bandpass filter of Figure 7.2.

7.3 10 bit Digital-to-Analog Converter

This section presents a somewhat more complicated example, depicted in Figure 7.5. It is a 10 bits R - $2R$ Digital-to-Analog (D/A) converter, taken from [Sten87]. The circuit is based on the following simple principle: The current that flows from the voltage source into the ladder network is evenly divided at each R - $2R$ junction. Half the current flows through the $2R$ resistor and the other half flows through the R resistor to

the rest of the ladder network at the right of the junction. The current that flows through the $2R$ resistor in series with switch S_i is $\frac{V_{\text{ref}}}{2R} \cdot (\frac{1}{2})^{9-i}$. Consider a 10 bit code word $b_9 \cdots b_0$, where each bit b_i is either 0 or 1, and b_0 is the Least Significant Bit (LSB). Let switch S_i be closed if $b_i = 1$ and open if $b_i = 0$. Then the total current that arrives at current meter A is

$$i_{\text{out}} = \frac{V_{\text{ref}}}{2R} \cdot \sum_{i=0}^9 (\frac{1}{2})^{9-i} \cdot b_i. \quad (7.1)$$

The current i_{out} is converted into a voltage by an ideal opamp (not shown), so

$$V_{\text{out}} = V_{\text{ref}} \cdot \sum_{i=0}^9 (\frac{1}{2})^{9-i} \cdot b_i. \quad (7.2)$$

Thus each code word is transformed into a corresponding output voltage. The resistors denoted by “ r ” in Figure 7.5 model switch and wiring resistances⁴ that are due to the implementation of the converter in an integrated circuit [Sten87].

A deviating converter is simulated by adding to each resistor value a Gaussian error, with a maximum value of 3%. Due to the resulting deviations of the resistors the converter exhibits a certain linearity error. This error is called the converter’s response deviation: it is the difference in the response of a deviating and a nominal converter. The maximum measurement error is 0.2 LSB.

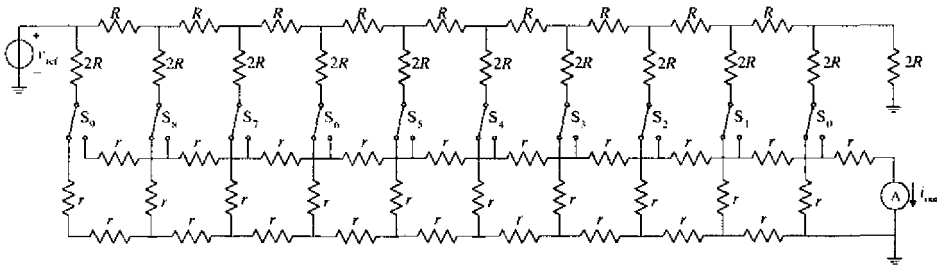
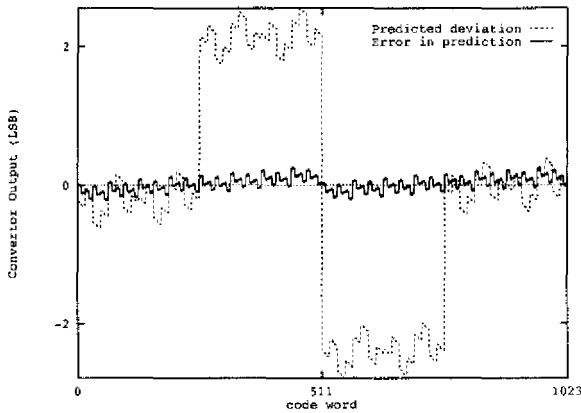


Figure 7.5. 10 bits R-2R D/A converter. $R = 100 \text{ k}\Omega$, $r = 10^{-3} \cdot R$.

⁴ The resistors “ r ” are short circuits, as far as the explanation of the operating principle is concerned.

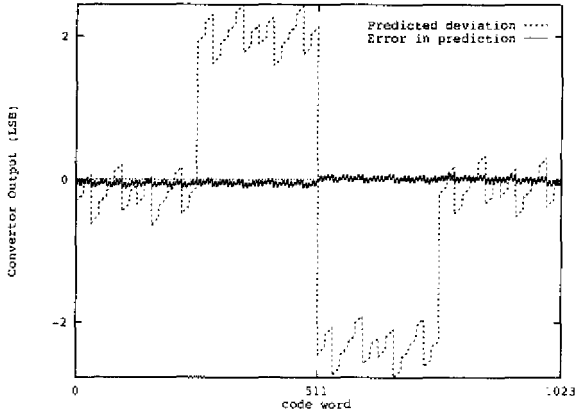
Sensitivities are generated for all 1024 possible code words, resulting in a 1024×49 sensitivity matrix. The direct approach of Section 4.3.2 selected the 7 test points listed in Figure 7.6, where the number of test points is determined using (4.35), with $\eta = 1.3$. The complete response deviation was predicted from 7 DC measurements of the output voltage at the selected code words. The result is depicted in Figure 7.6, which also shows the error in the prediction. The maximum error in the prediction is approximately equal to the measurement error. Thus a relatively high precision is obtained by measuring the response at a small subset of the 1024 code words.



Selected code words (in order of selection): 0, 511, 767, 895, 959, 991, 16

Figure 7.6. Predicted response deviation for the D/A converter, using the direct method for test-point selection.

Next, 7 test points were selected with the iterative method of Section 5.2. For the selected code words (given in Figure 7.7) $D = 1.67 \cdot 10^{-2}$. In contrast, for the code words selected with the direct method (shown in Figure 7.6) $D = 6.47 \cdot 10^{-4}$. Thus, the quality of the test points that were selected with the iterative method is considerably higher: D increased by a factor of 24.8. The quality of the new test points is confirmed by Figure 7.7. The error displayed there is clearly smaller than the error in Figure 7.6. Exact comparison showed that the maximum error decreased by a factor of 2.3, and the RMS error decreased by a factor of 2.1.



Selected code words (in arbitrary order): 112, 159, 303, 448, 591, 672, 784

Figure 7.7. Predicted response deviation for the D/A converter, using the iterative method for test-point selection.

To show that the displayed improvement does not depend upon the particular component values of the used deviating circuit, the standard deviation $\sigma_{\hat{y}}$ of the predicted response deviation was calculated from (4.19) for the sets of test points of Figures 7.6 and 7.7. Figure 7.8 displays the results. There is a large decrease in $\sigma_{\hat{y}}$, due to the higher quality of the test points that were selected with the iterative method. Thus the influence of measurement errors is reduced by a careful selection of test points with the iterative method.

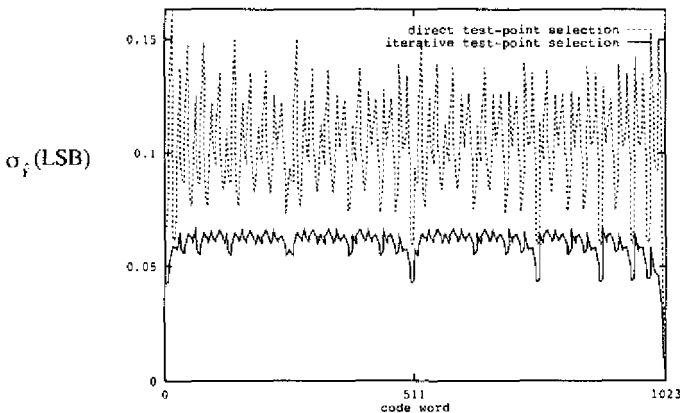


Figure 7.8. Comparison of $\sigma_{\hat{y}}$ for direct and iterative test-point selection.

To illustrate the progress of the iterative test-point selection process, Figure 7.9 plots the value of D against the number of iterations, for the iterative process that selected the test points of Figure 7.7. The value of D that is obtained with the direct method is indicated in Figure 7.9 by the dotted horizontal line.

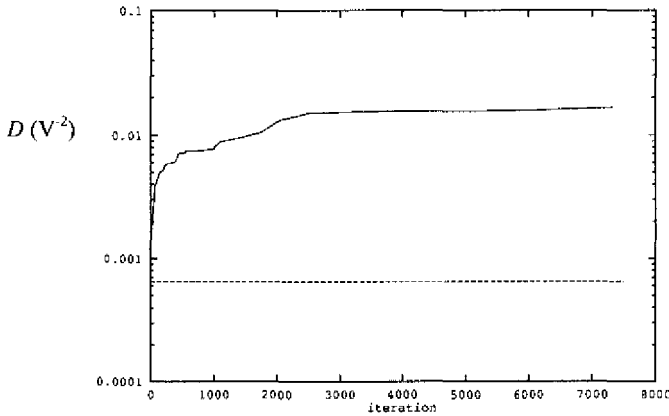


Figure 7.9. Optimization of D by the iterative test-point selection process.

According to Figure 7.9, the iterative algorithm quickly improves the quality of the test points in the first few iterations. In a later stage of the iterative refinement process, each iteration only slightly changes the current set of test points, leading to increasingly small improvements in D . It is noteworthy that already in the first stages of the iterative algorithm the value of D is higher than that obtained with direct test-point selection. Again this shows that the inherent greediness of the direct method may lead to a sub-optimal selection of test points.

The final example shows the applicability of the iterative method in the presence of large measurement errors. Again the D/A converter of Figure 7.5 is used, this time with other component deviations. The maximum component deviation is still 3%. The maximum measurement error was increased to 1 LSB, while the 7 test points of Figure 7.7 are used. Figure 7.10 shows the predicted output deviation for this situation. Due

to the large measurement errors, the maximum error in the predicted response deviation is now about 0.8 LSB.

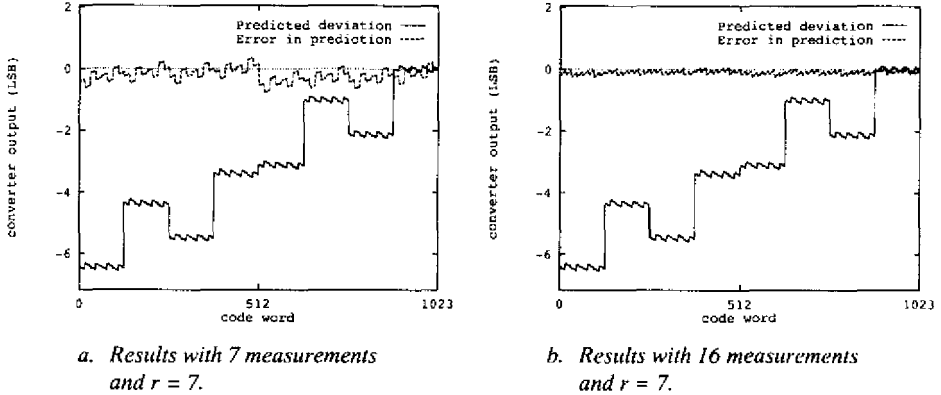


Figure 7.10. Reducing the influence of measurement errors by selecting more measurements than parameters.

To reduce the influence of the large measurement errors, the number of parameters and test points was iteratively adjusted, as explained in Section 5.3. This resulted in 16 test points, while the number of parameters remained unchanged. Figure 7.10 b shows that the higher number of test points significantly decreased the influence of the measurement errors: The maximum error in the predicted response deviation is reduced to 0.25 LSB. Note that this example also shows that the precision of the linear model is sufficient when only 7 model parameters are included.

7.4 Application of the SVD to Black box Testing

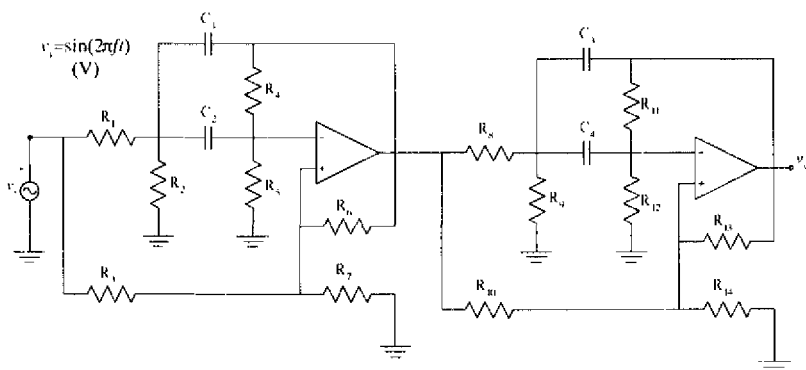
This section uses the SVD for “black box” testing, where a circuit is considered as a black box with input(s) and output(s). This example has been included to show that the SVD can also be used for testing in the case that a sensitivity matrix cannot be computed.

In black box testing, all that is known about the circuit is its nominal behavior. Because a circuit description is not available, a sensitivity matrix cannot be computed.

Therefore, to obtain a model of deviating circuit behavior, a *reference batch* of fabricated circuits is measured. The circuits in this batch show a representative spread of deviating behavior, caused by the production process.

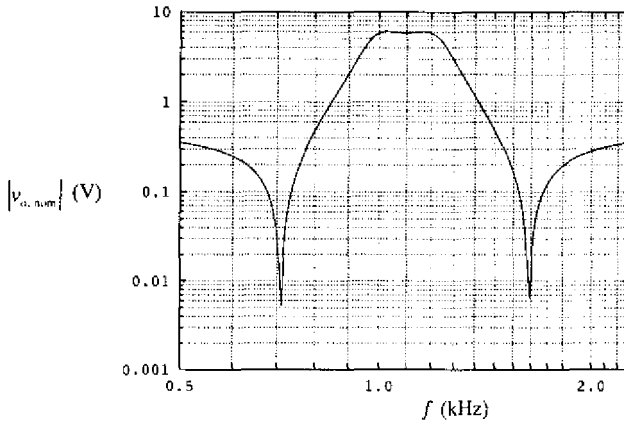
The procedure will be shown for the two-stage fourth-order bandpass filter shown in Figure 7.11. The circuit is excited with a sinusoidal input voltage with amplitude $|v_i| = 1$ V. Figure 7.12 depicts the nominal frequency response. The pass-band is from 1 to 1.2 kHz, and the stop bands are below 800 Hz and above 1.5 kHz. Deviating circuits are generated by disturbing the nominal component values with Gaussian distributed errors, with a standard deviation of 1 %. Measurement errors are modeled by Gaussian distributed errors with a standard deviation of 0.33 mV.

A reference batch of 25 deviating circuits was generated by introducing random component errors. For each of these circuits, $|v_o|$ was measured at 205 frequencies, evenly distributed on a logarithmic scale from 500 Hz to 2.4 kHz. The output deviation at one frequency is given by $\Delta|v_o| = |v_o| - |v_{o,nom}|$. The output deviations of the 25 circuits at all frequencies were collected as columns in a 205×25 matrix $\mathbf{A} = [\mathbf{a}_1 | \dots | \mathbf{a}_{25}]$, where \mathbf{a}_i denotes the deviation $\Delta|v_o|$ of the i^{th} circuit at all frequencies.



$R_1 = 26.967 \text{ k}\Omega$	$R_5 = 226.403 \text{ k}\Omega$	$R_9 = 92.380 \text{ k}\Omega$	$R_{13} = 100 \text{ k}\Omega$	$C_3 = 2.2 \text{ nF}$
$R_2 = 72.599 \text{ k}\Omega$	$R_6 = 100 \text{ k}\Omega$	$R_{10} = 15 \text{ k}\Omega$	$R_{14} = 60 \text{ k}\Omega$	$C_4 = 2.2 \text{ nF}$
$R_3 = 28.333 \text{ k}\Omega$	$R_7 = 42.5 \text{ k}\Omega$	$R_{11} = 213.682 \text{ k}\Omega$	$C_1 = 2.2 \text{ nF}$	
$R_4 = 158.487 \text{ k}\Omega$	$R_8 = 26.944 \text{ k}\Omega$	$R_{12} = 149.834 \text{ k}\Omega$	$C_2 = 2.2 \text{ nF}$	

Figure 7.11. Two-stage fourth-order active bandpass filter [Taken from Vlac94].



Selected frequencies in kHz (in order of selection):
 0.99, 1.19, 1.26, 1.04, 0.931, 0.712, 1.145, 1.681, 0.669, 1.346

Figure 7.12. Nominal filter response.

Next the test-point selection procedure of Section 4.3 was used to select a subset of 10 columns of \mathbf{A} . Figure 7.13 depicts the selected columns, denoted by $\mathbf{x}_1, \dots, \mathbf{x}_{10}$. The number of selected columns was determined on the basis of Criterion 5.9, with $\gamma = 0.98$. After column selection, the test-point selection procedure selected 10 rows of \mathbf{A} . These rows correspond with 10 frequencies (given in Figure 7.12) at which measurements will be made in the test phase. The selection of test frequencies concludes the pre-test phase.

In the test phase, the output deviation of a circuit under test was measured at the 10 selected frequencies. The measurements are denoted by a 10-dimensional vector \mathbf{c}_{red} . The measured response at all frequencies is denoted by \mathbf{c} .

In the post-test phase, the *parameter vector* $\mathbf{p} = (p_1, \dots, p_{10})^T$ was determined as the solution of the square system

$$\mathbf{X}_r \mathbf{p} = \mathbf{c}_r, \quad (7.3)$$

where \mathbf{X}_r is a 10×10 matrix that contains the selected rows of $\mathbf{X} = [\mathbf{x}_1 | \dots | \mathbf{x}_{10}]$.

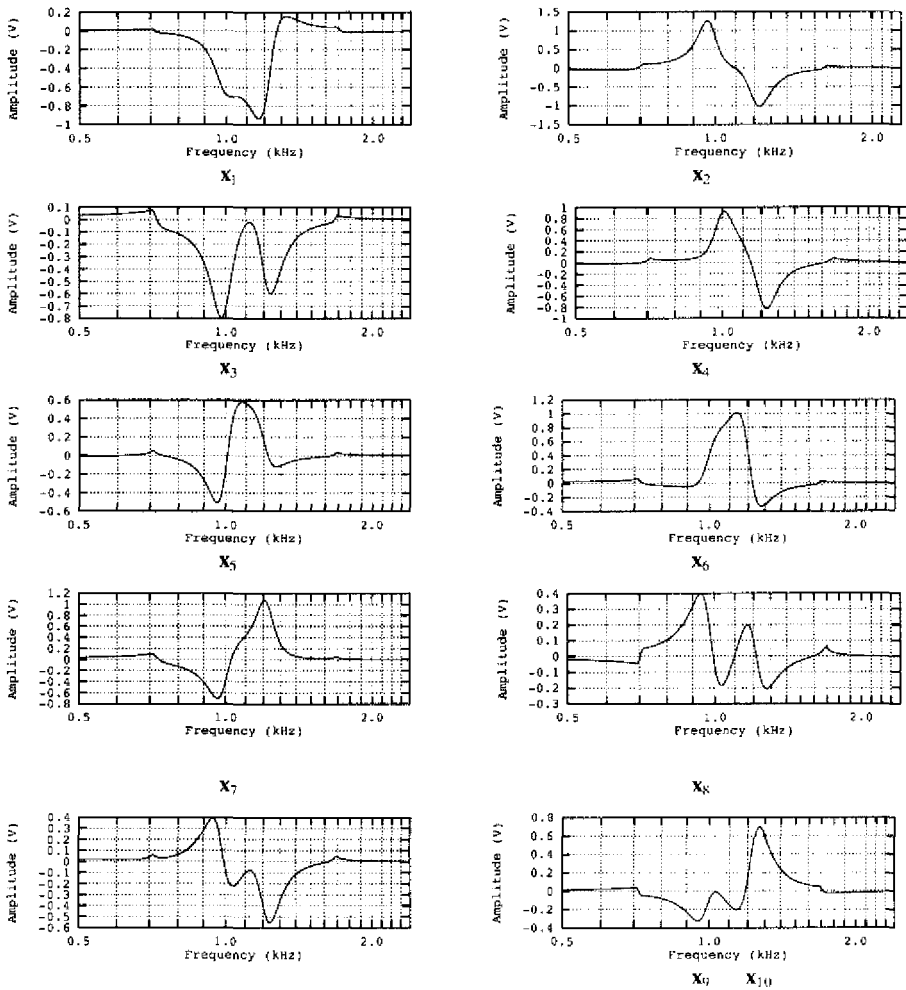


Figure 7.13. Graphical depiction of X_1, \dots, X_{10} .

The deviating output response c was approximated by the linear combination

$$\hat{c} = Xp, \quad (7.4)$$

Figure 7.14 plots the approximated output deviation (7.4), and the error $\hat{c} - c$ in the approximation. Evidently c is approximated by (7.4) with a high precision.

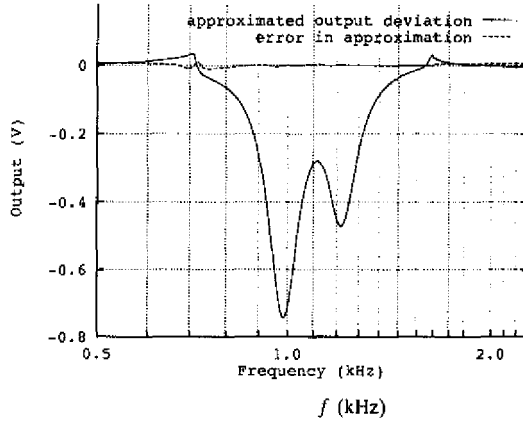


Figure 7.14. Predicted output deviation (7.4) and error in the prediction.

Next the standard deviation of the component errors was increased from 1 % to 3 %. This time a reference batch of 50 circuits was measured. The larger batch is necessary to account for the larger variety of deviating circuits, caused by the larger component deviations. On the basis of Criterion 5.9 (with $\gamma = 0.98$), 14 columns of \mathbf{A} are selected. In the test-phase, measurements at 14 selected frequencies were made. Figure 7.15 depicts the predicted output deviation and the error in the prediction. It shows that the relative maximum error increased. However, note that the maximum output deviation is large. It is even larger than the maximum nominal output voltage shown in Figure 7.12.

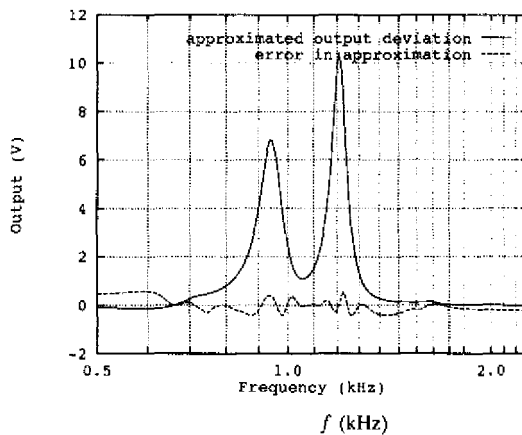


Figure 7.15. Predicted output deviation (7.4) and error in the prediction.

Thus the approach is capable of predicting *large* output deviations with reasonable accuracy. This accuracy is accomplished at the expense of a higher number of circuits in the reference batch. As discussed, the larger batch is necessary to account for the larger variety of deviating circuits (caused by the larger component deviations). Due to this larger variety, the rank estimate of \mathbf{A} (determined with Criterion 5.9) increases. This means that the number of *base functions* (examples of which are shown in Figure 7.13) increases. Thus a higher number of test points is selected to account for the larger output deviations.

As explained in previous examples, the predicted response deviation (7.4) may be used for a pass/fail test. For example, the maximum deviation in the pass-band, stop band and -3dB points may be checked against the specifications.

Summarizing, the described approach provides accurate response predictions from a relatively small number of measurements. It does not need a description of the circuit under test. Large output deviations are dealt with by increasing the number of measurements.

For larger output deviations, the number of circuits in the batch can become quite large. Therefore it takes a lot of time to measure the complete response deviation of these circuits. However, the cost of pass/fail testing is mainly determined by the on-line computational requirements. These are low because in the test phase the work consists mainly of solving the (small) system (7.3).

A problem of the described method is that it may be difficult to estimate the proper size of the batch. In addition, for large output deviations, the number of measurements might become impracticable high. These issues need to be further researched.

7.5 Discussion

Section 7.2 showed that a testability analysis may be performed with the direct method for test-point selection. For this example, the predicted standard deviations of the parameters are accurate. Thus these standard deviations may be used to assess the testability of circuit components, on the basis criterion (4.35). This criterion is useful to determine the number of testable parameters in the presence of random measurement errors. This was shown by accurately predicting the response deviation at 47 frequencies from measurements at just 5 frequencies. The precision of the prediction was shown to be equal to the precision that would be obtained from measuring the response at all 47 frequencies. This illustrates the benefits of careful test-point selection.

It was also shown that a relatively large mean error in a predicted component deviation was due to the use of first-order sensitivities. This shows that the use of a linearized circuit model results in errors when the component deviations are not infinitesimally small. Whether or not these errors are acceptable will in general depend on the application.

For the 10 bits D/A converter the direct method selects sub-optimal test points. This is mainly caused by the fact that the direct method selects test points one-by one, each time optimizing only the currently selected test-point. This was explained in Section 5.1. The iterative method selected test points of higher quality, resulting in a considerable reduction of the variance of the predicted response deviation. Finally, iterative refinement of the number of test points and parameters allowed for more precise predictions in the presence of large measurement errors. It was shown that the iterative method resulting in better test-points than the method presented in [Sten87].

Section 7.4 showed that the discussed principles of test-point selection and parameter selection may also be applied when a sensitivity matrix is not available. In this case a linear circuit model is obtained by measuring a batch of reference circuits. The

complete response deviation of a circuit under test is formed as a linear combination of the response deviations of selected circuits from the batch.

The examples show that the SVD may be applied to diagnostic testing as well as pass/fail production testing.

Next a suggested approach for black box testing in the time domain is discussed. The approach provides a direction for future research.

According to Sections 7.3 and 7.4, pass/fail testing can be accomplished by applying certain criteria on the predicted output deviation. However, also the computed parameter vector \mathbf{p} may be used for a pass/fail test in the following manner. Assume that, to characterize a reference batch of n circuits, the output deviation is measured at m time points. Here the output deviation is the deviation from nominal of the impulse response, at all time points. A circuit's impulse response is also used in some existing pass/fail methods, e.g., [Pan96] (discussed in Section 2.2). In [Pan96] it is explained that deviations of performance parameters in the DC domain and frequency domain also cause a deviation in the impulse response. Thus the impulse response can be used to implicitly measure deviations of various performance parameters in other domains, e.g. DC gain or bandwidth.

The results of the measurements of the reference batch are stored in a $m \times n$ matrix \mathbf{A} , where row i of \mathbf{A} corresponds to time t_i . After the SVD $\mathbf{A} = \mathbf{U}\mathbf{W}\mathbf{V}^T$ is computed, $\text{rank}(\mathbf{A})$ can be estimated with Criterion 5.9, as in Section 7.4. If $\text{rank}(\mathbf{A})$ is estimated as r , then

$$\mathbf{A} \approx \mathbf{U}_{:,r} \mathbf{W}_{rr} \mathbf{V}_{:,r},$$

where $\mathbf{U}_{:,r}$ is an $m \times r$ matrix containing the first r columns of \mathbf{U} , $\mathbf{V}_{:,r}$ is an $n \times r$ matrix containing the first r columns of \mathbf{V} , and \mathbf{W}_{rr} is an $r \times r$ diagonal matrix containing the first r singular values of \mathbf{A} . After the SVD of \mathbf{A} is computed, the pre-test phase is completed.

In the test phase, \mathbf{c} (the deviation from nominal of the impulse response) of a circuit under test is measured at m time points. Then a parameter vector \mathbf{p}_c is determined as the least-squares (LS) solution of

$$\mathbf{U}_r \mathbf{W}_r \boldsymbol{\pi} = \mathbf{c}. \quad (7.5)$$

The circuit under test is characterized by \mathbf{p}_c , in the sense that the response deviation \mathbf{c} is approximated by

$$\hat{\mathbf{c}} = \mathbf{U}_r \mathbf{W}_r \mathbf{p}_c. \quad (7.6)$$

If the approximation (7.6) is precise (so $\|\hat{\mathbf{c}} - \mathbf{c}\|_2$ is small) then \mathbf{p}_c characterizes the circuit under test. Like this the SVD is used to map the deviation at m time points to an r -dimensional parameter vector, where in general $r \ll m$.

The value of \mathbf{p}_c is used in the following way to decide whether or not the circuit under test is faulty (a comparable approach is used in [Pan96]). First, the response deviations $\mathbf{c}_1, \dots, \mathbf{c}_l$ of a large number l of deviating circuits are measured. Then $\mathbf{p}_1, \dots, \mathbf{p}_l$ are determined as the LS solutions of (7.5). In addition, each of the l circuits is characterized as good or faulty by making some measurements. As explained, the characterization may take place in other domains than the time domain. For example, an analog filter can be characterized by its passband gain, -3 dB points and stopband gains. After characterization, l points in parameter space are available, and for each point it is known whether it corresponds to a good or faulty circuit.

The l points define regions in parameter space, where each region contains (almost) only good or faulty circuits. Suppose that for a circuit under test \mathbf{p}_c is determined. Then the circuit is good if \mathbf{p}_c is in a region containing good circuits and vice versa. Exactly how to determine the regions in parameter space is the subject of future research.

Note that the outlined approach does not use test-point selection. While existing methods like [Pan96] and [Sten90] use test-point selection to determine a few time points at which the output deviation is measured, the above procedure uses the output deviation at *all* time points. The advantage is that all available information is used. The influence of noise and clock skew may be taken into account by zeroing the smallest

singular values of \mathbf{A} , as explained in Section 3.2. In addition, because approximation of matrices with the SVD is optimal, the number of elements in \mathbf{p}_c is minimal. This is important, since if \mathbf{p}_c has many elements then the dimensionality of the parameter space is high. In that case many circuits must be characterized to divide the space into regions corresponding with good and faulty circuits. Thus, the SVD is used to characterize a circuit by a minimal number of optimal parameters. The approach is intended for the time domain. In the frequency domain, measuring the complete response deviation of a circuit under test would take too much time, making the method impracticable.



8

Conclusions and Discussion

This chapter summarizes the results of the research that has been described in this thesis. Section 8.1 draws some conclusions. Section 8.2 summarizes the original contributions in this thesis.

8.1 Summary of Results and Discussion

The main part of the presented research focusses on test-point selection, testability analysis and response prediction for analog circuits. The presented methods use the SVD of a linearized circuit model (from here onwards called “the linear model”). The linear model provides a first-order approximation of non linear circuit behavior. It may be obtained by computing first-order (differential) sensitivities. Section 7.4 showed that a linear model may also be obtained by measuring the deviating response of a batch of circuits.

The main conclusion is that, within the limitations inherent to the approximation of a non linear system by a *first-order* model, the presented approach can be successfully applied to diagnostic testing as well as pass/fail production testing.

The SVD copes with possible rank deficiency of the linear model in an optimal manner. This is accomplished by transforming the sensitivity matrix to a new matrix of full rank. The new matrix retains a maximum amount of the information originally contained in the sensitivity matrix. Test points are selected using the new matrix.

A direct method for test-point selection selects independent measurements in the direction of highest sensitivity. These directions are found by computing the SVD of the sensitivity matrix. The result is an algorithmic approach that automatically selects mutually independent measurements.

In addition to the direct test-point selection algorithm, an iterative algorithm was presented. Its advantage is that in general it selects test points of higher quality (in the D -optimal sense) than the direct approach that is based on the SVD. The selected test points minimize the influence of random measurement errors, and maximize the amount of information that is obtained from measurements. The number of measurements and parameters of the linear model are carefully optimized, taking into account the influence of random measurement errors. Section 7.3 showed that the iterative algorithm selects test points of a higher quality, resulting in more precise predictions of a D/A converter's linearity error.

The discussed approach is preferable to an ad-hoc approach like (for example) [Boze94], where test point are selected on the basis of a visual inspection of plotted differential sensitivities. In [Boze94] it is not indicated how test points should be selected automatically. Of course, many approaches to automatic test-point selection are published. However, they lack a proper criterion that judges the optimality of a set of test points. For example, [Hemi90] uses the heuristic of removing the rows and column of the sensitivity matrix that are "nearly" dependent. The disadvantage is that it remains unclear in what sense the selected test points are optimal. Also the numerical properties of the presented approach remain unclear.

In contrast, the iterative method for test-point selection discussed in this thesis uses the well-established D -optimality criterion. This criterion minimizes the influence of measurement errors, and maximizes the independence of the measurements. It is shown with an example that this may lead to better test points than both the direct method of Section 7.2 and the approach published in [Sten90].

An important aspect of the discussed method is that it fully takes into account the influence of random measurement errors. It gives useful estimates of the resulting variance of the predicted parameter deviations and circuit output deviations. In addition, the iterative test-point selection algorithm allows the number of measurements to be increased to compensate for larger measurement errors.

The main limitation of the presented approach is the inability of a *linear* circuit model to accurately describe the influence of multiple *large* parameter deviation. This limits the applicability of the presented method to circuits that exhibit relatively small deviations from nominal behavior. However note that Chapter 7 demonstrated with the D/A converter that this may lead to precise predictions for a realistic example.

Finally, Section 7.4 presented an application of the SVD that might be able to predict the influence of larger parameter deviation by composing a linear model of higher rank, and selecting a higher number of measurements. However, the limitations of this approach should be further investigated.

8.2 Original contributions

This section presents a brief summary of the original contributions in this thesis.

- The Singular Value Decomposition (SVD) has not yet been applied to analog testing. The principles are published before as [Spaa95a], [Spaa95b]. This includes both the elimination of redundancy from the linear model (conform Section 4.3.1)

and the direct method for test-point selection (conform Section 4.3.2). Also determination of the necessary/sufficient number of measurements in a direct approach on the basis of the measurement errors has not been done before.

- The iterative method of test point selection in combination with the D -optimality criterion (and the SVD) is original and will be published in [Spaa96].
- The work on the application of PL techniques to analog testing was done in cooperation with D.M.W. Leenaerts. This work is based on the results presented in [Leen91]. It was published as [Leen92a] and [Leen92b].

9

References

- [Band85] J.W. Bandler and A.E. Salama, "Fault diagnosis of analog circuits," Proc. IEEE, Vol. 73, pp. 1279-1325, August 1985.
- [Bell95] I.M. Bell and S.J. Spinks, "Analogue fault simulation for the structural approach to analogue and mixed-signal IC testing," Intern. Mixed Signal Testing Workshop, pp. 10-14, 1995.
- [Bier81] R.M. Biernacki and J.W. Bandler, "Multiple-fault location of analog circuits," IEEE Trans. Circ. & Syst., Vol 28, pp. 361-367, May 1981.
- [Bokh86] W.M.G. van Bokhoven, "Piecewise linear analysis and simulation," in: *Circuit analysis, simulation and design*, ed. A.E. Ruehli, Amsterdam, North-Holland, 1986.
- [Bos91] A. van den Bos, "Measurement - the parametric approach," IEEE Conf. Instrum. and Meas., pp. 259-262, 1991.
- [Bowd71] H. Bowdler, R.S. Martin and J.H. Wilkinson, "The QR and QL algorithms for symmetric matrices," in J.H. Wilkinson and C. Reinsch, *Linear algebra*, Springer-Verlag, 1971.
- [Box71] M.J. Box and N.R. Draper, "Factorial designs, the $|X'X|$ criterion, and some related matters," Technometrics, Vol. 13, no. 4, November 1971.

- [Bray80] R.K. Brayton and R. Spence, *Sensitivity and optimization*, Elsevier, 1980.
- [Brow81] J.L. Brown Jr., E. Plotkin, L. Roytman and Z. Zayezdny, "Fault diagnosis in non-reciprocal networks," Proc. European Conf. Circuit Theory and Design, pp. 998-1002, 1981.
- [Chat96] A. Chatterjee, R. Jayabharathi, P. Pant and J.A. Abraham, "Non-robust tests for stuck-fault detection using signal waveform analysis: feasibility and advantages," proc. IEEE VLSI Test Symposium, pp. 354-359, 1996.
- [Chin90] Chin-Long Wey, "Built-in self-test (BIST) structure for analog circuit fault diagnosis," IEEE Trans. Instrum. Meas., Vol. 39, pp. 517-521, June 1990.
- [Chua77] L.O. Chua and S.M. Kang, "Section-wise piecewise linear functions: canonical representation, properties and applications," Proc. IEEE, Vol. 65, pp. 915-929, 1977.
- [Crem96] S. Cremous et al., "A new test pattern generation method for delay fault testing," proc. IEEE VLSI Test Symposium, pp. 296-301, 1996.
- [Dai90] H. Dai and T.M. Sounders, "Time domain testing strategies and fault diagnosis for analog systems," IEEE Trans. Instrum. and Meas., Vol. 39, no. 1, pp. 157-162, 1990.
- [Depr88] *SVD and signal processing, algorithms, applications and architectures*, edited by E.F. Deprettere, Elsevier science publishers B.V., North-Holland, 1988.
- [Dewi88] P. Dewilde and E.F. Deprettere, "Singular value decomposition, an introduction," in: *SVD and signal processing, algorithms, applications and architectures*, edited by E.F. Deprettere, Elsevier science publishers B.V., North-Holland, 1988.

- [Dire69] S.W. Director and R.A. Rohrer, "The generalized adjoint network and network sensitivities," *IEEE Trans. Circuit Theory*, Vol. 16, pp. 318-323, 1969.
- [Eind88] J.T.J. van Eijndhoven, "Piecewise linear analysis," in: *Analog Circuits: Computer Aided Analysis and Diagnosis*, ed. T. Ozawa, pp.65-92, Marcel Dekker inc. New York, March 1988.
- [Fedo72] V.V. Fedorov, *Theory of optimal experiments*, Academic press, New York, 1972.
- [Fid172] J.K. Fidler, "Differential-incremental sensitivity relationships," *Electr. Letters*, Vol. 8, no. 25, pp. 626-7, December 1972.
- [Fid175] J.K. Fidler, "Network sensitivity calculation," *IEEE Trans. Circ. & Syst.*, no. 9, pp. 567-71, 1976.
- [Golu71] G.H. Golub and C.F. van Loan, "Singular value decomposition and least squares solutions," in J.H. Wilkinson and C. Reinsch, *Linear algebra*, Springer-Verlag, 1971.
- [Golu89] G.H. Golub and C.F. van Loan, *Matrix computations*, second edition, The John Hopkins University Press, 1989.
- [Hemi90] G.J. Hemink, B.W. Meijer and H.G. Kerckhoff, "Testability analysis of analog systems," *IEEE Trans. CAD*, Vol. 9, pp. 573-583, June 1990.
- [Hoch79] W. Hochwald and J.D. Bastian, "A DC approach for analog fault determination," *IEEE Trans. Circ. & Syst.*, Vol 26, pp. 523-529, July 1979.
- [Host90] B.J. Hostica and W. Brockherde, "The art of analog circuits design in a digital VLSI world," *IEEE Intern. Symp. Circ. & Syst.*, pp. 1347-1350, 1990.
- [Huer93] J.L Huertas, "Test and design for testability of analog and mixed-signal integrated circuits: Theoretical Basis and Pragmatical Approaches," *European Conf. Circ. Theory and Design*, pp. 77-156, 1993.

- [Jenn92] A. Jennings and J.J. McKeown, *Matrix computation*, 2nd ed., Wiley, Chichester, 1992.
- [John79] A.T. Johnson, "Efficient fault analysis in linear analog circuits," *IEEE Trans. Circ. & Syst.*, Vol 26, pp. 475-484, July 1979.
- [Joll86] I.T. Jolliffe, *Principal component analysis*, Springer, Berlin, 1986.
- [Kahl90] C. Kahlert and L.O. Chua, "A generalized canonical piecewise linear representation," *IEEE Trans. Circ. & Syst.*, Vol. 37, no. 3, pp. 373-382, March 1990.
- [Keve91] T.A.M. Kevenaer and D.M.W. Leenaerts, "A flexible hierarchical piecewise linear simulator," *Integration, the VLSI journal*, Vol.12, pp. 211-235, 1991.
- [Keve92] T.A.M. Kevenaer and D.M.W. Leenaerts, "A comparison of piecewise linear model descriptions," *IEEE Trans. Circ. & Syst.*, Vol. 39, no. 12, pp. 996-1004, 1992.
- [Kirk83] S. Kirkpatrick, C.D. Gelatt, Jr., M.P. Vecchi, "Optimization by simulated annealing," *Science*, Vol. 220, 13 May 1983, pp. 671-680.
- [Leen90] D.M.W. Leenaerts, "Applications of interval analysis to circuit design," *IEEE Trans. Circ. & Syst.*, Vol. 37, no. 6, pp.803-807, 1990.
- [Leen91] D.M.W. Leenaerts, J.A. Hegt, "Finding all solutions of piecewise linear functions and applications to circuit theory," *Intern. J. Circuit Theory and Appl.*, Vol. 19, no. 2, pp. 107-123, 1991.
- [Leen93a] D.M.W. Leenaerts and J. van Spaandonk, "A Contribution to Testing of Analog Integrated Circuits in the DC Domain," *IEEE European Test Conference*, pp. 131-137, 1993.
- [Leen93b] D.M.W. Leenaerts and J. van Spaandonk, "DC testing of Analog Integrated Circuits with Piecewise Linear Approximation and Interval Analysis," *IEEE Intern. Symp. Circ. & Syst.*, pp. 1137-1340, 1993.

- [Mite74] T.J. Mitchell, "Computer construction of "D-optimal" first-order designs," *Technometrics*, Vol. 16, no. 2, May 1974.
- [Navi79] N. Navid and A.N. Willson, "A theory and algorithm for analog circuit fault diagnosis," *IEEE Trans. Circ. & Syst.*, Vol 26, pp. 440-457, July 1979.
- [Ogro94] J. Ogrodski, *Circuit simulation methods and algorithms*, CRC Press, Boca Raton, 1994.
- [Olbr95] T. Olbrich A. Richardson and A. Bratt, "Built-in self-test (BIST) for high-performance switched-current designs," 1995 Intern. Mixed Signal Testing Workshop, pp. 246-251, 1995.
- [Pan96] C.Y. Pan and K.T. Cheng, "Implicit functional testing for analog circuits," *proc. IEEE VLSI Test Symposium*, pp. 489-494, 1996.
- [Raja96] K.B. Rajan, D.E. Long and M. Abramovici, "Increasing testability by clock transformation (Getting rid of those darn states)," *proc. IEEE VLSI Test Symposium*, pp. 224-230, 1996.
- [Rans71] M.N. Ransom and R. Saeks, "Fault isolation with insufficient measurements," *IEEE Trans. Circ. Theory*, Vol 20, pp. 416-417, 1973.
- [Reno96] M. Renovell, F. Azais and Y. Bertrand, "A DFT technique for analog circuits," *proc. IEEE VLSI Test Symposium*, pp. 54-59, 1996.
- [Robi91] D.J.S. Robinson, *A course in linear algebra with applications*, World Scientific, Singapore, 1991.
- [Rute93] R.A. Rutenbar, "Analog design automation: Where are we? Where are we going?," *IEEE 1993 Custom Integr. Circ. Conf.*, pp. 13.1.1-13.1.8, 1993.
- [Saek72] R. Saeks, S.P. Singh and R. Liu, "Fault isolation via components simulation," *IEEE Trans. Circ. Theory*, Vol 19, pp. 634-640, November 1972.

- [Slam92] M. Slamani and B. Kaminska, "Analog circuit fault diagnosis based on sensitivity computation and functional testing," IEEE Design and Test of Computers, pp. 30-39, 1992.
- [Slam94] M. Slamani, B. Kaminska and G. Quesnel, "An integrated approach for analog circuits testing with a minimum number of detected parameters," IEEE Intern. Test Conf., pp. 631-640, 1994.
- [Solo80] A.S. Solodovnikov, *Systems of linear inequalities*, translated by L.M. Glasser and T.P. Branson, The University of Chicago Press, Chicago, 1980.
- [Soma91] M. Soma, "An experimental approach to analog fault models," IEEE 1991 Custom Integr. Circ. Conf., pp. 13.6.1-13.6.4, 1991.
- [Spaa94] J. van Spaandonk, "Towards a pass/fail test method for analog circuits," 1994 IEEE PRORISC workshop, pp. 225-229, 1994.
- [Spaa95a] J. van Spaandonk, "Using the singular value decomposition to test analog circuits with a sensitivity based approach", IEEE PRORISC Workshop, Mierlo, The Netherlands, pp. 295-302, 1995.
- [Spaa95b] J. van Spaandonk, "Application of the singular value decomposition to the testing of analog circuits," 1995 Intern. Mixed Signal Testing Workshop, pp. 159-164, 1995.
- [Spaa96] J. van Spaandonk and T.A.M. Kevenaer, "Selecting measurements to test the functional behavior of analog circuits", accepted for publication in: Journal of Electronic Testing: Theory and Applications, 1996.
- [Sten87] G.N. Stenbakken and T.M. Souders, "Test-point selection and testability measures via QR factorization of linear models," IEEE Trans. Instrum. Meas., Vol. 36, pp. 406-410, June 1987.
- [Sten89] G.N. Stenbakken, T.M. Souders and G.W. Stewart, "Ambiguity groups and testability," IEEE Trans. Instrum. Meas., Vol. 38, pp. 941-947, October 1989.

- [Swar87] F.A. Swartz, "Network sensitivity and tolerance analysis," in: *Computer-Aided design of microelectronic circuits and systems, Vol. 1: General Introduction and Analog circuit aspects*, chapter 5, 1987.
- [Tsch71] S.N. Tschernikow, *Lineare Ungleichungen*, Berlin: VEB Deutscher Verlag der Wissenschaft, 1971 (in German).
- [Vasq96] D. Vázquez, J.L. Huertas and A. Rueda, "Reducing the impact of DFT on the performance of analog integrated circuits: improved SW-OPAMP design," *proc. IEEE VLSI Test Symposium*, pp. 42-47, 1996.
- [Vlac94] J. Vlach and K. Singhal, *Computer methods for circuit analysis and design*, second edition, Van Nostrand Reinholdt, New York, 1994.
- [Walp93] R.E. Walpole and R.H. Myers, *Probability and statistics for engineering and scientists*, fifth edition, Macmillan Publishing Company, New York, 1993.
- [Wilk71] J.H. Wilkinson and C. Reinsch, *Linear algebra*, Springer-Verlag, 1971.
- [Wohl96] P. Wohl, J. Waicukauski and M. Graf, "Testing 'untestable' faults in three-state circuits," *proc. IEEE VLSI Test Symposium*, pp. 324-331, 1996.



Appendix **A**

Efficient Calculation of the Singular Value Decomposition

This appendix discusses the principles of an algorithm (due to Golub and Reinsch) that computes the Singular Value Decomposition of a matrix \mathbf{A} . Two stages can be distinguished in the algorithm. First, \mathbf{A} is transformed to bidiagonal form by two sequences of Householder transformations, as discussed in Section A.2. The second part of the SVD algorithm diagonalizes the bidiagonal form to obtain the SVD, as discussed in Section A.3. A brief discussion about the efficiency of this algorithm is found in Section A.4. First, Section A.1 provides an introduction to the computation of the SVD.

A.1. Introduction

According to Section 3.2.2, the SVD of an $m \times n$ matrix \mathbf{A} can in principle be found by calculating the eigenvalues and eigenvectors of $\mathbf{A}^* \mathbf{A}$, and the eigenvectors of $\mathbf{A} \mathbf{A}^*$ (the eigenvalues of $\mathbf{A} \mathbf{A}^*$ need not be computed separately since they are identical to the eigenvalues of $\mathbf{A}^* \mathbf{A}$). However, this is not the preferred approach because forming the products $\mathbf{A} \mathbf{A}^*$ and $\mathbf{A}^* \mathbf{A}$ may result in an unnecessary loss of numerical precision.

To illustrate, compare $c(\mathbf{A})$ and $c(\mathbf{A}^* \mathbf{A})$ (the condition numbers of \mathbf{A} and $\mathbf{A}^* \mathbf{A}$, respectively). According to Section 3.2.2, $c(\mathbf{A}) = w_1 / w_k$, where w_1 and w_k are the largest and smallest singular value of \mathbf{A} , respectively. Furthermore it is easy to show that $c(\mathbf{A}^* \mathbf{A}) = w_1^2 / w_k^2 = \{c(\mathbf{A})\}^2$. This increase in condition number corresponds with a numerical inaccuracy.

To illustrate, consider the real matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ \beta & 0 \\ 0 & \beta \end{bmatrix},$$

so

$$\mathbf{A}^T \mathbf{A} = \begin{bmatrix} 1 + \beta^2 & 1 \\ 1 & 1 + \beta^2 \end{bmatrix}.$$

Exactly computing the eigenvalues of $\mathbf{A}^T \mathbf{A}$ and taking their square root gives the singular values of \mathbf{A} :

$$\sigma_1 = (2 + \beta^2)^{1/2}, \quad \sigma_2 = |\beta|.$$

If $\beta^2 < \varepsilon_0$, the machine precision, then $\mathbf{A}^T \mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ and diagonalization of this matrix gives $\sigma_1 = \sqrt{2}$, $\sigma_2 = 0$. Thus it would be better to use a method that finds the eigenvalues and eigenvectors of $\mathbf{A}^* \mathbf{A}$ without explicitly forming that product. By the same argument, it is not advisable to form the normal equations (cf. Appendix C) to find the LS solution of $\mathbf{A}\mathbf{x} = \mathbf{b}$. Instead the SVD of \mathbf{A} should be used, as explained in Section 3.2.2.

A.2. Bidiagonalization

This section shows how a complex $m \times n$ matrix \mathbf{A} is reduced to bidiagonal form by the successive application of unitary transformations [Gol89]. The result is

$$\mathbf{P}^{(n)} \dots \mathbf{P}^{(1)} \mathbf{A} \mathbf{Q}^{(1)} \dots \mathbf{Q}^{(n)} = \left[\begin{array}{cccc} q_1 & e_2 & & \\ & q_2 & e_3 & \mathbf{0} \\ & & \ddots & \ddots \\ \mathbf{0} & & & \ddots & e_n \\ \hline & & & & q_n \\ & & & & \mathbf{0} \end{array} \right] = \mathbf{J}, \quad (\text{A.1})$$

where $\mathbf{P}^{(i)}$ and $\mathbf{Q}^{(i)}$ are unitary matrices (so $c(\mathbf{P}^{(i)}) = c(\mathbf{Q}^{(i)}) = 1$) for $i = 1, \dots, n$. Thus $c(\mathbf{J}^{(0)}) = c(\mathbf{A})$, and no precision is lost when \mathbf{J} is computed from \mathbf{A} as in (A.1). Because unitary matrices have condition number 1, they are widely used in matrix decompositions, for example to compute the QR decomposition of a matrix [Wilk71].

A unitary transformation does not affect the geometry of space. This was illustrated in Section 3.2.1, where Figure 3.6 shows that the orthogonal transformation that corresponds with \mathbf{V} does not warp the data set of Figure 3.4. Instead it moves the set as a whole, in this case by rotating and mirroring.

The advantage of the bidiagonal form (A.1) is that it can be diagonalized in a stable and efficient manner, as will be discussed in Section A.3. To obtain a bidiagonal form (A.1), Householder transformations are used. A Householder transformation is a rank 1 modification of the unity matrix \mathbf{I} , according to

$$\mathbf{P} = \mathbf{I} - 2\mathbf{v}\mathbf{v}^* / \mathbf{v}^*\mathbf{v}, \quad (\text{A.2})$$

where \mathbf{v} is a complex vector, called the Householder vector, with hermitian \mathbf{v}^* . It is easy to show that \mathbf{P} is hermitian and unitary. Geometrically, when a vector \mathbf{x} is multiplied by \mathbf{P} it is reflected in the hyperplane with normal \mathbf{v} . Therefore (A.2) is called a Householder reflection. It can be used to zero selected components of a vector in the following manner. If the Householder vector is defined as

$$\mathbf{v} = \mathbf{x} \pm \|\mathbf{x}\|_2 \mathbf{e}_1, \quad (\text{A.3})$$

then

$$\mathbf{P}\mathbf{x} = \left(\mathbf{I} - 2 \frac{\mathbf{v}\mathbf{v}'}{\mathbf{v}'\mathbf{v}} \right) \mathbf{x} = \mp \|\mathbf{x}\|_2 \mathbf{e}_1.$$

Thus the Householder transformation zeros all but the first component of \mathbf{x} . In principle, the choice of the sign in (A.3) is arbitrary. However, if \mathbf{x} is close to a multiple of \mathbf{e}_1 then \mathbf{x} and $\|\mathbf{x}\|_2 \mathbf{e}_1$ have about the same norm, and a loss of precision might occur in (A.3). This is avoided by letting

$$\mathbf{v} = \mathbf{x} + \text{sign}(x_1) \|\mathbf{x}\|_2 \mathbf{e}_1,$$

where $\text{sign}(x_1) = 1$ if $x_1 \geq 0$, and $\text{sign}(x_1) = -1$ if $x_1 < 0$.

The $m \times n$ matrix \mathbf{A} is bidiagonalized with an iterative process. In the i^{th} iteration, $\mathbf{P}^{(i)}$ is constructed to zero elements $i+1, \dots, m$ of column i . After the premultiplication (according to (A.1)) with $\mathbf{P}^{(i)}$, $\mathbf{Q}^{(i)}$ is constructed to zero elements $i+2, \dots, n$ of row i . For example, for a 5×5 matrix, the second iteration can be visualized as follows.

$$\begin{bmatrix} \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \xrightarrow{\mathbf{P}^{(2)}} \begin{bmatrix} \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{bmatrix} \xrightarrow{\mathbf{Q}^{(2)}} \begin{bmatrix} \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{bmatrix}$$

Because transformations $\mathbf{P} \triangleq \mathbf{P}^{(n)} \dots \mathbf{P}^{(1)}$ and $\mathbf{Q} \triangleq \mathbf{Q}^{(1)} \dots \mathbf{Q}^{(n)}$ are unitary, the singular values of \mathbf{A} are identical to the singular values of \mathbf{J} . Thus, if the SVD of \mathbf{J} is

$$\mathbf{J} = \mathbf{G}\mathbf{W}\mathbf{H}^T$$

then the SVD of \mathbf{A} is

$$\mathbf{A} = \mathbf{P}\mathbf{G}\mathbf{W}\mathbf{H}^T\mathbf{Q}^T \triangleq \mathbf{U}\mathbf{W}\mathbf{V}^T,$$

with $\mathbf{U} = \mathbf{P}\mathbf{G}$ and $\mathbf{V} = \mathbf{Q}\mathbf{H}$.

The discussed procedure reduces the problem of determining the SVD of \mathbf{A} to the problem of finding the SVD of a bidiagonal matrix (A.1).

A.3 Diagonalization of the Bidiagonal Form

The procedure that finds the SVD of \mathbf{J} by diagonalization of this bidiagonal matrix is described in [Gol71], and is not discussed in detail here.

According to (A.1), \mathbf{J} is a bidiagonal matrix, so $\mathbf{J}^* \mathbf{J}$ is a tridiagonal matrix. It is well-known that the eigenvalues of a tridiagonal matrix may be obtained by a succession of QR transformations (with origin shifts to improve convergence) [Bow71]. However, as discussed in Section A.1, explicit formation of $\mathbf{J}^* \mathbf{J}$ in general results in a loss of numerical precision. Therefore the SVD of \mathbf{J} is found by implicitly computing the eigenvalues of $\mathbf{J}^* \mathbf{J}$, without forming this matrix product.

The basic idea is to form in iteration i the product

$$\mathbf{J}^{(i+1)} = \mathbf{S}^{(i)\top} \mathbf{J}^{(i)} \mathbf{T}^{(i)}.$$

The matrix $\mathbf{T}^{(i)}$ is chosen such that the product $\mathbf{J}^{(i)\top} \mathbf{J}^{(i)}$ converges to a diagonal matrix. The matrix $\mathbf{S}^{(i)}$ is chosen such that $\mathbf{J}^{(i+1)}$ is again bidiagonal. Here $\mathbf{S}^{(i)}$ and $\mathbf{T}^{(i)}$ are both unitary matrices corresponding to Givens rotations. The successive application of matrices iteratively annihilates the off-diagonal elements of \mathbf{J} , so

$$\mathbf{J} \stackrel{\Delta}{=} \mathbf{J}^{(0)} \rightarrow \mathbf{J}^{(1)} \rightarrow \dots \rightarrow \mathbf{W},$$

where \mathbf{W} contains the singular values of \mathbf{J} and \mathbf{A} . Convergence is obtained after k iterations, if the off-diagonal elements of $\mathbf{J}^{(k)}$ are below a certain threshold.

A.4 Discussion

The total complexity of the discussed algorithm is of the order $4m^2n + 8mn^2 + 9n^3$, for an $m \times n$ matrix. Because of the contribution of the second term of this sum, the Golub-Reinsch algorithm is not particularly efficient when applied to a matrix for which $m > n$. Note that this is often the case for a sensitivity matrix, because the number of frequency points, time steps, or DC points (i.e., the number of rows) is generally much higher than the number of parameters (i.e., the number of columns).

If $m > n$ a computationally more efficient method, the R-SVD algorithm, can be used. This method, due to Chan and described in [Golu89], only differs from the discussed Golub-Reinsch algorithm by an extra preparatory step that transforms \mathbf{A} into an upper triangular matrix \mathbf{R} . Next, \mathbf{R} is transformed into bidiagonal form with Householder transformations, in the manner described in Section A.2. The complete procedure is called *R*-bidiagonalization. After bidiagonalization, the procedure described in Section A.3 is applied. The total efficiency of the R-SVD algorithm is of the order $4m^2n + 22n^3$, which is more efficient than the Golub-Reinsch algorithm if $m > n$.

In test-point generation, computation of the SVD does not constitute the major part of the total work. Computing parameter sensitivities and iteratively selecting test points are often more expensive. Therefore, since the Golub-Reinsch algorithm was available, this algorithm is used to calculate the SVD even though it is less efficient than the R-SVD algorithm.

Appendix **B**

Some results from statistics

This appendix presents some results from statistics. First an expression is derived for the variance of a linear combination of random variables. Then some common formulas are given that estimate the mean and variance of a random variable from a limited number of observations.

Let X be a discrete random variable that may attain one of n values x_1, \dots, x_n . $P(X = x_i)$ denotes the probability that the value of X is x_i . Now

$$E[X] = \sum_{i=1}^n \{x_i \cdot P(X = x_i)\} \quad (\text{B.1})$$

is the expected value of X . Furthermore, the expected value of a *continuous* statistical variable X is

$$E[X] = \int_{-\infty}^{\infty} x_i p(X) dx, \quad (\text{B.2})$$

where $p(X)$ is the probability density function of X .

Let X_i be a random variable. If X_i is discrete then $E[X_i]$ is given by (B.1), else $E[X_i]$ is given by (B.2). The rest of this appendix uses the following definitions.

- $\mu_{X_i} \triangleq E[X_i]$ is the *mean* of X_i .
- $\sigma_{X_i}^2 \triangleq E[(X_i - \mu_{X_i})^2]$ is the *variance* of X_i . Sometimes the notation σ_{X_i, X_i} is used for convenience.
- $\sigma_{X_i, X_j} \triangleq E[(X_i - \mu_{X_i})(X_j - \mu_{X_j})]$ is the *covariance* of X_i and X_j .
- X denotes the n -vector $(X_1, \dots, X_n)^T$ and Y denotes the m -vector $(Y_1, \dots, Y_m)^T$.

Consider the linear combination $Y = a_1 X_1 + \dots + a_n X_n$ of the n random variables X_1, \dots, X_n . The variance σ_Y^2 is

$$\begin{aligned}
 \sigma_Y^2 &= E[(a_1 X_1 + \dots + a_n X_n - \mu_{a_1 X_1 + \dots + a_n X_n})^2] \\
 &= E[(a_1 X_1 + \dots + a_n X_n - a_1 \mu_{X_1} - \dots - a_n \mu_{X_n})^2] \\
 &= E[(a_1 (X_1 - \mu_{X_1}) + \dots + a_n (X_n - \mu_{X_n}))^2] \\
 &= E[a_1 (X_1 - \mu_{X_1}) a_1 (X_1 - \mu_{X_1}) + \dots + a_1 (X_1 - \mu_{X_1}) a_n (X_n - \mu_{X_n}) \\
 &\quad + \dots \\
 &\quad + a_n (X_n - \mu_{X_n}) a_1 (X_1 - \mu_{X_1}) + \dots + a_n (X_n - \mu_{X_n}) a_n (X_n - \mu_{X_n})] \\
 &= a_1 a_1 E[(X_1 - \mu_{X_1})(X_1 - \mu_{X_1})] + \dots + a_1 a_n E[(X_1 - \mu_{X_1})(X_n - \mu_{X_n})] \\
 &\quad + \dots \\
 &\quad + a_n a_1 E[(X_n - \mu_{X_n})(X_1 - \mu_{X_1})] + \dots + a_n a_n E[(X_n - \mu_{X_n})(X_n - \mu_{X_n})] \\
 &= a_1 a_1 \sigma_{X_1, X_1} + \dots + a_1 a_n \sigma_{X_1, X_n} \\
 &\quad + \dots \\
 &\quad + a_n a_1 \sigma_{X_n, X_1} + \dots + a_n a_n \sigma_{X_n, X_n}
 \end{aligned}$$

Thus

$$\begin{aligned}\sigma_Y^2 &= (a_1, \dots, a_n) \cdot \begin{bmatrix} \sigma_{X_1, X_1} & \cdots & \sigma_{X_1, X_n} \\ \vdots & & \vdots \\ \sigma_{X_n, X_1} & \cdots & \sigma_{X_n, X_n} \end{bmatrix} \cdot \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \\ &= \mathbf{a}^\top \mathbf{C}_X \mathbf{a}\end{aligned}\tag{B.3}$$

where $\mathbf{a} = (a_1, \dots, a_n)^\top$ and \mathbf{C}_X denotes the $n \times n$ covariance matrix of X_1, \dots, X_n .

Now let Y_1, \dots, Y_m be m linear combinations of X_1, \dots, X_n , where the i^{th} combination is given by $Y_i = \mathbf{a}_i^\top X$. The row vectors $\mathbf{a}_1^\top, \dots, \mathbf{a}_m^\top$ may be stacked to form an $m \times n$ matrix \mathbf{A} . Thus, with (B.3) the following result is obtained:

$$\text{if } Y = \mathbf{A}X \text{ then } \mathbf{C}_Y = \mathbf{A} \mathbf{C}_X \mathbf{A}^\top,\tag{B.4}$$

where \mathbf{C}_Y denotes the $m \times m$ covariance matrix of Y_1, \dots, Y_m .

Note that the above derivation did not make any assumptions with regard to the probability distribution of the variables X_1, \dots, X_n .

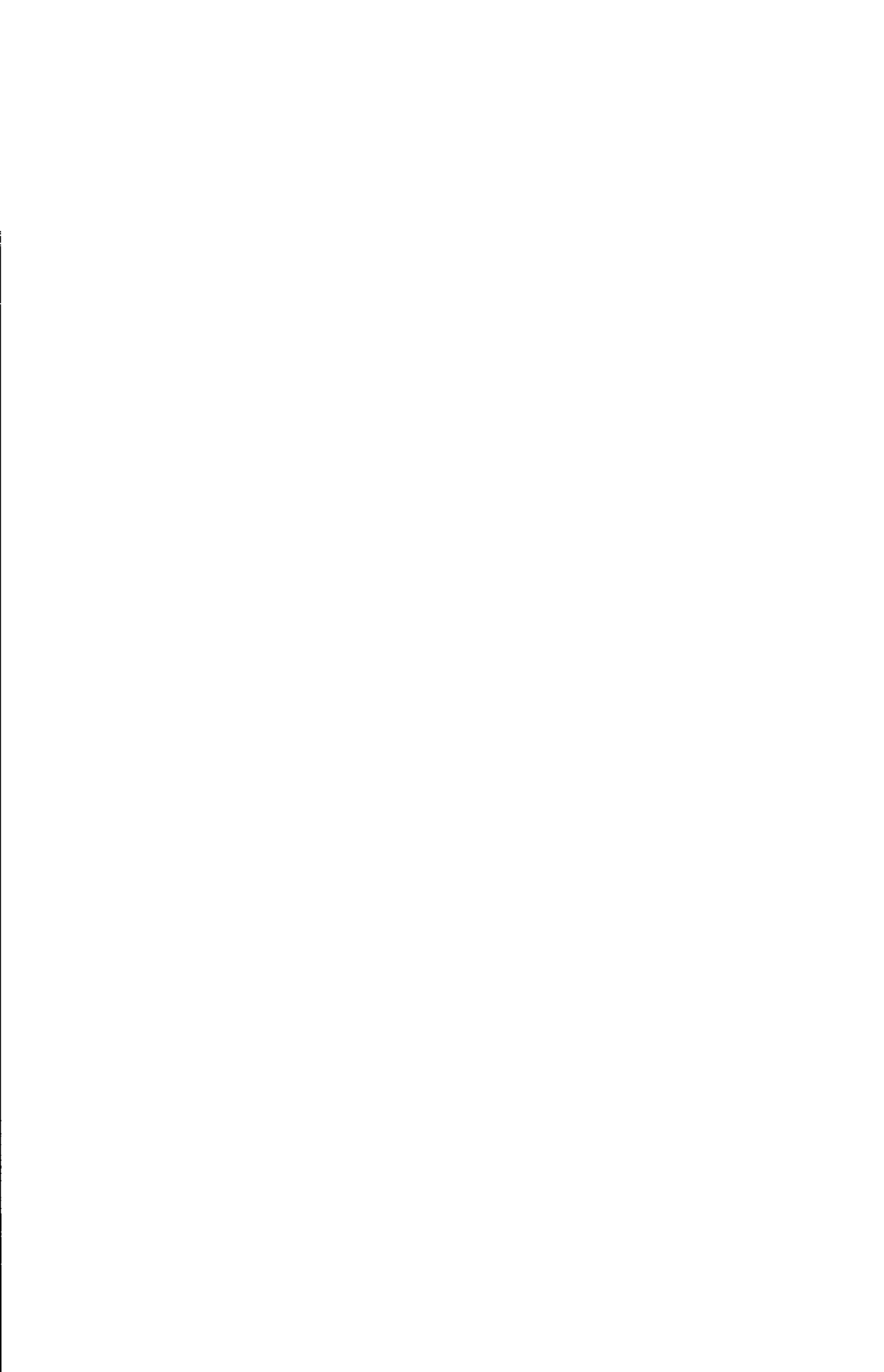
According to (B.1) and (B.2) μ_X can only be calculated if the probability distribution of X is known. However it is possible to obtain an estimate $\hat{\mu}_X$ of μ_X from a limited number of observations of X . Assume that n observations x_1, \dots, x_n are made. Then μ_X may be estimated as

$$\hat{\mu}_X = \frac{1}{n} \sum_{i=1}^n x_i.\tag{B.5}$$

Now suppose that there are n pairs (x_i, y_i) of observations of X and Y . Then $\hat{\mu}_X$ and $\hat{\mu}_Y$ may be estimated with (B.5). After these estimates are obtained σ_{XY} may be estimated as

$$\hat{\sigma}_{XY} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \hat{\mu}_X)(y_i - \hat{\mu}_Y),\tag{B.6}$$

where $\hat{\mu}_X$ and $\hat{\mu}_Y$ denote the estimated mean of X and Y respectively. Chapter 3 uses (B.6). An argument based on degrees of freedom motivates the multiplication factor $1/(n-1)$ instead of $1/n$ [Walp93].



Appendix C

Geometrical Interpretation of the Least-Squares Approximation

This appendix derives an expression for the least-squares approximation, on the basis of the so-called normal equations. Furthermore it gives a geometrical interpretation of the least-squares approximation.

Consider the system

$$\mathbf{Ax} = \mathbf{b}, \tag{C.1}$$

where \mathbf{A} is a real $m \times n$ matrix, \mathbf{x} is a real n vector and \mathbf{b} is a real m vector. According to Definition 3.4, the *least-squares approximation* (LS) $\hat{\mathbf{x}}$ of \mathbf{x} minimizes $\|\mathbf{b} - \mathbf{Ax}\|_2$.

Thus

$$SSE \triangleq \sum_{i=1}^m (b_i - \mathbf{a}_i^T \mathbf{x})^2 \tag{C.2}$$

is minimized, where \mathbf{a}_i^T denotes the i^{th} row of \mathbf{A} . When (C.2) reaches its minimum,

$\forall_{i \in \{1, \dots, n\}} \frac{\partial(SSE)}{\partial x_i} = 0$. This gives the following set of n equalities:

$$\begin{cases} \frac{\partial SSE}{\partial x_1} = -2 \sum_{i=1}^m (b_i - \mathbf{a}_i^T \mathbf{x}) \mathbf{a}_i^T \mathbf{e}_1 = 0 \\ \vdots \\ \frac{\partial SSE}{\partial x_n} = -2 \sum_{i=1}^m (b_i - \mathbf{a}_i^T \mathbf{x}) \mathbf{a}_i^T \mathbf{e}_n = 0 \end{cases}, \quad (\text{C.3})$$

where \mathbf{e}_i denotes the i -th unit vector of \mathcal{R}^n . Rewriting (C.3) in matrix-vector notation obtains the *normal equations*

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}. \quad (\text{C.4})$$

If \mathbf{A} is complex then \mathbf{A}^T is replaced by \mathbf{A}^* . The solution of (C.4) is the LS-squares approximation of \mathbf{x} . Suppose that $m > n$, and $\text{rank}(\mathbf{A}) = n$. It can be proven that in that case the inverse of $\mathbf{A}^T \mathbf{A}$ exists, so the LS solution of (C.1) is

$$\hat{\mathbf{x}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \triangleq \mathbf{A}^L \mathbf{b}, \quad (\text{C.5})$$

where \mathbf{A}^L is called the *left-inverse*¹ of \mathbf{A} . Thus the left-inverse is used instead of the (non existing) inverse of \mathbf{A} to solve (C.1) if that system is overdetermined and \mathbf{A} has full column rank. On the other hand, if $n > m$ and $\text{rank}(\mathbf{A}) = m$ then the LS solution of (C.1) is

$$\hat{\mathbf{x}} = \mathbf{A}^T (\mathbf{A} \mathbf{A}^T)^{-1} \mathbf{b} \triangleq \mathbf{A}^R \mathbf{b}, \quad (\text{C.6})$$

where \mathbf{A}^R is the *right-inverse* of \mathbf{A} .

By substituting the SVD (3.34) of \mathbf{A} in (C.5) and (C.6), these expressions both reduce to the generalized inverse (3.43). This generalized inverse also exists for a rank-deficient matrix \mathbf{A} , which is not true for (C.5) and (C.6). Furthermore, note that if the condition number of \mathbf{A} is $c(\mathbf{A}) = \alpha$, the condition number of $\mathbf{A}^T \mathbf{A}$ is α^2 . Thus (C.5) and (C.6) are poorly conditioned, and therefore it is better to obtain the LS solution of (C.1) with the SVD, as explained in section 3.2.3.

¹ Because $\mathbf{A}^L \mathbf{A} = \mathbf{I}$

Next a geometrical interpretation of (C.5) will be given. Consider the vector $\mathbf{u} \triangleq \mathbf{A}\hat{\mathbf{x}} = \mathbf{A}\mathbf{A}^L\mathbf{b}$, with $\mathbf{u} \in R(\mathbf{A})$. $\mathbf{A}\mathbf{A}^L$ is the matrix of the orthogonal projection on $R(\mathbf{A})$, so $\mathbf{b} - \mathbf{u} \perp R(\mathbf{A})$. Figure C.1 depicts the situation. In this example, \mathbf{A} is a 3×2 matrix, so it maps each vector in \mathbf{R}^2 to a vector on $R(\mathbf{A})$, which is a plane in \mathbf{R}^3 , spanned by the columns \mathbf{a}_1 and \mathbf{a}_2 of \mathbf{A} . In this case $\hat{\mathbf{x}}$ is mapped to \mathbf{u} . Because $\mathbf{u} = \mathbf{A}\hat{\mathbf{x}}$ is the orthogonal projection of \mathbf{b} on $R(\mathbf{A})$, the distance $\|\mathbf{b} - \mathbf{u}\|_2$ is minimal. This shows that indeed $\hat{\mathbf{x}}$ is the LS solution of (C.1).

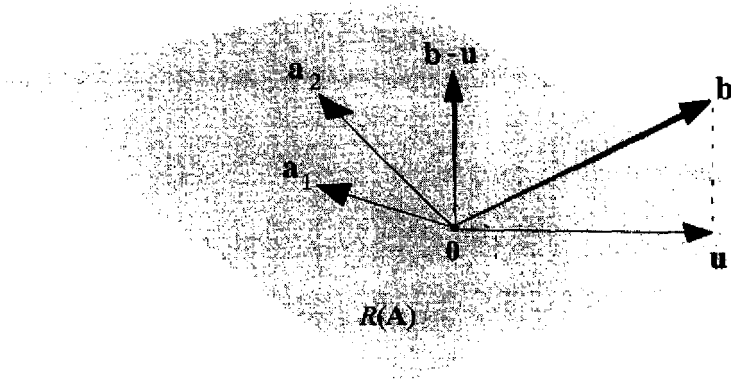
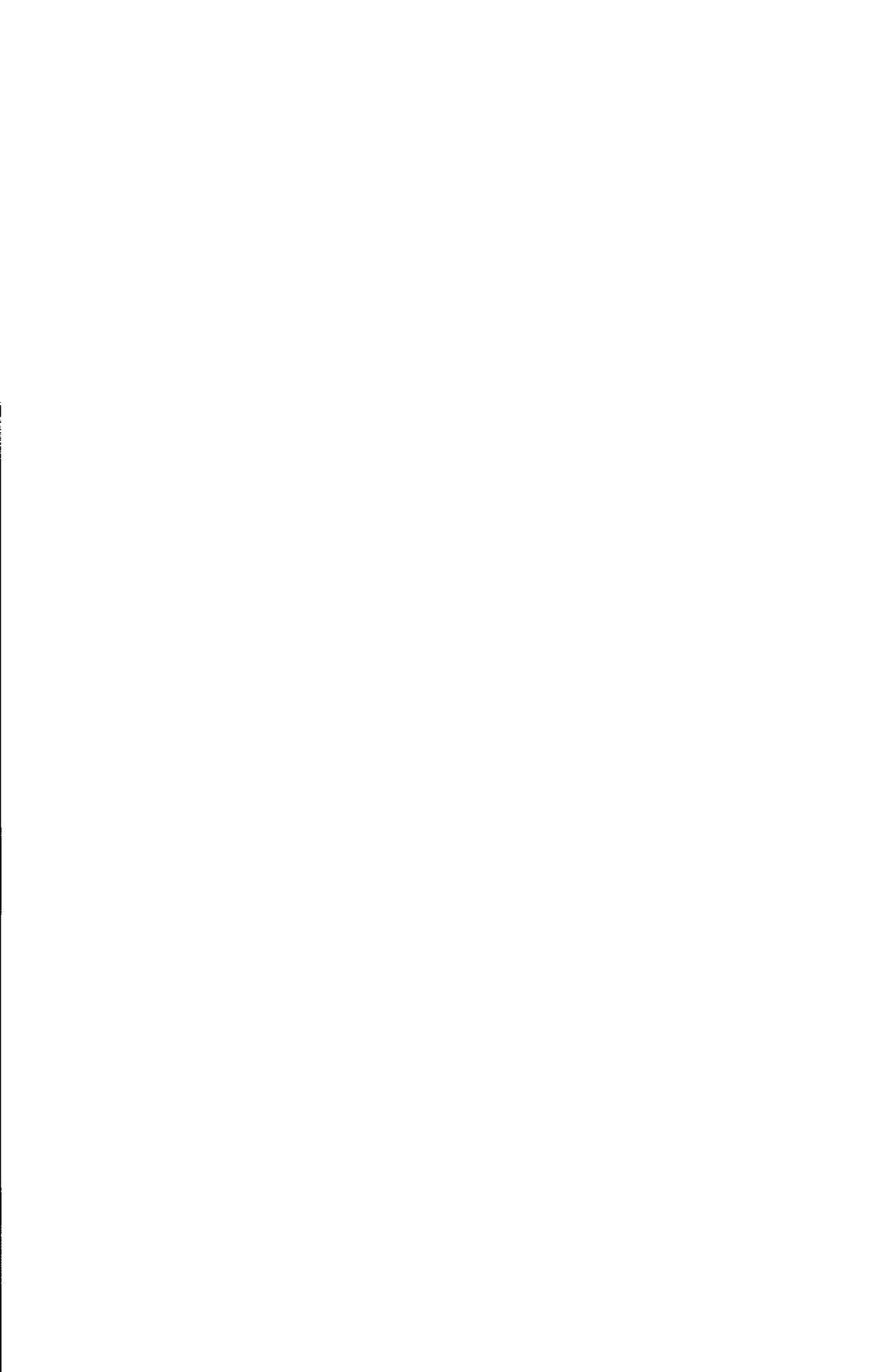


Figure C.1. Graphical interpretation of the least-squares approximation.



Acknowledgement

Many people helped me in some way with the creation of this thesis. Naturally, I would first like to express my gratitude to my first promotor, Wim van Bokhoven, for giving me the opportunity to carry out the research described in this thesis. I especially thank him for letting me diverge from the preplotted route, sailing my own course. My copromotor, Domine Leenaerts was also very flexible in this respect. I also thank him for reviewing many revisions of this thesis. I am indebted to prof. Otten, prof. Vandewalle and prof. Wallinga for their comments on an earlier version of this thesis.

My room mates Wim Kruiskamp and Petar Veselinovic also deserve to be mentioned here. We had many fruitful discussions (along with many useless ones), which really helped me to understand what I myself was thinking. Also I have to say that I admire Petar for not getting angry at me for all my 'language' related jokes.

Special thank to my friends Steven van de Par, René Emmen and Tom Kevenaer, for reading parts of this thesis, and helping me over some seemingly insurmountable hurdles in the progress of the research.

I would like to thank Antoinette, my girlfriend and wife to be, for putting up with me in the moments that I was not responsive at all to external factors (such as her). I hope I can pay her back when she has to write her final law paper.

Finally I thank the people at OCE for allowing me to print this thesis a number of times when the postscript printers at my university department gave up or (most recently) just disappeared. The speed of their printers is impressive (this thesis was printed in under 3 minutes).

Curriculum Vitae

The author of this thesis was born on June 7, 1966 in Goirle, the Netherlands. He graduated at the Odulphus Lyceum in May 1984. On August 30, 1990 he received the Ir. degree in electrical engineering from the Eindhoven University of Technology. From 1990 until 1992 he worked at NCR Systems Engineering Nieuwegein on the management of software in a network environment. From 1992 until 1996 he has been working as a Ph.D. student in the Electronic Design group of the Eindhoven University of Technology. Currently the author is working for ICT Automatisering B.V. in Eindhoven.

This work was supported by the Dutch Foundation for Innovative Research Projects (IOP).

STELLINGEN

behorende bij het proefschrift

A test method for analog circuits
using sensitivity analysis and the singular value decomposition

door J. van Spaandonk

1. De bewering "Since the QRF process chooses maximum norms for the diagonal elements of R , $|A^T A|$ is nearly maximized" in onderstaand artikel is onjuist. In tegenstelling tot wat de auteurs beweren zal bij een niet-triviaal circuit en een relatief groot aantal testpunten de geselecteerde set testpunten verre van D -optimaal zijn.
(G.N. Stenbakken and T.M. Souders, "Test-point selection and testability measures via QR factorization of linear models,"
IEEE Trans. Instrum. Meas., Vol. 36, pp. 406-410, June 1987)
2. Aan het nut van de in onderstaand artikel gebruikte incremental sensitivities voor het testen van analoge circuits moet worden gewijfeld, daar ze slechts toepasbaar zijn voor een beperkte klasse van circuits.
(M. Slamani and B. Kaminska, "Analog circuit fault diagnosis based on sensitivity computation and functional testing,"
IEEE Design and Test of Computers, pp. 30-39, 1992)
3. De ad-hoc aanpak die wordt beschreven in onderstaand artikel voegt weinig toe aan reeds bestaande methoden omdat ze geen garanties geeft m.b.t. de optimaliteit van de geselecteerde metingen en parameters.
(G.J. Hemink, B.W. Meijer and H.G. Kerkhoff, "Testability analysis of analog systems,"
IEEE Trans. CAD, Vol. 9, pp. 573-583, June 1990)
4. De in onderstaand artikel gebruikte definities van "fault masking" en "fault equivalence" zijn in feite equivalent.
(M. Slamani, B. Kaminska and G. Quesnel, "An integrated approach for analog circuits testing with a minimum number of detected parameters,"
IEEE VLSI. Test Symposium, pp. 54-59, 1994)
5. Als circuitbetrouwbaarheid buiten beschouwing wordt gelaten is het behalen van een hoge "fault coverage" bij analoge testen niet zinvol, en kan worden volstaan met een functionele verificatie van het circuitgedrag.
(Dit proefschrift, hoofdstukken 1 en 2)

6. Het klakkeloos gebruik van aan het digitale domein ontleende technieken voor analoog testen leidt tot sub-optimale testmethoden.
(Dit proefschrift, hoofdstukken 1 en 2)

7. Niettegenstaande de overtuiging van Roger Penrose dat “a conscious mind cannot work like a computer” blijft onverkort geldig dat aan het menselijk bewustzijn uitsluitend fysische processen ten grondslag liggen. Zodoende kan het in principe worden gemodelleerd en in een computer gesimuleerd.
(Roger Penrose, *The emperor's new mind : concerning computers, minds, and the laws of physics*, Oxford university press, New York, 1989
Daniel C. Dennet, *Consciousness Explained*, Little brown and company, Boston, 1991)

8. Door de toenemende bezuinigingen is research steeds meer gericht op het behalen van resultaten die op de korte termijn toepasbaar zijn. Net zoals bij een “greedy” aanpak van een optimalisatieprobleem is het gevolg dat een belangrijk deel van de oplossingsruimte nooit wordt bereikt. Op de lange termijn zal dit nadelig uitwerken.

9. In het algemeen wordt te weinig onderkend dat problemen m.b.t. het milieu in principe worden veroorzaakt door een te hoge concentratie mensen op een beperkt oppervlak.

10. Volgens de briljante natuurkundige R.P. Feynman geldt “A scientist looking at nonscientific problems is as dumb as the next guy”. Vanuit dit standpunt bezien moet in de meeste gevallen aan de laatste stellingen bij proefschriften niet al te veel waarde worden gehecht.
(R.P. Feynman, *What do You care what other people think*, Bantam books, New York, 1989)