

## Model order reduction and sensitivity analysis

**Citation for published version (APA):**

Ilievski, Z. (2010). *Model order reduction and sensitivity analysis*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mathematics and Computer Science]. Technische Universiteit Eindhoven.  
<https://doi.org/10.6100/IR685256>

**DOI:**

[10.6100/IR685256](https://doi.org/10.6100/IR685256)

**Document status and date:**

Published: 01/01/2010

**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](http://www.tue.nl/taverne)

**Take down policy**

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

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# Model Order Reduction and Sensitivity Analysis

Copyright ©by Zoran Ilievski, Eindhoven, The Netherlands.

All rights are 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 prior permission of the author.

The work described here is partly financially supported by the European Commission in the framework of the CoMSON RTN project, grant number MRTN-2005-019417.

A catalogue record is available from the Eindhoven University of Technology Library  
ISBN: 978-90-386-2313-9

# Model Order Reduction and Sensitivity Analysis

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de  
Technische Universiteit Eindhoven, op gezag van de  
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een  
commissie aangewezen door het College  
voor Promoties in het openbaar te verdedigen  
op woensdag 25 augustus 2010 om 16.00 uur

door

Zoran Ilievski

geboren te Carshalton, Groot Brittannië

Dit proefschrift is goedgekeurd door de promotor:

prof.dr. W.H.A. Schilders

Copromotor:

dr. J.M.L. Maubach

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Simulations in the electronics industry . . . . .	1
1.2	The COMSON project . . . . .	5
1.3	Outline of this thesis . . . . .	6
<b>2</b>	<b>Electronic Circuit Modelling and Simulation</b>	<b>9</b>
2.1	Electronic circuit modeling . . . . .	9
2.1.1	Circuit networks and topology . . . . .	10
2.1.2	Physical network laws . . . . .	12
2.1.3	Circuit elements . . . . .	13
2.1.4	Modified Nodal Analysis (MNA) . . . . .	14
2.1.5	System equations . . . . .	17
2.2	Properties of MNA systems . . . . .	18
2.3	Electronic circuit simulation . . . . .	20
2.3.1	Linear systems . . . . .	20
2.3.2	Solving the nonlinear system . . . . .	22
2.3.3	Final remarks . . . . .	23
2.4	Summary . . . . .	24
<b>3</b>	<b>Model Order Reduction</b>	<b>25</b>
3.1	Model order reduction in a nutshell . . . . .	25
3.2	Reduction by Projection . . . . .	27
3.3	Balancing methods . . . . .	28
3.4	Proper Orthogonal Decomposition . . . . .	30
3.4.1	Introducing POD . . . . .	30
3.4.2	Derivation . . . . .	30
3.4.3	Singular Value Decomposition . . . . .	32
3.4.4	Application of POD to circuit models . . . . .	34
3.4.5	Error Analysis . . . . .	34
3.5	Summary . . . . .	36
<b>4</b>	<b>Sensitivity Analysis</b>	<b>37</b>
4.1	Adjoint equations . . . . .	37

4.2	Basic sensitivity analysis . . . . .	41
4.3	Sensitivity analysis for differential-algebraic systems . . . . .	42
4.4	Sensitivities for electronic circuits . . . . .	45
4.4.1	State sensitivity . . . . .	46
4.4.2	Observation function sensitivity . . . . .	48
4.5	Applying the direct forward method to circuit sensitivity . . . . .	50
4.6	Application of the backward adjoint method to circuit sensitivity . . . . .	53
4.7	Summary . . . . .	57
<b>5</b>	<b>The Backward Reduced Adjoint Method</b>	<b>59</b>
5.1	Introduction . . . . .	59
5.2	The structure of the backward adjoint equation . . . . .	62
5.3	Tellegen's Theorem . . . . .	64
5.4	Tellegen's theorem and adjoint sensitivities . . . . .	65
5.5	Tellegen's theorem mirrored in vector space . . . . .	69
5.6	The backward reduced adjoint method . . . . .	71
5.7	Analysis of the backward reduced adjoint method . . . . .	73
5.7.1	The cost of the backward reduced adjoint method . . . . .	73
5.7.2	System subspace restriction concerns . . . . .	74
5.7.3	The effect of the projection matrix on sensitivity values . . . . .	75
5.8	Summary . . . . .	76
<b>6</b>	<b>Backward Reduced Adjoint Method II</b>	<b>77</b>
6.1	Method overview . . . . .	77
6.2	Setting up BRAM II . . . . .	78
6.3	Analysis of the BRAM II method . . . . .	82
6.4	The Galerkin projection and POD error estimation . . . . .	84
6.5	Experimental validation of the backward reduced adjoint method . . . . .	89
6.6	Summary . . . . .	90
<b>7</b>	<b>Parameter Dependence of POD basis</b>	<b>91</b>
7.1	State sensitivity approximation complications . . . . .	92
7.2	Projection matrices and basis comparison . . . . .	93
7.3	Principal angles of two identical subspaces . . . . .	95
7.3.1	The simple case $q = m$ . . . . .	95
7.3.2	The general case $q \leq m$ . . . . .	96
7.4	Battery example . . . . .	97
7.5	Summary . . . . .	103
<b>8</b>	<b>Academic &amp; Industrial Examples</b>	<b>105</b>
8.1	A simple test example . . . . .	105
8.1.1	Sensitivity - direct forward method . . . . .	106
8.1.2	Sensitivity - backward adjoint method . . . . .	107
8.1.3	Comparison of the direct forward and backward sensitivity . . . . .	108
8.1.4	The final comparison . . . . .	112

---

8.1.5	The remaining integral . . . . .	113
8.1.6	The complete sensitivity expressions . . . . .	114
8.2	A simple rectifier example . . . . .	115
8.3	Industrial examples . . . . .	121
8.4	Battery charger . . . . .	122
8.4.1	Voltage states only: 0-200u . . . . .	124
8.5	Ring Oscillator . . . . .	131
8.6	Car Transceiver . . . . .	134
8.7	An alternative reduction technique using sensitivities . . . . .	138
8.8	Conclusions . . . . .	139
<b>9</b>	<b>Conclusions</b>	<b>141</b>
9.1	Main results described in the thesis . . . . .	141
9.2	Suggestions for future work . . . . .	142
	<b>Bibliography</b>	<b>145</b>
	<b>Summary</b>	<b>151</b>
	<b>Samenvatting</b>	<b>153</b>
	<b>Acknowledgements</b>	<b>155</b>
	<b>Curriculum vitae</b>	<b>157</b>





# Chapter 1

## Introduction

In this chapter, we give a brief introduction to the field of electronic circuit simulation. It contains a short historical account, and a discussion of the latest requirements that have sparked quite a lot of research in the new century. The chapter also contains an overview of the thesis.

### 1.1 Simulations in the electronics industry

The electronics industry provides the core technology for numerous industrial innovations. Progress in the area of microelectronics is highlighted by several milestones in chip technology, for example microprocessors and memory chips. The ongoing increase in performance and memory density would not have been possible without the extensive use of computer simulation techniques, especially electronic circuit simulation. The basis of the latter is formed by a sound framework of methods from the area of numerical methods, or scientific computing as some would say. In fact, it is not widely known that advances in numerical algorithms also satisfy a "law" that is similar to the well known Moore's law describing advances in chip technology. Figure 1.1 compares the two, and clearly demonstrates that methods for solving linear systems of equations, at the core of every circuit simulation program, have indeed developed in a similar way over the past 35 years.

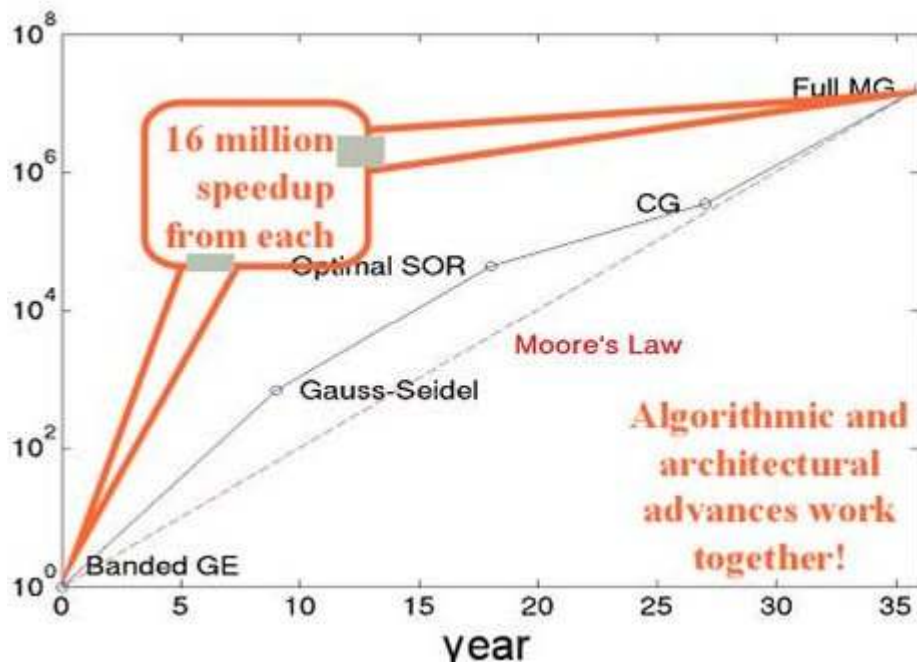


Figure 1.1: Relating Moore's law to advances in linear system solution

Electronic circuit simulation started in the 1970's, and gradually has grown to become one of the most important ingredients of today's virtual design environments in the electronics industry. Designers of electronic circuits, both digital and analog, use these virtual design environments in order to avoid prototyping that would increase the design cycle by many months. Such increased design time is unaffordable nowadays in view of competition. On the other hand, this places an enormous responsibility on the shoulders of the mathematicians and software engineers that provide the knowledge going into these virtual design environments. The latter should be perfectly compatible with experimental set-up, and provide accurate and efficient solutions that enable the designer to design according to the right first time adagio.

During the 1980's and 1990's, many companies developed and used their own in-house circuit simulation software, many of these developments being based upon the Spice simulator [60]. Siemens, later Infineon and Qimonda, developed the Titan simulator [23], whereas Philips, now NXP Semiconductors, developed Philpac [48], which in the 1990's became Panacea and is now named Pstar [34]. But just as in the area of semiconductor device simulation, we now observe a shift towards commercial EDA ("electronic design automation") simulators being used. The three main EDA companies, Cadence, Synopsis, and Mentor, all have their own circuit simulators. Interestingly, most of these are based on university codes that have been upgraded and updated. This holds both for Spectre and Spice [60], the latter appearing in many forms like H-Spice, E-Spice and others. An important observation is that the developments with respect

to the in-house simulators have been essential for the development of the commercial simulators. This is not surprising, as in-house knowledge about the industrial problems is essential for the development of the right tools, and this knowledge is usually only available in the companies.

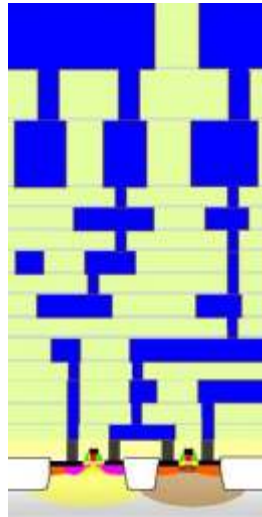


Figure 1.2: Interconnect structure

An important type of analysis in circuit simulators is time domain analysis, which calculates the time-dependent (transient) behaviour of electrical signals in a circuit responding to time varying input signals. A network description of the circuit is generated automatically in computer-aided electronics-design systems from designers drafts or fabrication data files. An input processor translates this network description into a data format reflecting the mathematical model of the system. The mathematical network equations are based on the application of basic physical laws like energy or charge conservation onto network topology and characteristic equations for the network elements. This automatic modeling approach preserves the topological structure of the network and does not aim at systems with a minimal set of unknowns. Hence an initial value problem of differential-algebraic equations (DAEs) is generated which covers characteristic time constants of several orders of magnitude (stiff equations) and suffers from poor smoothness properties of modern transistor model equations.

In recent years, the demands on the capabilities of circuit simulation have become even more stringent. Circuit simulators are actually at the core of all simulations within the electronics industry. Crosstalk effects in interconnect structures<sup>1</sup> (see Figure 1.2), observed only in the new era, are modeled by appending large extracted RLC netlists to the electronic circuit at hand. Also, substrate effects that start playing a crucial role in

<sup>1</sup>In order to connect all devices in the silicon area, a three dimensional metal interconnect structure must be used. Nowadays, this consists of up to 10 metal layers containing "wires", with so-called "vias" between the layers

determining the performance are modeled by extracting, again, a large resistive or RC network and appending it to the original nonlinear electronic circuit. The total network, consisting of the original one and the additional extracted networks, needs to be simulated accurately, both in the time and frequency domain. Many circuit simulation programmes have difficulty in solving such extremely large networks, often in size larger than one million. Thus, new algorithms are needed to cope with such situations that are extremely crucial for designers. This is addressed in many papers, a recent example being [70] where it is shown that the latest techniques in numerical linear algebra can be used to solve such extremely large problems.

Another important aspect is the fact that there is an increasing deviation between design and manufacturing. Due to the ever decreasing feature sizes in modern chips, deviations from the intended dimensions are becoming more probable. Designers need to cope with this, and design the circuits in such a way that a deviation from intended dimensions does not alter the functionality of the circuit. In order to investigate this properly, one needs to assume that all components can possibly be slightly different after manufacturing. The effects this has on the performance of the circuit can be studied by introducing many thousands or even millions of parameters, describing the deviations, and performing a sensitivity analysis of the circuit w.r.t. parameter changes.

Also due to the high complexity, extensive modeling of parasitic<sup>2</sup> effects leads to a very high number of elements to be taken into account during simulation. This is even more true for timing critical circuits where it is necessary to use a very accurate model for the extraction of parasitic elements. Simulation of such circuits can be very time consuming and in some cases not even possible. The complexity caused by this extraction must be reduced to facilitate the simulation of the circuit while preserving accuracy. Now we make the observation that highly accurate parasitic extraction is not necessary for all parts of the design. In fact, each layout contains critical blocks or paths whose timing and performance is crucial for the overall functionality of the chip. High precision interconnect modeling must be used for these circuit parts to verify the functionality of the design. On the other hand, there is interconnect outside of critical paths which adds to the complexity but whose exact model is not necessary and can be simplified. Here too, sensitivity analysis can bring a major achievement in speed-up, by automatically determining the critical parasitic elements that provide the most dominant influence.

The latter two problems have inspired us to study the topic of this thesis. Sensitivity analysis is crucial for the correctness of virtual design environments based on electronic circuit simulators, and gives designers insight in how to alter the designs in order to guarantee more robustness with respect to variability in the design. The problem is that a thorough sensitivity analysis requires derivatives of the solution with respect to a large amount of parameters. This is not feasible using classical methods, being far too time-consuming for modern circuits. Recently proposed methods using the adjoint problem to calculate sensitivities are far more efficient, and these form the basis for our methodology as well. Our work has concentrated on making such methods even more

---

<sup>2</sup>Electronic circuits are first designed using a so-called schematic, based only upon functionality. Actual production needs a layout design which, however, may introduce undesired effects such as crosstalk and other parasitic effects

efficient, by mixing them with concepts from the area of model order reduction. As will be shown in this thesis, this leads to very efficient, robust and accurate methods for sensitivity analysis, even if the underlying circuit is large and the number of parameters is excessive.

## 1.2 The COMSON project

The research described in this thesis was carried out within COMSON, a Marie Curie Research Training Network supported by the European Commission in the framework of the programme "Structuring the European Research Area" within the 6th Framework Research Programme of the European Union. It was established on October 1st, 2005, and ended March 31, 2010. Project leader was the Bergische Universität Wuppertal, and the other partners came from several European countries. Most essential about the latter is that 3 of the main European semiconductor companies were involved, namely NXP Semiconductors, Qimonda and ST Microelectronics, plus universities specialized in electronics: TU Eindhoven, University of Catania and Polytechnic University Bucharest.

COMSON is an acronym for *COupled Multiscale Simulation and Optimization in Nano-electronics*. The main objective of the consortium was to implement an experimental Demonstrator Platform in software code. This platform comprises coupled simulation of devices, interconnects, circuits, EM fields and thermal effects in one single framework. It connects each individual achievement, and offers an adequate simulation tool for optimization in a compound design space.

The consortium used the Demonstrator Platform as a framework to test mathematical methods and approaches. The Demonstrator Platform was used also to assess whether these methods are capable of addressing the industry's problems. Finally the Demonstrator Platform was used, and can be used in future, to train and educate young researchers by hands-on experience on state-of-the-art problems.

The platform did not aim at replacing existing industrial or commercial codes. However, it is capable of analyzing medium sized coupled problems of industrial relevance, thus offering a chance to develop advanced mathematics for realistic problems. Such a platform is urgently needed for academic research, since it provides a natural test bench with state-of-the-art models and parameters from the different domains rather than academic simplifications. The second benefit of such a platform is to collect the knowledge about models and methods, which is widespread distributed over the different nodes of the consortium, thus giving excellent opportunities for transfer of knowledge and mutual stimulation of new research. In addition, the Demonstrator Platform has become a central realization of all needed documents, reports, manuscripts, and courses developed during the project execution. It also played a central role in the training and transfer of knowledge within the COMSON consortium.

The basis of the Demonstrator Platform is the development and validation of appropriate mathematical models to describe the coupling, their analysis (well-posedness) and related numerical schemes. To this end, COMSON was divided in 5 main areas of research:

- Mathematical Modeling and Analysis (MOD)
- Simulation Techniques for Coupled Domains (SIM)
- Model Order Reduction (MOR)
- Optimization (OPT)
- E-Learning (e-L)

The research described in this thesis was mainly conducted within the MOR area. Being only interested in an adequate input-output behavior, distributed effects are described by behavioral models. In the COMSON project, the focus was mainly on generating adequate low-order circuit models by extending MOR to differential-algebraic equations and PDAEs, including first steps towards nonlinearity and time domain.

In the mean time, the COMSON project has finished. As a major deliverable, a book will appear in the course of this year or 2011, describing the achievements of the project in the various areas. See <http://www.comson.org> or [35].

Besides the research that is described in this thesis, also work on the Demonstrator Platform was performed. This work is not described explicitly in the thesis, as it was not a research task.

### 1.3 Outline of this thesis

This thesis presents new theories and efficient computational methods for noise and sensitivity analysis of electronic circuits depending on parameters. To this end, various tools from numerical analysis and circuit simulation are needed: modified nodal analysis, model order reduction, proper orthogonal decomposition and the backward adjoint method. These are the basic building blocks on which our work is based, and methods have been developed using these tools.

As for an outline of the thesis, in Chapter 2 we start with a brief introduction to electronic circuits and the way they are modeled in modern circuit simulation programmes. We introduce the concept of modified nodal analysis (MNA), and show that the resulting systems of equations belong to the class of differential algebraic equations (DAE).

The next chapter reviews methods for model order reduction. This is an extremely popular and important field of research nowadays, both in the scientific computing community and in the area of dynamical systems and control. In the former field, Krylov

subspace techniques is the basis of most techniques, whereas in the latter field one is usually concerned with the solution of large systems of Lyapunov equations and the calculation of so-called Hankel singular values. For linear problems, the theory in both areas is fairly well developed, but for nonlinear problems this is certainly not the case. The most effective technique for such situations is proper orthogonal decomposition, and this method will be discussed in more detail as it forms the basis of our methods.

Chapter 4 then discusses in detail the problem of sensitivity in electronic circuits, and methods to perform analyses. Methods that have been proposed are discussed, and the state-of-the-art is summarized. We then derive the backward adjoint method for the specific case of electronic circuits, discuss stability issues and present an estimate of the computational cost. The latter then clearly rules out the use of forward methods in an industrial context, and also shows that the backward method leaves a lot of room for improvement.

In Chapter 5, the first new development is then discussed, which is the backward reduced adjoint method, published under the name BRAM. It consists of employing the POD method for the forward problem, and then using the resulting basis for the solution of the backward adjoint problem. An analysis of the method is also provided, and the important relation to Tellegen's theorem is demonstrated.

As the original version of BRAM was fairly difficult to analyse, a different version named BRAM II was developed. This method is also based on the use of the POD basis for the forward problem, but first projects the original circuit before constructing the adjoint problem. This is discussed in Chapter 6. It turns out that, for this method, a much more thorough analysis is possible.

A very interesting part of the research concentrated on the parameter dependence of the POD basis. These investigations, presented in Chapter 7, serve two purposes. First of all, it is an extremely intriguing research question to find how POD bases depend on problem parameters for electronic circuits, which, to the best of our knowledge, has not been demonstrated before in this detail. Secondly, the analysis of the methods BRAM and BRAM II is impossible without a statement about the parameter dependence of the POD basis. This is the first time results of this kind are presented for truly industrial examples.

As the proof of the pudding is in the eating, several designers within NXP Semiconductors were asked to provide examples for which we could test our methods. The results of these experiments are presented in Chapter 8. In this chapter we discuss both the parameter dependence of the POD basis and the performance of the newly developed methods for sensitivity analysis.

The thesis is concluded with Chapter 9 which summarizes results achieved and presents an outlook and recommendations for future research in the area of sensitivity analysis.





## Chapter 2

# Electronic Circuit Modelling and Simulation

In this chapter we present an overview of techniques used for modeling and simulation of electronic circuits. We will first introduce the idea of modeling an electronic circuit by breaking down an example circuit into its constituting parts. A description of the general network and electrical properties will be provided, including that of some common electronic elements; resistors, inductors and capacitors. Next, making use of these circuit properties and by applying the method of Modified Nodal Analysis (MNA), it is shown that the resulting system is a differential algebraic equation (DAE). The properties of this DAE model and the matrix representation of a circuit system will be discussed, and we will cover finally how such models can be solved in the time domain. A detailed description can be found in [34].

### 2.1 Electronic circuit modeling

As usual in scientific computing, a model is the first requirement before one can even start to think about simulations. In this section, we therefore discuss the modeling of electronic circuits by providing an example and showing how this can be modeled. The same technique is used also for the much more complex electronic circuits that are encountered in an industrial context. Furthermore, we will discuss some properties of the resulting systems of equations, essential knowledge for the subsequent choice of numerical simulation techniques. The discussion will be somewhat concise, as there are many good works about circuit modeling and simulation available.

### 2.1.1 Circuit networks and topology

A circuit model is an idealized representation of an imperfect real world physical system, created by the combination of ideal components within a network structure of branches and nodes. A branch is defined as the connection between two nodes, and a node as a point connecting at least two circuit components.

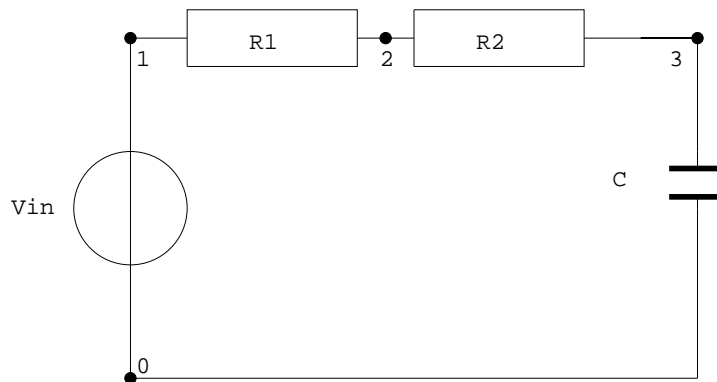


Figure 2.1: An Electronic Circuit

Figure 2.1 is a simple example of a complete circuit diagram. Stripping away the circuit components gives a clearer representation of the network structure which is shown in Figure 2.2. The assumption we make is that the connections between components, the branches they exist on, i.e. the 'wires' of our circuit are ideal and have no resistance, capacitance or inductance.

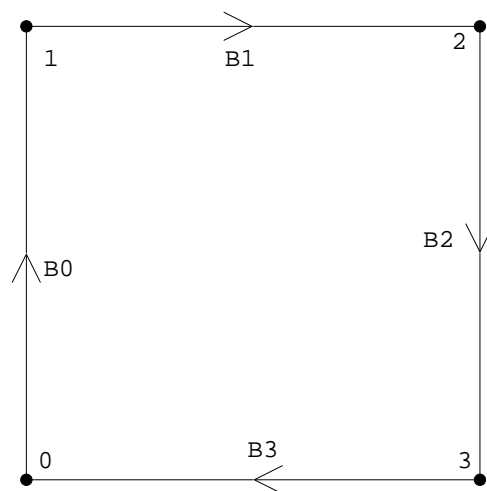


Figure 2.2: Branch-Node Network

The flow of electric current through a circuit has an associated magnitude and a direction. A circuit network is actually a directed graph consisting of nodes connected by branches that possess a direction property. This network topology can be described by an incidence matrix  $A \in \{-1, 1, 0\}^{N_n \times N_b}$ , the columns of this matrix represent the circuit branches, the rows represent circuit nodes. A branch entering a node is signaled by a matrix entry of “-1” if entering, a “1” if leaving and a “0” if there are no branch connections to a node.  $N_n$  is the number of circuit nodes and  $N_b$  the number of circuit branches.

On applying the above rules to the example shown in Figure 2.2 the following incidence matrix is developed.

$$A = \begin{pmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}. \quad (2.1)$$

If we take a look at the first row of (2.1), the row that represents the first node, labeled as 0 in figure 2.2, the entry “1” signals the first branch, branch B0 leaving this node. The entry “-1” in the fourth column of this row signals that the fourth branch, labeled B3 is entering node 0. On picking the first “0” column entry in this first row, one can see that this signals that the second branch, labeled B1 is not connected with node 0 - neither is the third branch, labeled B2, as there is a “0” entry in the third column entry of this row. As a final inspection of this incidence matrix, looking at the final row, the row that represents the fourth node, labeled 3, one can see that the third branch, B2 enters this node and the fourth branch, B3, leaves this node.

In this directed graph network we associate a branch current value  $i_j(t)$  flowing through the  $j^{\text{th}}$  branch, a branch voltage  $u_j(t)$  across the same branch where  $j \in \{1 \cdots N_b\}$  and finally a node voltage for the  $k^{\text{th}}$  node as  $v_k(t)$ , where  $k \in \{1 \cdots N_n\}$ . The branch voltage is defined as the voltage difference between two nodes at either end of a branch. The node voltage is defined as the voltage drop between a node and a common node, usually the ground node, this is labeled as node 0 in figure 2.1.

In column vector form, we write the branch current, branch voltage and node voltage as  $\mathbf{I}(t) \in N_b$ ,  $\mathbf{U}(t) \in N_b$  and  $\mathbf{V}(t) \in N_n$  respectively. These vectors are used to describe fully the characteristics of a circuit, for any topology, combination of circuit elements and input signals. These vectors will also store the current and voltage transient circuit responses, in case of a time dependent simulation.

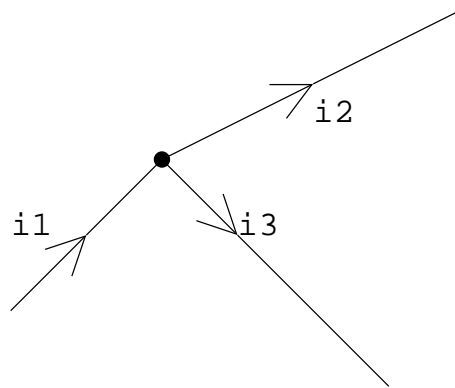


Figure 2.3: KCL

## 2.1.2 Physical network laws

### Kirchhoff's Current Law (KCL)

The algebraic sum of all currents traversing each cutset of a circuit network is always equal to zero. A cutset is a subset of branches that if cut isolates a subset of nodes from the rest of a branch-node graph. A special case is a cutset that isolates exactly one node, this is shown in figure 2.3. This node has two branch currents leaving and one branch current entering, the total current sum on this node is equal to zero.

For a larger cutset or for a complete circuit network, KCL can be expressed by using the incidence matrix  $A$ , and the column vector  $\mathbf{I}(t)$  as,

$$A\mathbf{I}(t) = 0. \quad (2.2)$$

### Kirchhoff's Voltage Law (KVL)

The algebraic sum of all branch voltages around each loop of a network is always equal to zero. An illustration of this, for our example circuit, is shown in figure 2.4. There are four branch voltages  $u_0$ ,  $u_1$ ,  $u_2$  and  $u_3$ , the total sum of these branch voltages is equal to zero.

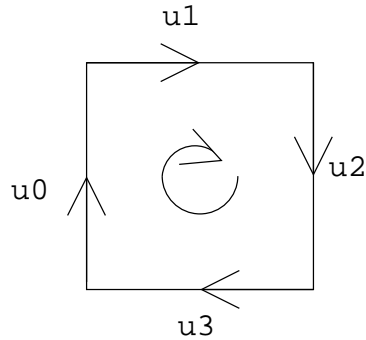


Figure 2.4: Kirchhoff's Voltage Law

For a complete circuit network, by using the incidence matrix, the relationship between all branch and node voltages can be expressed at once as

$$A^T \mathbf{V}(t) = \mathbf{U}(t). \quad (2.3)$$

### 2.1.3 Circuit elements

The five basic building blocks used in circuit modeling are resistors, capacitors, inductors, voltage sources and current sources. Any other real world circuit element, such as diodes and transistors, are often modeled by a combination of these. These elements and circuits can be categorized as either linear or non-linear devices.

#### Linear circuits and elements

Characteristic of the basic building blocks is their linear behavior. For example, the application of a sine wave voltage to a single ideal resistor will return a proportional sine wave current response. An example of an ideal linear circuit would be the potential divider, a circuit composed of two ideal resistors and a voltage source. In that case, application of a sine wave voltage at the input would result in a proportional sine wave voltage of the same form at the output.

Table 2.1 shows the relationship between currents and voltages for the five basic circuit components. The capacitor current  $i_C$  is a function of the capacitance value  $C$  and the dynamics of the branch voltage  $u_C$ , the resistor current  $i_R$  as a function of the resistance value  $R$  and an applied branch voltage of  $u_R$ , and the relationship for an ideal inductor between its branch voltage  $u_L$ , inductance  $L$  and the dynamics of the current  $i_L$ .  $v(t)$ ,  $i(t)$  are the voltage and current sources.

Resistor	$i_R = \frac{u_R}{R}$
Capacitor	$i_C = C \frac{d}{dt} u_C$
Inductor	$u_L = L \frac{d}{dt} i_L$
Voltage Source	$v(t)$
Current Source	$i(t)$

Table 2.1: Common circuit components.

### Nonlinear circuits and elements

Examples of a non-linear components include diodes and transistors, (2.4) models a diode as a nonlinear resistor where  $I_d$  is the diode current,  $I_s$  the reverse bias saturation current,  $V_D$  the voltage drop across the diode and  $V_{th}$  the threshold voltage. It is clear that the response of this component model is non-linear. If we apply a sine wave voltage to this diode it will not preserve the form of the sine wave signal.

$$I_d = I_s (e^{V_D/V_{th}} - 1) \quad (2.4)$$

Real world circuits are composed of many diodes and transistors, their designs can be highly complex and their behavior highly nonlinear. Over the years, we see a significant development in ever more complex models. Originally this development started with the relatively simple Gummel-Poon models, whereas at this moment in time the Penn State Philips (PSP) model is the world standard in MOS modeling [72]. A complicating issue is that these models are derived with a mix of physical insight, huge amounts of experimental results, curve fitting and heuristics. Mathematical properties of the resulting models are not an issue in the derivation, so that commonly used assumptions on smoothness and other important properties are difficult to guarantee. Also, accuracy is the main target of the models, whereas efficiency is not, and this means that the models have a negative impact on the overall performance of circuit simulations.

#### 2.1.4 Modified Nodal Analysis (MNA)

So far we have identified and described the properties and components of electronic circuits, by demonstrating this for a simple example. However, KCL and KVL are readily applied also to larger circuits, and the resulting equations hold also for the more general case. Modified Nodal Analysis, abbreviated MNA, is a method that uses these properties to generate a model of an electronic circuit. We will describe this method by applying it to the circuit shown in Figure 2.1, taking a look at this figure it is easy to identify the node-branch network, circuit components and the physical properties.

The first step is to start with the incidence matrix,

$$\mathbf{A} = \begin{pmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}. \quad (2.5)$$

We next obtain the branch voltages by the application of KVL, equation (2.3), which gives,

$$\begin{pmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}^T \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{pmatrix}. \quad (2.6)$$

This reveals our branch voltages as,

$$u_0 = v_0 - v_1, \quad (2.7)$$

$$u_1 = v_1 - v_2, \quad (2.8)$$

$$u_2 = v_2 - v_3, \quad (2.9)$$

$$u_3 = v_3 - v_0. \quad (2.10)$$

We choose node 0 as our reference (ground) node, the measured voltage drop with itself is always zero and there is no need to calculate this obvious value. Using this fact and updating the branch voltage expressions gives,

$$u_0 = -v_1, \quad (2.11)$$

$$u_1 = v_1 - v_2, \quad (2.12)$$

$$u_2 = v_2 - v_3, \quad (2.13)$$

$$u_3 = v_3. \quad (2.14)$$

Next we apply KCL, equation (2.2),

$$\begin{pmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} i_{b0} \\ i_{b1} \\ i_{b2} \\ i_{b3} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad (2.15)$$

giving,



$$i_{b0} - i_{b3} = 0, \quad (2.16)$$

$$i_{b1} - i_{b0} = 0, \quad (2.17)$$

$$i_{b2} - i_{b1} = 0, \quad (2.18)$$

$$i_{b3} - i_{b2} = 0. \quad (2.19)$$

Equation (2.16) is the current sum across our ground node, and for the reasons stated before we can eliminate this equation. We are only interested in the following sums:

$$i_{b1} - i_{b0} = 0, \quad (2.20)$$

$$i_{b2} - i_{b1} = 0, \quad (2.21)$$

$$i_{b3} - i_{b2} = 0. \quad (2.22)$$

Next we create, starting from our ideal component models, expressions for the currents  $i_{b1}, i_{b2}$  and  $i_{b3}$ . Notice that these currents belong to branches that are not a host to voltage sources, this is why  $i_{b0}$  is left out. The addition of voltage sources to our model will be introduced later.

$$i_{b1} = u_1/R1 = (v_1 - v_2)/R1, \quad (2.23)$$

$$i_{b2} = u_2/R2 = (v_2 - v_3)/R2, \quad (2.24)$$

$$i_{b3} = C \frac{du_3}{dt} = C \frac{dv_3}{dt}. \quad (2.25)$$

These current expressions are substituted into (2.20), (2.21) and (2.22). After grouping time derivative dependent and non-dependent elements we have the following set of differential and algebraic equations. This system of three equations is not yet a complete system for the four unknowns  $u_1, u_2, u_3, i_{b0}$ , there is one more step before we have the full circuit equations.

$$\begin{pmatrix} (v_1 - v_2)/R1 - i_{b0} \\ (v_2 - v_3)/R2 - (v_1 - v_2)/R1 \\ -(v_2 - v_3)/R2 \end{pmatrix} + \frac{d}{dt} \begin{pmatrix} 0 \\ 0 \\ Cv_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (2.26)$$

This final step is the addition of all voltage defining elements, which are the inductors and the voltage sources. We have only one voltage source to add,  $V_{in}$ . Equation (2.27) is the full DAE model for our circuit example, where  $i(V(t))$  contains the circuit current information,  $q(V(t))$  the charge information and  $s(t)$  the circuit input signals.

$$\underbrace{\begin{pmatrix} (v_1 - v_2)/R1 - i_{b0} \\ (v_2 - v_3)/R2 - (v_1 - v_2)/R1 \\ -(v_2 - v_3)/R2 \\ v_1 \end{pmatrix}}_{i(V(t))} + \frac{d}{dt} \underbrace{\begin{pmatrix} 0 \\ 0 \\ Cv_3 \\ 0 \end{pmatrix}}_{q(V(t))} = \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ Vin(t) \end{pmatrix}}_{s(t)}. \quad (2.27)$$

Here is the complete MNA process in short:

1. Apply KCL.
2. Replace all current defining elements (resistors, capacitors and current sources), by the corresponding element equation. This conveniently introduces branch voltages.
3. Apply KVL to express branch voltages in terms of node voltages, substitute these into step 2.
4. Add voltage defining elements to the equation, these are the inductors and voltage sources.

Modified Nodal Analysis is the most frequently used method for setting up the circuit equations. All known circuit simulators use this method, as it is quite efficient in dealing with the full set of KCL and KVL equations. Alternative methods are discussed, for example, in the classic book by Chua [19].

### 2.1.5 System equations

In a few simple steps (2.27) can be restated in a matrix system form. Our unknown circuit values, or circuit states, are used to build a vector  $\mathbf{x}$  of unknowns, called the state vector:

$$\mathbf{x} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ i_{b0} \end{pmatrix}. \quad (2.28)$$

The vectors  $i(V(t))$ ,  $q(V(t))$  are differentiated with respect to the state vector (2.28), which enables us to rewrite (2.27) as

$$\begin{pmatrix} 1/R1 & -1/R1 & 0 & -1 \\ -1/R1 & 1/R1 + 1/R2 & -1/R2 & 0 \\ 0 & -1/R2 & 1/R2 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ i_{b0} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & C & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \frac{d}{dt} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ i_{b0} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ V_{in} \end{pmatrix}. \quad (2.29)$$

Equation (2.29) is a matrix system representation of our circuit example, the general short form is written as

$$\mathbf{G}\mathbf{x}(t) + \mathbf{C}\dot{\mathbf{x}}(t) = \mathbf{s}(t), \quad (2.30)$$

where the matrices  $\mathbf{C}$  &  $\mathbf{G}$  are referred to as the system matrices defined as,

$$\mathbf{C} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & C & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad (2.31)$$

$$\mathbf{G} = \begin{pmatrix} 1/R1 & -1/R1 & 0 & -1 \\ -1/R1 & 1/R1 + 1/R2 & -1/R2 & 0 \\ 0 & -1/R2 & 1/R2 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}. \quad (2.32)$$

## 2.2 Properties of MNA systems

Here we briefly describe some properties of MNA systems that are important for our subsequent analysis and proposed new methods. We will not go into detail, as there are many good books and publications available that treat the MNA systems in-depth, see for example [34] or Chapter 4 in [45] for a rather complete and mathematically deep analysis.

Suppose we are given a circuit with  $N_n$  nodes and  $N_b$  branches, and select one node as the ground node. As usual, the topology of the circuit is described by the incidence matrix  $A$ . Each of the five different types of branches given in Table 2.1 gives a different contribution to the equations. This can be made explicit by grouping branches by category, and split the current and voltage vectors in smaller vectors. So, for example,  $\mathbf{I} = (\mathbf{I}_R^T, \mathbf{I}_C^T, \mathbf{I}_L^T, \mathbf{I}_I^T, \mathbf{I}_V^T)^T$ . Also, the incidence matrix is split:

$$A = (A_R | A_C | A_L | A_I | A_V).$$

Using this notation, we can write the Kirchhoff's Laws in the following form:

$$\sum_{p \in \{R, C, L, I, V\}} A_p \mathbf{I}_p = 0,$$

$$A_p^T \mathbf{V} = \mathbf{U}_p, \quad p \in \{R, C, L, I, V\}.$$

Also, the branch equations can be grouped into five equations. The complete circuit equation for general nonlinear circuits in the end becomes

$$\frac{d}{dt} \mathbf{q}(\mathbf{x}, t) + \mathbf{j}(\mathbf{x}, t) = \mathbf{s}(t) = 0, \quad (2.33)$$

where  $\mathbf{x} = (\mathbf{U}^T, \mathbf{I}_L^T, \mathbf{I}_V^T)^T$ . The circuit matrices  $\mathbf{G}$  and  $\mathbf{C}$  are then given by

$$\mathbf{G} = \begin{pmatrix} A_R \frac{dj_R}{dv_R} A_R^T & A_L & A_V \\ A_I^T & 0 & 0 \\ A_V^T & 0 & 0 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} A_C \frac{dq_C}{dv_C} A_C^T & 0 & 0 \\ 0 & -\frac{dq_L}{dI_L} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

(2.33) is a differential algebraic system which, by the substitution  $\mathbf{y}(\mathbf{x}, t) = \mathbf{q}(\mathbf{x}, t)$  can be written in the charge-oriented form

$$\frac{d}{dt} \mathbf{y}(\mathbf{x}, t) = -\mathbf{j}(\mathbf{x}, t) + \mathbf{s}(t),$$

$$\mathbf{0} = \mathbf{y}(\mathbf{x}, t) - \mathbf{q}(\mathbf{x}, t).$$

The price of this transformation is a doubling of the number of unknowns; on the other hand, it is a form which is more easily handled by the theory. Since  $\mathbf{q}$  does not always have an inverse, one usually considers the function

$$\mathbf{g}(\mathbf{x}, t) = \mathbf{q}(\mathbf{x}, t) + \lambda \mathbf{j}(\mathbf{x}, t),$$

where  $\lambda$  is a positive constant such that  $\mathbf{g}$  is an invertible function with inverse  $\mathbf{g}^{\text{inv}}$ . If  $\mathbf{q}$ ,  $\mathbf{j}$  are continuously differentiable with Jacobian matrices  $G(\mathbf{x}, t)$  and  $C(\mathbf{x}, t)$ , respectively, the Lipschitz constants  $L$ ,  $S$  can be defined as

$$L := \max (\|G(\mathbf{x}, t)\|),$$

$$S := \max (\|C(\mathbf{x}, t) + \lambda G(\mathbf{x}, t)\|),$$

where the maximum is taken over the time domain  $[0, T]$  and all  $\mathbf{x}$  in the domain considered. Then we have the following theorem:

**Theorem 2.1** *The DAE (2.33) (with consistent initial condition) has a unique solution for  $t \in [0, T]$  if the functions  $\mathbf{j}$  and  $\mathbf{g}^{\text{inv}}$  are Lipschitz continuous with finite Lipschitz constants  $L, S$ .*

For this specific case of the DAE arising from an MNA formulation of the circuit equations, a very detailed analysis is given in [79, 80]. The system can be shown to be of index-1 under rather general conditions:

**Theorem 2.2** *Equation (2.33) is of index-1 if and only if*

- *the circuit contains no inductor/current-source cutsets, and*
- *the circuit contains no capacitor/voltage-source loops with at least one voltage source.*

In practice these conditions are usually fulfilled. The theory developed by Tischendorf is very elegant, and we have the impression that its potential has not been used to full extent. For example, we feel that model order reduction could benefit from the detailed structure.

## 2.3 Electronic circuit simulation

A transient simulation is done by finding the solution of the circuit system (2.30) for the state vector, at each time interval, over a time period,  $(0, T)$ .

For system (2.30) to be solvable its matrices must have a regular matrix pencil,  $\{\mathbf{C}, \mathbf{G}\}$ . The linear combination of these two matrices must be regular, an inverse must exist. Even though, for electronic circuits, the  $\mathbf{C}$  matrix is not regular, the matrix pencil always is. The reason for this will soon become clear.

### 2.3.1 Linear systems

Equation (2.34) below is the equation of a linear system, the charge and current vectors are split, as shown before, into a product of the matrices  $\mathbf{C}, \mathbf{G}$  and the state vector  $\mathbf{x}$ . This is not a complicated task for linear circuits, the  $\mathbf{C}, \mathbf{G}$  matrices remain constant over all time periods:

$$\mathbf{C} \frac{d}{dt} \mathbf{x}(t) + \mathbf{G} \mathbf{x}(t) - \mathbf{s}(t) = 0. \quad (2.34)$$

We assume that at the initial condition, at the first time point, the solution is known and found by a DC analysis at  $t = 0$ . This system has state vector solutions at each time point  $t_{(n-1)}$ , where  $n$  is the time point for which we are solving and is written as  $\mathbf{x}_{(n-1)} = \mathbf{x}(t_{(n-1)})$ . The goal here is to determine  $\mathbf{x}_n = \mathbf{x}(t_n)$ , the state vector at time point  $t_n = t_{(n-1)} + \Delta$ , where  $\Delta$  is some stepsize.  $\mathbf{x}_n$  would satisfy the following,

$$\mathbf{C} \frac{d}{dt} \mathbf{x}(t) \Big|_{t=t_n} + \mathbf{G} \mathbf{x}(t_n) - \mathbf{s}(t_n) = 0. \quad (2.35)$$

Equation (2.35) is a differential algebraic equation, which can be solved numerically for  $\mathbf{x}_n$ . There are various methods to perform this simulation and, as demonstrated in [34], the preferred method is a backward difference formula (BDF). For illustration purposes, here we apply an implicit, or backward, Euler discretization to the differential operator:

$$\frac{d}{dt} \mathbf{x}(t) \approx \frac{\mathbf{x}(t_n) - \mathbf{x}(t_{(n-1)})}{t_n - t_{(n-1)}} = \frac{1}{\Delta} (\mathbf{x}(t_n) - \mathbf{x}(t_{(n-1)})). \quad (2.36)$$

Equation (2.34) can now be written as,

$$\mathbf{C} \frac{1}{\Delta} (\mathbf{x}_n - \mathbf{x}_{(n-1)}) + \mathbf{G} \mathbf{x}_n - \mathbf{s}(t_n) = 0. \quad (2.37)$$

Finally it can be written as a simple linear system,

$$\begin{aligned} \left( \frac{1}{\Delta} \mathbf{C} + \mathbf{G} \right) \mathbf{x}_n &= (\mathbf{s}(t_n) - \frac{1}{\Delta} \mathbf{C} \mathbf{x}_{(n-1)}), \\ \mathbf{Y} \mathbf{x}_n &= \mathbf{b}, \end{aligned} \quad (2.38)$$

where the latter equation also contains the initial condition.  $\mathbf{Y}$  and  $\mathbf{b}$  are known, the only unknown is the state vector which we are looking for,  $\mathbf{x}_n$ . The role of the matrix pencil is instantly clear here, the linear combination of the system matrices,  $\mathbf{Y}$ , must be regular to allow a solution for  $\mathbf{x}_n$  to be found.

The following is an outline of how a solution is found. First carry out an LU decomposition on  $\mathbf{Y}$ ,

$$\mathbf{Y}\mathbf{x}_n = \mathbf{b} \quad (2.39)$$

$$\mathbf{Y}_L\mathbf{Y}_U\mathbf{x}_n = \mathbf{b}, \quad (2.40)$$

then find a solution for the linear system for the lower triangular matrix,

$$\mathbf{Y}_L\mathbf{a} = \mathbf{b}, \quad (2.41)$$

and finally use these solutions together with the upper triangular matrix to find the state solution.

$$\mathbf{Y}_U\mathbf{x}_n = \mathbf{a}. \quad (2.42)$$

As circuits are often built in a hierarchical way, it is also possible to use a hierarchical way of solving the linear systems. This is done, for example, in the in-house software package Pstar of NXP Semiconductors that we used for our simulations. We refer to [34] for a detailed discussion of these aspects.

### 2.3.2 Solving the nonlinear system

The non linear circuit system is written as,

$$\frac{d}{dt}\mathbf{q}(\mathbf{x}(t)) + \mathbf{j}(\mathbf{x}(t)) - \mathbf{s}(t) = 0. \quad (2.43)$$

We make the same assumption of having an initial solution to the first time point, and our goal is still to find  $\mathbf{x}_n = \mathbf{x}(t_n)$ . As before we can find  $\mathbf{x}_n$  that satisfies the non-linear differential equation,

$$\frac{d}{dt}\mathbf{q}(\mathbf{x}(t))\Big|_{t=t_n} + \mathbf{j}(\mathbf{x}(t_n)) - \mathbf{s}(t_n) = 0. \quad (2.44)$$

The implicit Euler discretization is again carried out and gives us,

$$\frac{1}{\Delta}[\mathbf{q}(\mathbf{x}_n) - \mathbf{q}(\mathbf{x}_{(n-1)})] + \mathbf{j}(\mathbf{x}(t_n)) - \mathbf{s}(t_n) = 0. \quad (2.45)$$

Grouping the same terms

$$\left[\frac{1}{\Delta}\mathbf{q}(\mathbf{x}_n) + \mathbf{j}(\mathbf{x}_n)\right] - \mathbf{s}(t_n) - \frac{1}{\Delta}\mathbf{q}(\mathbf{x}_{(n-1)}) = 0. \quad (2.46)$$

Equation (2.46) can be solved by applying the Newton method.

First we define,

$$H(\mathbf{x}_n) = \left[\frac{1}{\Delta}\mathbf{q}(\mathbf{x}_n) + \mathbf{j}(\mathbf{x}_n)\right] - \underbrace{\left[\mathbf{s}(t_n) + \frac{1}{\Delta}\mathbf{q}(\mathbf{x}_{(n-1)})\right]}_{\text{Known constant value for each time step}}. \quad (2.47)$$

then,

$$H'(\mathbf{x}_n) = \left[\frac{1}{\Delta} \frac{d}{dx} \mathbf{q}(\mathbf{x}_n) + \frac{d}{dx} \mathbf{j}(\mathbf{x}_n)\right],$$

or

$$H'(\mathbf{x}_n) = \left[\frac{1}{\Delta} \mathbf{C}(\mathbf{x}_n) + \mathbf{G}(\mathbf{x}_n)\right]. \quad (2.48)$$

We then make the following iteration to find  $\mathbf{x}_n$ , the converged value of a sequence of  $\mathbf{x}_n^k$ :

$$\mathbf{x}_n^{k+1} = \mathbf{x}_n^k - \left[\frac{1}{\Delta} \mathbf{C}(\mathbf{x}_n^k) + \mathbf{G}(\mathbf{x}_n^k)\right]^{-1} [H(\mathbf{x}_n^k)]. \quad (2.49)$$

Notice that the matrix pencil is again an important factor in (2.49). Besides this, we need to take care of constructing a good initial guess, and also have to employ damping techniques such as "gmin stepping" in order to ensure convergence of the sequence of Newton iterates. These aspects are discussed in detail in [34].

### 2.3.3 Final remarks

In practice, more general BDF and one step schemes are used for the discretization of the time dependent problem. Also, sophisticated adaptive time stepping schemes are used that are able to detect discontinuities in the source terms and, consequently, changing character of the solution. It is not our intention to discuss here all of the advanced



features that are used in the very mature field of circuit simulation. Other types of simulations, such as AC analysis or periodic steady state analysis (PSS) are also not discussed here. We refer to [34] for a recent and fairly complete overview.

## **2.4 Summary**

In this chapter, we briefly discussed how electronic circuits can be modeled mathematically and solved numerically. An important aspect is the differential algebraic character of the system of equations, which has an influence on the subsequent analysis and numerical solution procedures. It also has an influence when using model order reduction, as most of these techniques have been developed for state space systems and not descriptor type systems. Finally, it is extremely important to take the DAE character into account when deriving the adjoint equations that are used in sensitivity analysis. In subsequent chapters, several of the properties discussed in this chapter are used.

## Chapter 3

# Model Order Reduction

For a small system model, such as the model examined in the previous chapter, the solution can be found after a short time integration. As systems increase in size, so do their models, and so does the time taken to generate system solutions. Model Order Reduction is a wide field that is concerned with finding a lower dimensional system, whence faster to solve, that gives a very good approximation to the original system. Each method finds an approximating subspace that spans the major characteristics of a system's response, a projection matrix is also found that is used to project the larger system onto the smaller subspace. In this chapter we will first introduce the concept of reduction by projection, and then discuss the method of proper orthogonal decomposition that is currently the method of choice for nonlinear systems.

### 3.1 Model order reduction in a nutshell

There are several definitions of model order reduction, and it depends on the context which one is preferred. Originally, MOR was developed in the area of systems and control theory, which studies properties of dynamical systems in application for reducing their complexity, while preserving their input-output behavior as much as possible. An example of such a system is the following:

$$\frac{dx}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}), \quad (3.1)$$

where  $\mathbf{u}$  is the input of the system,  $\mathbf{y}$  is the output, and  $\mathbf{x}$  is the so-called state variable. The complexity of the system is characterized by the number of its state variables, and model order reduction may be viewed as the task of reducing the dimension of the state vector. In other words, we should find a dynamical system of the form

$$\frac{d\hat{\mathbf{x}}}{dt} = \hat{\mathbf{f}}(\hat{\mathbf{x}}, \mathbf{u}), \quad \hat{\mathbf{y}} = \hat{\mathbf{g}}(\hat{\mathbf{x}}, \mathbf{u}), \quad (3.2)$$

where the dimension of  $\hat{\mathbf{x}}$  is much smaller than that of the original state space vector  $\mathbf{x}$ .

The field has also been taken up by numerical mathematicians, especially after the publication of methods such as PVL [28]. Nowadays, model order reduction is a flourishing field of research, both in systems and control theory and in numerical analysis. This has a very healthy effect on MOR as a whole, bringing together different techniques and different points of view, pushing the field forward rapidly.

So what is model order reduction about? We need to deal with the simplification of dynamical models that may contain many equations and/or variables ( $10^5 - 10^9$ ). Such simplification is needed in order to perform simulations within an acceptable amount of time and limited storage capacity, but with reliable outcome. In some cases, we would even like to have on-line predictions of the behavior with acceptable computational speed, in order to be able to perform optimizations of processes and products.

Model Order Reduction tries to capture the essential features of a structure. This means that at an early stage of the process, the most basic properties of the original model must already be present in the lower dimensional approximation. At a certain moment the process of reduction is stopped. At that point all necessary properties of the original model must be captured with sufficient precision. All of this has to be done automatically.

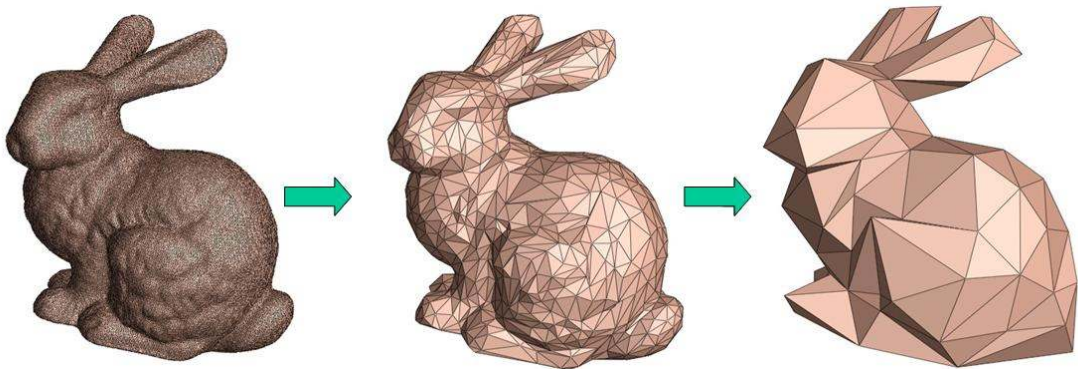


Figure 3.1: Graphical illustration of model order reduction

Figure 3.1 illustrates the concept in a graphical easy-to-understand way, demonstrating that sometimes very little information is needed to describe a model. This example with pictures of the Stanford Bunny shows that, even with only a few facets, the rabbit can still be recognized as such (Graphics credits: Harvard University, Microsoft Research). Although this example was constructed for an entirely different purpose, and does not

contain any reference to the way model order reduction is performed mathematically, it can be used to explain (even to lay persons) what model order reduction is about.

In the history of mathematics we see the desire to approximate a complicated function with a simpler formulation already very early. In the year 1807 Fourier (1768-1830) published the idea to approximate a function with a few trigonometric terms. In linear algebra the first step in the direction of model order reduction came from Lanczos (1893-1974). He looked for a way to reduce a matrix in tridiagonal form [52, 53]. W.E. Arnoldi realized that a smaller matrix could be a good approximation of the original matrix [5]. He is less well-known, although his ideas are used by many numerical mathematicians. The ideas of Lanczos and Arnoldi were already based on the fact that a computer was available to do the computations. The question, therefore, was how the process of finding a smaller approximation could be automated.

The fundamental methods in the area of Model Order Reduction were published in the eighties and nineties of the last century. In 1981 Moore [59] published the method of Truncated Balanced Realization, in 1984 Glover published his famous paper on the Hankel-norm reduction [33]. In 1987 the Proper Orthogonal Decomposition method was proposed by Sirovich [76]. All these methods were developed in the field of systems and control theory. In 1990 the first method related to electronic systems was born, in Asymptotic Waveform Evaluation [66]. The focus of this paper was on finding Padé approximations. Then, in 1993, Freund and Feldmann proposed Padé Via Lanczos [28] and showed the relation between the Padé approximation and Krylov spaces. In 1995 another fundamental method was published. The authors of [63] introduced PRIMA, a method based on the ideas of Arnoldi, instead of those of Lanczos, the emphasis being on the preservation of passivity.

In more recent years much research has been done in the area of the Model Order Reduction. Consequently a large variety of methods is available. Some are tailored to specific applications, others are more general. There is one striking fact, however, namely that most methods are based on projection. In the next section, we will go into this in more detail. For an extensive recent overview of methods, see [3, 13, 73].

## 3.2 Reduction by Projection

Consider the linear equation

$$C\dot{\mathbf{x}} + G\mathbf{x} - \mathbf{s}(t) = 0, \quad \mathbf{x} \in \mathbb{R}^n \quad (3.3)$$

The DAE circuit model (3.3) has solutions in the full basis,  $\mathbf{x} \in \mathbb{R}^n$ , this could be approximated by using a smaller vector  $\mathbf{z} \in \mathbb{R}^r$  where  $\mathbf{z} \approx V\mathbf{x}$ . The projection matrix  $V$  is of row dimension  $n$ , but the number of columns  $r$  is usually much smaller. We substitute this approximation into (3.3) to give,

$$\underbrace{CV\dot{\mathbf{z}} + GV\mathbf{z} - \mathbf{s}(t)}_{\mathbf{r}(\mathbf{z}), \text{ Residual}} \neq 0 \quad \mathbf{z} \in \mathbb{R}^r. \quad (3.4)$$

System (3.4) is no longer equal to zero, it has a residual value  $\mathbf{r}(\mathbf{z})$ . It is also an overdetermined system with more equations than unknowns.

We are interested in the solution of  $\mathbf{z}(t)$  such that the remaining residual  $\mathbf{r}(\mathbf{z})$  is orthogonal to the subspace spanned by the approximating basis vectors. We apply Galerkin projection giving the following constraint,

$$V^T \mathbf{r}(\mathbf{z}) = \mathbf{0}. \quad (3.5)$$

Applying (3.5) to (3.3) we finally have,

$$\begin{aligned} V^T CV\dot{\mathbf{z}} + V^T GV\mathbf{z} - V^T \mathbf{s}(t) &= 0, \\ \hat{C}\dot{\mathbf{z}} + \hat{G}\mathbf{z} - \hat{\mathbf{S}}(t) &= 0 \end{aligned} \quad (3.6)$$

We now have a determined system with an equal number of unknowns and equations, it is a system that is also solvable by Newton methods if the matrix pencil  $\{\hat{C}, \hat{G}\}$  is satisfied.

Projection methods are extremely popular with the area of model order reduction. PVL, Arnoldi and PRIMA all are within this category, and also the Laguerre method [37, 49]. Most of these methods are developed for linear problems, and researchers in recent years have concentrated on aspects like passivity, parametrization and structure preservation [9, 30]. Also, the concept of realizability plays an important role, as one would like the reduced order model to be in the form of an electronic circuit. These are non-trivial problems, however, for which more research is needed.

### 3.3 Balancing methods

Model order reduction is certainly not the exclusive domain of numerical analysts; in the area of dynamical systems and control, one has worked on reduced order modeling for a long time. Within this area, methods have been developed that are entirely different from Krylov subspace methods. To illustrate how the methods work, consider a linear dynamical system

$$\frac{dx}{dt} = Ax + Bu,$$

$$\mathbf{y} = \mathbf{C}^T \mathbf{x} + \mathbf{D}\mathbf{u}.$$

This is the most-used form for a linear input-output system where, as before,  $\mathbf{u}$  is the input,  $\mathbf{y}$  is the output, and  $\mathbf{x}$  is the state space vector [3]. It is easy to see that applying a state space transformation

$$\mathbf{T}\hat{\mathbf{x}} = \mathbf{x},$$

does not affect the input-output behavior of the system. This transformation can, thus, be chosen in a more or less optimal way (later it will become clear what this means). The method of truncated balanced realization (TBR) uses a transformation that is based upon finding solutions of the Lyapunov equations<sup>1</sup>

$$\begin{aligned} \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T &= 0, \\ \mathbf{A}^T\mathbf{Q} + \mathbf{Q}\mathbf{A} + \mathbf{C}^T\mathbf{C} &= 0. \end{aligned}$$

The matrices  $\mathbf{P}$  and  $\mathbf{Q}$  are the controllability and observability Gramians associated with the linear time-invariant system  $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ . Finding these is a rather time-consuming effort, but whenever found a balancing transformation can be carried out in such a way that  $\mathbf{P} = \mathbf{Q} = \mathbf{\Sigma} = \text{diag}(\sigma_i)$ , where the latter are the singular values of the Hankel operator [71].

The balancing transformation changes the matrices in the dynamical system, and also leads to transformed Gramians [62]:

$$\begin{aligned} \mathbf{P}' &= \mathbf{T}^{-1}\mathbf{P}\mathbf{T}^{-T}, \\ \mathbf{Q}' &= \mathbf{T}^T\mathbf{Q}\mathbf{T}, \end{aligned}$$

that are balanced.

Reduction can then be achieved by discarding the smallest singular values. A big advantage of balancing methods is that the error, in a suitable norm, can be shown to be bounded by the sum of discarded singular values.

For nonlinear problems, balancing techniques have also been developed, but the field is still under development. Early work can be found in [71]. More recent work can be found in [31, 84].

---

<sup>1</sup>In order to guarantee the existence of  $\mathbf{P}$  and  $\mathbf{Q}$ , we need to assume asymptotic stability of the original system. Furthermore,  $\mathbf{T}$  must be non-singular, otherwise the input-output behavior may be affected.

## 3.4 Proper Orthogonal Decomposition

Proper Orthogonal Decomposition, also known as Principal Component Analysis or Karhunen-Loève theorem, is a method that is used to create lower dimensional models that accurately approximate data sets extracted from an original system. A data set can be extracted from almost anything, for example; image data, video footage, sound recordings and circuit response data. POD first identifies a set of basis functions that spans a given data set, next a subset of these basis vectors is selected in such a way as to preserve the most dominant, or required, characteristics of a system. A reconstruction or approximate full model is carried out by projection on a reduced space consisting of a linear combination of these dominant basis vectors, the basis selection procedure also provides an error estimate of the reconstructed system. In this section we will introduce the concepts of POD, followed by its derivation and show how POD can be applied to a data set generated by our DAE circuit model and some error analysis.

### 3.4.1 Introducing POD

The data set collected from a transient circuit simulation is referred to as the snapshot matrix, it is the collection of vector states at each time point in the simulation (in the discrete case, we assume that the solution is stored at  $m$  time points):

$$X = \{\mathbf{x}(t)\} \quad t \in [0, T] \text{ or } t \in \{0 = t_1, t_2, \dots, t_m = T\} \quad (3.7)$$

POD finds a subspace approximating the space spanned by the columns of this snapshot matrix (for the case of a finite number of snapshots) in an optimal least-squares sense, it is looking for a  $d$ -dimensional subspace  $S = \text{span}\{\varphi_1, \dots, \varphi_d\} \subset \mathbb{R}^n$  that minimizes,

$$\|X - \rho X\|^2 := \kappa \cdot \|\mathbf{x} - \rho \mathbf{x}\|_{L_2}^2. \quad (3.8)$$

The components of equation (3.8) are  $\kappa = \frac{1}{T}$  for continuous time or  $\kappa = \frac{1}{m}$  discrete time data.  $\rho : \mathbb{R}^n \rightarrow S$  is the orthogonal projector on to the subspace  $S$ , where  $S$  is the approximating orthonormal basis that is selected by POD, it is the POD basis.

### 3.4.2 Derivation

We will give a short derivation here to introduce important concepts of the POD method. Our starting point is the truncated POD basis,

$$\{\varphi_1, \dots, \varphi_d\}. \quad (3.9)$$

This orthonormal basis  $S$  can be completed by the Gram-Schmidt process to form a full basis of  $\mathbb{R}^n$ ,

$$\text{span}\{\varphi_1, \dots, \varphi_d, \varphi_{d+1}, \dots, \varphi_n\} = \mathbb{R}^n \quad (3.10)$$

Each snapshot  $\mathbf{x}(t)$  in the snapshot matrix can be expressed as an expansion in the full basis,

$$\mathbf{x}(t) = \sum_{j=1}^n (\mathbf{x}(t), \varphi_j) \varphi_j, \quad (3.11)$$

or as an approximating expansion in the smaller POD basis,

$$\rho \mathbf{x}(t) = \sum_{j=1}^d (\mathbf{x}(t), \varphi_j) \varphi_j. \quad (3.12)$$

At this point we can already find an expression for the POD basis projection error, the error between the expansions (3.11) and (3.12), this is done by the application of (3.8),

$$\frac{1}{T} \|\mathbf{x} - \rho \mathbf{x}\|_{L_2}^2 = \frac{1}{T} \int_0^T \left\| \sum_{j=d+1}^n (\mathbf{x}(t), \varphi_j) \varphi_j \right\|_{L_2}^2 dt \quad (3.13)$$

$$= \sum_{j=d+1}^n (\mathbf{x}(t)) \frac{1}{T} \int_0^T (\varphi_j, \varphi_j)^2 dt \text{ using } (\varphi_j, \varphi_k) = \delta^{j,k} \quad (3.14)$$

$$= \sum_{j=d+1}^n \langle \mathbf{x}, \varphi_j \rangle^2. \quad (3.15)$$

The overall strategy to derive POD is to find an ordered orthonormal basis of  $\mathbb{R}^n$  such that  $\mathbf{x}$  can be expressed as an expansion,

$$\langle \mathbf{x}, \varphi_1^2 \rangle > \dots > \langle \mathbf{x}, \varphi_d^2 \rangle > \langle \mathbf{x}, \varphi_{d+1}^2 \rangle > \dots > \langle \mathbf{x}, \varphi_n^2 \rangle. \quad (3.16)$$

The first basis vector  $\varphi_1$  is chosen so that it maximizes the averaged projection,

$$\langle \mathbf{x}, \varphi_1 \rangle^2 = \underbrace{\max}_{\varphi \in \mathbb{R}^n} \langle \mathbf{x}, \varphi \rangle^2, \quad (3.17)$$



the following basis vector  $\varphi_2$ , is chosen so that it has the second highest maximising effect on the averaged projection. The constraints placed on these successive choices in this sequence would be  $\varphi_i \perp \varphi_j$  for  $j = 1, \dots, i - 1$  and that  $\|\varphi\|_2 = 1$ . In this way, we ascertain we have build an orthonormal basis from a set of orthogonal vectors  $\varphi_1, \dots, \varphi_n$  with singular values  $\lambda_i \geq 0$ .

The above process is equivalent to the eigenvalue problem,

$$\mathbf{M}(\varphi) = \lambda\varphi. \quad (3.18)$$

where,

$$\mathbf{M}(\varphi) = \langle (\mathbf{X}, \varphi) \cdot \mathbf{x} \rangle \quad (3.19)$$

The averaging operator  $\mathbf{M}(\varphi)$  can be written as,

$$\mathbf{M}(\varphi) = \langle (\mathbf{x}, \varphi)\mathbf{x} \rangle = \frac{1}{m} \sum_{j=1}^m \sum_{k=1}^m \underbrace{\mathbf{x}_k^T \varphi}_{\in \mathbb{R}} \mathbf{x}_j = \frac{1}{m} \sum_{j=1}^m \sum_{k=1}^m \mathbf{x}_j \mathbf{x}_k^T \varphi = \frac{1}{m} \mathbf{X}\mathbf{X}^T \varphi \quad (3.20)$$

giving a final expression, in terms of our snapshot matrix  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_1) \in \mathbb{R}^{n \times m}$ , for the eigenvalue problem as,

$$\frac{1}{m} \mathbf{X}\mathbf{X}^T \varphi = \lambda\varphi \quad (3.21)$$

On solving (3.21), we will instantly have found our POD basis,

$$\varphi_1, \dots, \varphi_n, \quad (3.22)$$

ordered such that,

$$\lambda_1 > \dots > \lambda_n. \quad (3.23)$$

### 3.4.3 Singular Value Decomposition

The singular value decomposition can be used to solve the eigenvalue problem. We will demonstrate the svd decomposition on a snapshot matrix  $\mathbf{X}$ ,

$$\mathbf{X} = \Phi \begin{pmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \Psi^T \quad (3.24)$$

where,

$$\Phi = (\varphi_1, \dots, \varphi_n) \in \mathbb{R}^{n \times n} \quad (3.25)$$

$$\Psi = (\nu_1, \dots, \nu_m) \in \mathbb{R}^{m \times m} \quad (3.26)$$

$$\Sigma = \text{diag}(\sigma_1, \dots, \sigma_k) \quad (3.27)$$

Equation (3.25), is an orthonormal set of left singular vectors and, (3.26) is an orthonormal set of right singular vectors, (3.27) contains the singular values. These vectors and singular values satisfy,

$$\mathbf{X}\nu_i = \sigma_i\varphi_i \quad (3.28)$$

$$\mathbf{X}^T\varphi_i = \sigma_i\nu_i. \quad (3.29)$$

To solve (3.21), we apply POD to the eigenvalue problem formed from the correlation matrix.

$$\mathbf{X}\mathbf{X}^T\varphi_i = \sigma_i^2\varphi_i, \quad (3.30)$$

$$\lambda_i = \sigma_i^2. \quad (3.31)$$

Once we have the left and right vectors, and the ordered singular values we can find an optimal POD basis from an energy point of view, here we choose a set of containing  $d$  basis vectors so that the corresponding summed singular values give 99% energy conservation in the following,

$$\text{Energy} = \frac{\sum_{i=1}^d \sigma_i}{\sum_{i=1}^n \sigma_i} \times 100. \quad (3.32)$$

We now have a suitable matrix  $\mathbf{V}$  that can be used to approximate our original system.

$$\rho = \mathbf{V}\mathbf{V}^T, \quad (3.33)$$

$$\mathbf{V}^T\mathbf{V} = \mathbf{I}_{(d \times d)}. \quad (3.34)$$

### 3.4.4 Application of POD to circuit models

The first step is to build a snapshot of states over a time period by carrying out a simulation of the circuit equation. Here, for simplicity, we state the linear equation, but in principle POD is suited for nonlinear equations as well. The main problem is in the use of the Galerkin projection in the nonlinear case, which is described in more detail in Chapter 7. So, let's consider here the linear circuit equation

$$C \frac{d}{dt} \mathbf{x}(t) + G\mathbf{x}(t) - \mathbf{S}(t) = 0. \quad (3.35)$$

Once we have generated a POD basis and projection matrix  $V$ , we are able to approximate  $\mathbf{x}(t)$

$$\mathbf{x}(t) \approx V\mathbf{z}(t). \quad (3.36)$$

By using the Galerkin projection method described earlier, we can create the reduced system,

$$\hat{C} \frac{d}{dt} \mathbf{z}(t) + \hat{G}\mathbf{z}(t) - \hat{\mathbf{S}}(t) = 0 \quad (3.37)$$

with,

$$\hat{C} = V^T C V \in \mathbb{R}^{d \times d} \quad (3.38)$$

$$\hat{G} = V^T G V \in \mathbb{R}^{d \times d} \quad (3.39)$$

$$\hat{\mathbf{S}} = V^T \mathbf{S} \in \mathbb{R}^d \quad (3.40)$$

### 3.4.5 Error Analysis

Proper orthogonal decomposition is not a method that is easy to analyze. The main reason is, of course, that the method deals with nonlinearities that may come in many different variations. Several researchers have become interested in the method, and published on error estimates [39, 85, 87]. The most complete and relatively recent reference is, however, the seminal paper by Rathinam and Petzold [68]. In that paper, basic properties of POD are investigated and, more importantly, an analysis of all errors involved in solving an ODE initial value problem using a POD reduced order model is provided. The main theorem that is proved in the paper is the following:

**Theorem 3.1** Consider solving the initial value problem  $\mathbf{x}' = \mathbf{f}(\mathbf{x}, t)$ ,  $\mathbf{x}(0) = \mathbf{x}_0$ , using the POD reduced order model in the interval  $[0, T]$ . Let  $\rho \in \mathbb{R}^{k \times n}$  be the relevant projection matrix, and let  $S$  denote the affine subspace onto which POD projects. Write the solution of the full model  $\mathbf{x}(t)$  and the solution  $\hat{\mathbf{x}}(t)$  of the reduced model as

$$\mathbf{x}(t) = \rho^T \mathbf{u}(t) + \rho_c^T \mathbf{v}(t) + \bar{\mathbf{x}},$$

and

$$\hat{\mathbf{x}}(t) = \rho^T \mathbf{u}(t) + \rho^T \mathbf{w}(t) + \bar{\mathbf{x}},$$

so that the errors  $\mathbf{e}_0(t)$  and  $\mathbf{e}_i(t)$  and the projected solution  $\tilde{\mathbf{x}}(t)$  are given by

$$\begin{aligned} \mathbf{e}_0(t) &= -\rho_c^T \mathbf{v}(t), \\ \mathbf{e}_i(t) &= \rho^T \mathbf{w}(t), \end{aligned}$$

and

$$\tilde{\mathbf{x}}(t) = \rho^T \mathbf{u}(t) + \bar{\mathbf{x}}(t).$$

Note that  $\mathbf{u}(t) \in \mathbb{R}^k$ ,  $\mathbf{w}(t) \in \mathbb{R}^k$ , and  $\mathbf{v}(t) \in \mathbb{R}^{n-k}$ . Let  $\gamma \geq 0$  be the Lipschitz constant of  $\rho \mathbf{f}(\mathbf{x}, t)$  in the directions orthogonal to  $S$  in a region containing  $\mathbf{x}(t)$  and  $\tilde{\mathbf{x}}(t)$ . To be precise, suppose

$$\|\rho \mathbf{f}(\tilde{\mathbf{x}}(t) + \rho_c^T \mathbf{v}(t), t) - \rho \mathbf{f}(\tilde{\mathbf{x}}(t), t)\| \leq \gamma \|\mathbf{v}\|$$

for all  $(\mathbf{v}, t) \in D \subset \mathbb{R}^{n-k} \times [0, T]$ , where the region  $D$  is such that the associated region  $\tilde{D} = \{(\tilde{\mathbf{x}}(t) + \rho_c^T \mathbf{v}, t) : (\mathbf{v}, t) \in D\} \subset \mathbb{R}^n \times [0, T]$  contains  $(\tilde{\mathbf{x}}(t), t)$  and  $(\mathbf{x}(t), t)$  for all  $t \in [0, T]$ . Let  $\mu \left( \rho \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\tilde{\mathbf{x}} + \rho^T \mathbf{z}, t) \rho^T \right) \leq \bar{\mu}$  for  $(\mathbf{z}, t) \in V \subset \mathbb{R}^k \times [0, T]$ , where the region  $V$  is such that it contains  $(\mathbf{u}(t), t)$  and  $(\mathbf{u}(t) + \mathbf{w}(t), t)$  for all  $t \in [0, T]$  and  $\mu$  denotes the logarithmic norm related to the 2-norm. Let  $\epsilon = \|\mathbf{e}_0\|_2$ . Then the error  $\mathbf{e}_i$  in the  $\infty$ -norm satisfies

$$\|\mathbf{e}_i\|_\infty \leq \epsilon \frac{\gamma}{\sqrt{2\bar{\mu}}} \sqrt{e^{2\bar{\mu}T} - 1},$$

and the 2-norm of the total error satisfies

$$\|\mathbf{e}\|_2 \leq \epsilon \sqrt{1 + \frac{\gamma^2}{4\bar{\mu}^2} (e^{2\bar{\mu}T} - 1 - 2\bar{\mu}T)}.$$

This is an important theorem that provides a quantitatively reasonable error estimate, and also explains the various factors that affect the error.

### **3.5 Summary**

In this chapter, we briefly reviewed methods that are used to reduce original models to more concise representations. To this end, the class of Krylov projection methods can be used, or the class of balancing methods. Both are fairly well developed for linear problems, but for nonlinear problems the techniques are still in their infancy. Of the methods proposed, proper orthogonal decomposition (POD) is the most popular and most frequently used method. In principle, it provides a representation for solutions, but via the Galerkin (or Petrov-Galerkin) projection it can also lead to reduced order models. This is the method of choice that will be used in subsequent chapters.

## Chapter 4

# Sensitivity Analysis

Sensitivity analysis is a key tool in the design of very large scale integrated circuits. It plays a role in evaluating the impact of the variability of design parameters on the performance, which is crucial knowledge in determining the next design iteration. In addition, sensitivity information is indispensable during automatic optimization of electronic circuits. In this chapter, we discuss the concept of sensitivity in more detail, and present commonly used methods for calculating it. The forward method can be used in the case of a small number of parameters, whereas the backward adjoint method has been designed specifically to deal with a multitude of parameter sensitivities.

### 4.1 Adjoint equations

Before we start the discussion about techniques for sensitivity analysis, it is important that we briefly discuss the notion of *adjoint equation*<sup>1</sup>. This is a crucial concept in obtaining more efficient methods for sensitivity analysis, especially in the case of a large number of parameters. The adjoint operator is a well known concept in functional analysis, and we refer to standard works on the subject for the precise meaning of

Consider first the linear ordinary differential equation

$$Ax' + Bx = f, \tag{4.1}$$

with continuous coefficients  $A, B : [0, T] \rightarrow L(\mathbb{R}^m)$ , with  $A$  nonsingular and where  $'$  denotes time differentiation. For ease of notation, we leave out the time dependence of

---

<sup>1</sup>The adjoint operator is a well known concept in functional analysis, and we refer to standard works on the subject for the precise meaning of it.

A, B. To obtain the adjoint equation, we first transform (4.1) into the form

$$\mathbf{x}' + \mathbf{A}^{-1}\mathbf{B}\mathbf{x} = \mathbf{A}^{-1}\mathbf{f},$$

which is possible due to our assumption on A. The adjoint corresponding to the latter equation is given by [20]

$$\mathbf{z}' - \mathbf{B}^T\mathbf{A}^{-T}\mathbf{z} = \mathbf{g}.$$

Now we set  $\mathbf{y} = \mathbf{A}^{-T}\mathbf{z}$ , and hence obtain the adjoint of the original equation (4.1):

$$(\mathbf{A}^T\mathbf{y})' - \mathbf{B}^T\mathbf{y} = \mathbf{g}. \quad (4.2)$$

If (4.1) is not an ODE but a differential algebraic equation, i.e. when the coefficient matrix A is singular, a different situation is encountered, and we need to take care when forming the adjoint equation. In this case, we follow the theory of März for (canonical) projectors (see [56] and also Section 6.4 for a more detailed discussion) and first construct a projector  $\mathbf{P} : [0, T] \rightarrow \mathbb{L}(\mathbb{R}^m)$  projecting along the null space of A. For example, we could choose  $\mathbf{P} = \mathbf{A}^+\mathbf{A}$ , where "+" indicates the Moore-Penrose inverse. Equivalently, we can choose a projector Q onto the null space, and define  $\mathbf{P} = \mathbf{I} - \mathbf{Q}$ .

Using the projector P, we can rewrite (4.1) into the form

$$\mathbf{A}(\mathbf{P}\mathbf{x})' + (\mathbf{B} - \mathbf{A}\mathbf{P}')\mathbf{x} = \mathbf{f}, \quad (4.3)$$

In [11], it is shown that this formulation is independent of the choice of projector P, and that, because of this, we can write it in the form (4.1).

Now following März, who has actually set up a whole theory of (canonical) projectors for differential algebraic equations of any index [56], we define:

$$\mathbf{B}_0 = \mathbf{B} - \mathbf{A}\mathbf{P}',$$

$$\mathbf{A}_1 = \mathbf{A} + \mathbf{B}_0\mathbf{Q}.$$

We can then decompose our problem into two parts, namely the regular ODE

$$\mathcal{L}_s\mathbf{P}\mathbf{y} = \mathbf{P}\mathbf{A}_1^{-1}\mathbf{f}, \quad (4.4)$$

and the algebraic equation

$$Q_s \mathbf{y} = Q A_1^{-1} \mathbf{f}, \quad (4.5)$$

where

$$\begin{aligned} \mathcal{L}_s \mathbf{z} &:= \mathbf{z}' + (P A_1^{-1} B_0 - P') \mathbf{z}, \\ Q_s \mathbf{z} &:= Q \mathbf{z} + Q A_1^{-1} B P \mathbf{z}. \end{aligned}$$

It is easily seen that  $Q_s^2 = Q_s$ , so that  $Q_s$  is a projector. The operator  $\mathcal{L}_s$  is well defined, and the corresponding initial value problem is uniquely solvable in  $C^1([0, T], \mathbb{R}^m)$ .

The adjoint of  $\mathcal{L}_s$  is given by

$$\mathcal{L}_s^* \mathbf{w} := \mathbf{w}' - (B_0^T A_1^{-T} P^T - P'^T) \mathbf{w} \quad (4.6)$$

for  $\mathbf{w} \in C^1([0, T], \mathbb{R}^m)$ . The corresponding initial value problem

$$\begin{aligned} \mathcal{L}_s^* \mathbf{w} &= \mathbf{h}, \\ \mathbf{w}(0) &= \mathbf{w}_0, \end{aligned}$$

is uniquely solvable in  $C^1([0, T], \mathbb{R}^m)$ . Furthermore, we have that

$$P^T \mathbf{w} \in C^1([0, T], \mathbb{R}^m),$$

and

$$\begin{aligned} (P^T \mathbf{w})' - (P^T B_0^T A_1^{-T} - P'^T) P^T \mathbf{w} &= P^T \mathbf{h}, \\ P^T \mathbf{w}(0) &= P^T \mathbf{w}_0. \end{aligned}$$

Defining then

$$P_{*s} := A_1^{-T} P^T A_1^T = A_1^{-T} A^T,$$

and

$$\mathbf{u} := A^+ P^T \mathbf{w},$$



a straightforward calculation shows that (see also [11])

$$P^T \mathbf{h} = [A^T(P_{*s} \mathbf{u})]' - B^T(P_{*s} \mathbf{u}).$$

The projector  $P_{*s}$  can be shown to be independent of the choice of projector  $P$ . Combining all these facts, the following conclusion can be drawn:

**Theorem 4.1** *Let  $A : [0, T] \rightarrow L(\mathbb{R}^m, \mathbb{R}^m)$ ,  $\dim(\text{Im}(A(t))) < m$  for all  $t \in [0, T]$ . Assume that the matrix pencil  $(A(t), B(t))$  is regular of index 1 for all  $t \in [0, T]$ , and that there exists a projector  $Q \in C^1([0, T], L(\mathbb{R}^m, \mathbb{R}^m))$  onto  $\ker(A)$ . Then the initial value problem involving the operator  $\mathcal{L}\mathbf{x} := A\mathbf{x}' + B\mathbf{x}$  is well defined and uniquely solvable, and the corresponding adjoint problem is given by  $\mathcal{L}^*\phi = \mathbf{s}$  where*

$$\mathcal{L}^*\phi := (A^T\phi)' - B^T\phi.$$

As can be seen in the foregoing, in this case the adjoint operator is what we expect it to be. For DAEs with a higher index, however, the adjoint problem may look quite different, and we may have to solve a so-called augmented set of equations. For a very detailed analysis, we refer to [10, 11]. This work has been generalized further by März in [57], where equations of the type

$$A(D\mathbf{x})' + B\mathbf{x} = \mathbf{q}, \tag{4.7}$$

are considered, with corresponding adjoint equation

$$D^T(A^T\mathbf{y})' - B^T\mathbf{y} = \mathbf{p}.$$

For nonlinear DAEs, the situation may be quite different. Therefore, we will give a detailed derivation of the corresponding adjoint problem in the special case of the electronic circuit equations. The equations in that case are of a form similar to (4.7):

$$A(\mathbf{q}(A^T\mathbf{x}(t)))' + \mathbf{b}(\mathbf{x}(t), t) = \mathbf{q}(\mathbf{x}(t), t);$$

see also [25, 38]. Before that, we will review some methods for finding sensitivities in the next two sections. As one will see, the notion of the adjoint operator plays an extremely important role in this.

## 4.2 Basic sensitivity analysis

The main goal of sensitivity analysis is to calculate the rate of change of the output variables of a system as a result of changes in a set of parameters of that system. There are two sensitivity types that are of interest, the state solution sensitivity to a parameter change and the sensitivity of an observable circuit function, the observation function sensitivity.

For illustration purposes, consider the general non-linear system of the form

$$\mathbf{F}(\mathbf{x}, \mathbf{p}) = \mathbf{0}, \quad (4.8)$$

where  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{p} \in \mathbb{R}^m$ ,  $\mathbf{F} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ . Assume that  $\frac{\partial \mathbf{F}}{\partial \mathbf{x}}$  is non-singular for all parameter settings  $\mathbf{p}$ . Then the state sensitivity amounts to calculating  $\frac{d\mathbf{x}}{d\mathbf{p}}$ , i.e. the sensitivity of the state variable  $\mathbf{x}$  with respect to the problem parameters  $\mathbf{p}$ . In many cases, however, there is an observable derived function  $\mathbf{G}(\mathbf{x}, \mathbf{p}) \in \mathbb{R}^k$  with  $k$  much smaller than  $m, n$ , and one is more interested in the behavior of this function with respect to parameter changes. Thus, one needs to calculate  $\frac{d\mathbf{G}}{d\mathbf{p}}$ , which can be done by writing

$$\frac{d\mathbf{G}}{d\mathbf{p}} = \frac{\partial \mathbf{G}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{p}} + \frac{\partial \mathbf{G}}{\partial \mathbf{p}}. \quad (4.9)$$

Basically, there are two methods to calculate both types of sensitivity: the forward method and the backward adjoint method. Let's first discuss the forward method, which is a direct approach to find the desired sensitivities. To this end, we linearize the system in (4.8) around some nominal parameter set:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{p}} + \frac{\partial \mathbf{F}}{\partial \mathbf{p}} = \mathbf{0}, \quad (4.10)$$

from which we can find the  $n \times m$  matrix  $\frac{d\mathbf{x}}{d\mathbf{p}}$  of state sensitivities due to the assumption of non-singularity of  $\frac{\partial \mathbf{F}}{\partial \mathbf{x}}$ . Note that, in order to find this matrix of sensitivities, we need to solve  $m$  linear systems, which is feasible if the number of parameters is small. The sensitivity of the observable function  $\mathbf{G}$  is then easily computed using the obtained state sensitivities.

Whenever the number of parameters is large, the forward method is often not feasible anymore and computationally too expensive, as too many linear systems need to be solved. In that case, the backward adjoint method provides a useful alternative as it avoids the direct calculation of the matrix  $\frac{d\mathbf{x}}{d\mathbf{p}}$ , and directly leads to the sensitivity of the observable function  $\mathbf{G}$ . It works as follows. First, we multiply the linearized equation (4.10) from the left by a new parameter  $\lambda \in \mathbb{R}^n$ , called Lagrangian multiplier, so as to obtain

$$\lambda^T \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{p}} + \lambda^T \frac{\partial \mathbf{F}}{\partial \mathbf{p}} = 0.$$

Now assume that  $\lambda$  is the solution of the linear system

$$\lambda^T \frac{\partial \mathbf{F}}{\partial \mathbf{x}} = -\frac{\partial \mathbf{F}}{\partial \mathbf{p}}. \quad (4.11)$$

Then we have

$$\frac{\partial \mathbf{G}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{p}} = -\lambda^\top \frac{\partial \mathbf{F}}{\partial \mathbf{p}},$$

and hence, using (4.9):

$$\frac{d\mathbf{G}}{d\mathbf{p}} = -\lambda^\top \frac{\partial \mathbf{F}}{\partial \mathbf{p}} + \frac{\partial \mathbf{G}}{\partial \mathbf{p}}.$$

Thus, the sensitivity of the observable function  $\mathbf{G}$  is now obtained by solving only one linear system! Hence, there is no dependence anymore on the number of parameters; having calculated the parameter  $\lambda$ , all desired sensitivities can be calculated. This explains why the adjoint method is especially attractive when the number of parameters in the problem is very large.

The conclusion is evident, without even having to go into a detailed operation count: the original method is best suited to find sensitivities of a possibly large number of state variables with respect to a small number of parameters, while the adjoint method is ideal for finding the sensitivity of a low-dimensional observable function of the state variables with respect to a large number of parameters. Note that the adjoint method will become more expensive if the number of right hand sides in (4.11) goes up; this is directly related to the dimension of the range of the function  $\mathbf{G}$ .

### 4.3 Sensitivity analysis for differential-algebraic systems

In the foregoing section, we discussed the basics of sensitivity analysis. In practice, one will often be confronted with a situation in which a time dependent system must be solved. This changes the sensitivity analysis, as we will need to solve time dependent systems rather than just linear systems as given in (4.11). In order to see how this changes our methods, let's consider the system

$$\mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}, t, \mathbf{p}) = 0, \quad (4.12)$$

where  $\dot{\mathbf{x}} = \frac{\partial \mathbf{x}}{\partial t}$ , with initial condition  $\mathbf{x}(0) = \mathbf{x}_0(\mathbf{p})$ .

An important issue before turning our attention to the sensitivity analysis in this case, is the matter of adjoints of differential-algebraic equations. This topic has been discussed in Section 4.1. As we have seen in that section, for a linear DAE without parameters,

$$A\dot{\mathbf{x}} + B\mathbf{x} = 0,$$

with sufficiently smooth matrix functions  $A$  and  $B$  (dependent on time), we obviously have that

$$\mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}, t, \mathbf{p}) = A\dot{\mathbf{x}} + B\mathbf{x},$$

and the adjoint DAE is given by

$$\frac{\partial}{\partial t} (A^\top \lambda) - B^\top \lambda = 0.$$

In [10,11], a detailed analysis is provided. The most interesting result, as far as our work described in the following chapters is concerned, is that Balla and März prove that for an index-1 DAE the adjoint system is also of index-1. This strengthens our ideas about similarities between the structure of the original DAE and the adjoint equation.

Another important remark in this context is that März, in a separate paper [57], has shown that for a certain class of differential algebraic systems, the operations of "adjoint" and "discretization" commute. This means that we can either first form the adjoint equation, then discretize it, or alternatively first discretize the original DAE and then form the discretized adjoint equations. A corollary is that the stability of the adjoint equation and its discretized form is maintained. Again, this is important material for our work in subsequent chapters.

We now return to the general nonlinear case as displayed in (4.12). The forward method works very much as described in the previous section, but now we need to take into account the time derivatives. Thus, we construct the linearized system (around some nominal parameter  $\mathbf{p}_0$ )

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{p}} + \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{p}} + \frac{\partial \mathbf{F}}{\partial \mathbf{p}} = 0,$$

and observe that ("Schwarz")

$$\frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{p}} = \partial \left( \frac{\partial \mathbf{x}}{\partial t} \right) / \partial \mathbf{p} = \frac{\partial}{\partial t} \frac{\partial \mathbf{x}}{\partial \mathbf{p}}.$$

Hence, the linearized system takes the form

$$\left( \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \frac{\partial}{\partial t} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right) \frac{\partial \mathbf{x}}{\partial \mathbf{p}} + \frac{\partial \mathbf{F}}{\partial \mathbf{p}} = 0,$$

providing a differential-algebraic system for the state sensitivity  $\frac{\partial \mathbf{x}}{\partial \mathbf{p}}$ . Its initial condition is provided by

$$\frac{\partial \mathbf{x}}{\partial \mathbf{p}}(0) = \frac{\partial \mathbf{x}_0(\mathbf{p})}{\partial \mathbf{p}}.$$

Thus, the forward method consists of solving a linear differential-algebraic system for the state sensitivities.

Now we turn our attention to the construction of the backward adjoint method. This is, however, much more complicated than in the previous section. We follow the approach as described in detail in [17], where an adjoint approach is presented for the sensitivity of the derived function

$$\mathbf{G}(\mathbf{x}, \mathbf{p}) \equiv \int_0^T \mathbf{g}(\mathbf{x}, t, \mathbf{p}) dt.$$

First, a Lagrange multiplier  $\lambda$  is introduced and the augmented objective function is formed:

$$\mathbf{I}(\mathbf{x}, \mathbf{p}) \equiv \mathbf{G}(\mathbf{x}, \mathbf{p}) - \int_0^T \lambda^T(t) \mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}, t, \mathbf{p}) dt.$$

Since we know that  $\mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}, t, \mathbf{p}) = 0$ , we find that the sensitivity of  $\mathbf{G}$  with respect to  $\mathbf{p}$  is the same as the sensitivity of  $\mathbf{I}$ . Hence, using the definition of  $\mathbf{G}$ ,

$$\frac{d\mathbf{G}}{d\mathbf{p}} = \frac{d\mathbf{I}}{d\mathbf{p}} = \int_0^T \left( \frac{\partial \mathbf{g}}{\partial \mathbf{p}} + \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{p}} \right) dt - \int_0^T \lambda^\top \left( \frac{\partial \mathbf{F}}{\partial \mathbf{p}} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{p}} + \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{p}} \right) dt. \quad (4.13)$$

Now we use integration by parts on the last term in this expression:

$$\int_0^T \lambda^\top \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{p}} dt = \left[ \lambda^\top \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \frac{\partial \mathbf{x}}{\partial \mathbf{p}} \right]_0^T - \int_0^T \frac{d}{dt} \left( \lambda^\top \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \right) \frac{\partial \mathbf{x}}{\partial \mathbf{p}} dt.$$

Substituting this into (4.13), we find

$$\frac{d\mathbf{G}}{d\mathbf{p}} = \int_0^T \left( \frac{\partial \mathbf{g}}{\partial \mathbf{p}} - \lambda^\top \frac{\partial \mathbf{F}}{\partial \mathbf{p}} \right) dt - \int_0^T \left[ -\frac{\partial \mathbf{g}}{\partial \mathbf{x}} + \lambda^\top \frac{\partial \mathbf{F}}{\partial \mathbf{x}} - \frac{d}{dt} \left( \lambda^\top \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \right) \right] \frac{\partial \mathbf{x}}{\partial \mathbf{p}} dt - \left[ \lambda^\top \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \frac{\partial \mathbf{x}}{\partial \mathbf{p}} \right]_0^T.$$

Setting the argument in the middle integral equal to zero, i.e. requiring that

$$\frac{d}{dt} \left( \lambda^\top \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \right) - \lambda^\top \frac{\partial \mathbf{F}}{\partial \mathbf{x}} = -\frac{\partial \mathbf{g}}{\partial \mathbf{x}}, \quad (4.14)$$

we find that

$$\frac{d\mathbf{G}}{d\mathbf{p}} = \int_0^T \left( \frac{\partial \mathbf{g}}{\partial \mathbf{p}} - \lambda^\top \frac{\partial \mathbf{F}}{\partial \mathbf{p}} \right) dt - \left[ \lambda^\top \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \frac{\partial \mathbf{x}}{\partial \mathbf{p}} \right]_0^T. \quad (4.15)$$

Note that we have

$$\left[ \lambda^\top \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \frac{\partial \mathbf{x}}{\partial \mathbf{p}} \right]_0^T = \lambda^\top \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \frac{\partial \mathbf{x}}{\partial \mathbf{p}}(T) - \lambda^\top \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \frac{\partial \mathbf{x}}{\partial \mathbf{p}}(0) = \left( \lambda^\top \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \right)(T) \frac{\partial \mathbf{x}}{\partial \mathbf{p}}(T) - \left( \lambda^\top \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \right)(0) \frac{\partial \mathbf{x}_0}{\partial \mathbf{p}}.$$

The final term in this expression is easily obtained, as it is the sensitivity of the initial condition with respect to the parameters. At  $t = T$ , the situation is more complicated, as this will provide the initial condition for the adjoint sensitivity calculation. This can be rather tricky for higher index systems, as is explained in [17]. Here we will restrict to index-0 and index-1 problems (assuming the conditions of Tischendorf [79, 80] are satisfied), and then we can use the initial condition

$$\left( \lambda^\top \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \right)(T) = 0. \quad (4.16)$$

Then we find that the sensitivity of  $\mathbf{G}$  is given by

$$\frac{d\mathbf{G}}{d\mathbf{p}} = \int_0^T \left( \frac{\partial \mathbf{g}}{\partial \mathbf{p}} - \lambda^\top \frac{\partial \mathbf{F}}{\partial \mathbf{p}} \right) dt + \left( \lambda^\top \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} \right)(0) \frac{\partial \mathbf{x}_0}{\partial \mathbf{p}}. \quad (4.17)$$

Hence, summarizing the calculation of the sensitivity of  $\mathbf{G}$  with respect to the parameters  $\mathbf{p}$ :

1. solve the forward DAE in (4.12) for  $\mathbf{x}$

2. solve the adjoint problem (4.14), subject to the initial condition (4.16) at  $t = T$ , for  $\lambda$
3. find the sensitivity of  $\mathbf{G}$  using the expression in (4.17)

The second step explains why one usually refers to the method as the backward adjoint method.

## 4.4 Sensitivities for electronic circuits

Without a deeper insight into the response of circuits, their states and intended functions, to small physical changes, a new circuit implementation created without the benefit of an experienced designer may be open to volatile responses, chip to chip, to the smallest parameter variation in the manufacturing step. Sensitivity Analysis is the study of the reaction of a system or model to the change of the parameters it contains. A measure of the sensitivity of a given state or circuit function to a list of parameters is revealed once a system has been analyzed, there is an instantly useful set of data that can be used to confirm a good circuit design or identify a subcircuit that needs better care or implementation. All of this circuit analysis can take place before the fabrication process which helps to keep the yield loss to a minimum.

The calculation cost of circuit sensitivities varies with the size of a circuit, for larger circuits one can expect a proportionally larger calculation cost. In this section we will introduce the different types of circuit sensitivities, their derivation and also the basic building blocks, following mostly the discussion in the previous sections. We will also cover the existing methods for the calculation of circuit sensitivities, the direct forward brute force approach and a cheaper Lagrangian method based on a backward adjoint approach.

The most basic sensitivity measure is taken by observing the reaction of a single circuit state, for example the voltage drop between two resistors, to a change in the parameters of the surrounding components. More complex sensitivity measurements can be taken by constructing more complex sensitivity expressions upon the basic state sensitivities, these are used to measure the sensitivity of more complex circuit functions, such as potential dividing or amplifier gain, which are a combination of the basic circuit state values.

Circuit parameters are intrinsic to the circuit model, the characteristics of each individual circuit component depends upon them and also the performance of the total circuit. This is reflected in the circuit equations, a DAE model of a more specialized form than the general form given in (4.12):

$$\frac{d}{dt} [\mathbf{q}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})] + \mathbf{j}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) = \mathbf{s}(t, \mathbf{p}). \quad (4.18)$$

This equation reveals the dependence of the circuit states, the charge and current vector behavior and the signal source upon the parameter vector  $\mathbf{p}$ , which contains the list of circuit parameters. Here the state solution  $\mathbf{x}(t, \mathbf{p})$  is a function of time and the parameter vector  $\mathbf{p}$ ; the charge  $\mathbf{q}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})$  and current  $\mathbf{j}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})$  behavior of the system are dependent on the state vector and also themselves depend upon the parameter vector.

An imperfection in the manufacture stage, unless extreme, will not change the circuit topology and therefore not change the equations modeling the circuit. However, the resulting change in parameter values for the affected components will be registered in the circuit model (4.18) as a change in a parameter value in the vector  $\mathbf{p}$ . It is clear from the circuit dependency relationships mentioned, and just by referring to the circuit equation model, that any change inside this list of parameter values has the potential to completely change the circuit state solutions and push the circuit outside its original design tolerances or even modify the behavior of the circuit beyond its intended function.

A pronounced example is found in the area of fault analysis for analog circuits. Faults are modeled as resistors that bridge a pair of nodes which are not connected in the original circuit. For some pairs of nodes, such resistive bridges may cause the transient solution to deviate dramatically from the solution of the original circuit. Hence, when appending many bridges to the original circuit between all possible pairs of nodes, a large number of parameters (the resistive values of the bridges) is introduced, some causing no deviation from the original solution, others deviating a lot. Such analyses enable designers to quickly trace the location of faults. Sensitivity analysis is currently being explored within the electronics industry as a means of obtaining such quick insights. Note that this specific example implies a change of topology, deviating from the original circuit.

Before going into the specific methods for sensitivity analysis of electronic circuits in the next section, we briefly discuss some aspects in the following subsections that are important in dealing with parameters in practical situations.

#### 4.4.1 State sensitivity

The circuit state vector  $\mathbf{x}(t, \mathbf{p})$  is dependent upon time,  $t$ , and the parameter vector  $\mathbf{p}$ . Each individual circuit state listed in the state vector could have a dependence on none, one or more of the parameter values listed in the parameter vector, this is dependent upon the type of components adjacent to the circuit states and the choice taken over which parameters to observe. A resistor is dependent upon a number of parameters, its length  $l$ , its electrical resistivity  $\rho$  and its cross-sectional area  $A = w \cdot d = \text{width} \cdot \text{depth}$ :

$$\mathbf{R} = \frac{l \cdot \rho}{A}. \quad (4.19)$$

A voltage state in the node between two resistors will have a dependence upon each of the parameters contained in (4.19). If a capacitor is then connected to this node, of capacitance

$$\mathbf{C} = \epsilon_r \epsilon_0 \frac{\mathbf{A}}{d}, \quad (4.20)$$

then the voltage state of this node suddenly has a whole new set of parameters that could influence its value. Again, a choice has to be made on which parameters should be entered into the parameter vector.

As stated in the general case in Section 4.1, the state sensitivity of an individual circuit state to a particular parameter is then defined as the differential of that state with respect to a parameter in the list. The state sensitivity matrix  $\hat{\mathbf{x}}(\mathbf{t}, \mathbf{p})$  contains the sensitivity values of each state with respect to each individual parameter:

$$\hat{\mathbf{x}}(\mathbf{t}, \mathbf{p}) \equiv \frac{\partial \mathbf{x}(\mathbf{t}, \mathbf{p})}{\partial \mathbf{p}} \in \mathbf{R}^{n \times m} \quad (4.21)$$

The state sensitivity result is a matrix of size  $n \times m$ , where  $n$  is the number of system states and  $m$  the number of parameters under investigation. If only one parameter is under investigation then a column vector of equal size to the state vector is returned as the result. Thus,

$$\hat{\mathbf{x}}(\mathbf{t}, \mathbf{p}) \equiv \begin{pmatrix} \frac{\partial x_1}{\partial p_1} & \cdots & \frac{\partial x_1}{\partial p_m} \\ \vdots & & \vdots \\ \frac{\partial x_n}{\partial p_1} & \cdots & \frac{\partial x_n}{\partial p_m} \end{pmatrix}. \quad (4.22)$$

### Chain rule access to parameters

There exist a very large number of components and there are also many model variations of any type of component, each having their own individual parameter lists. Bearing this in mind one can expect that for large circuits a comprehensive parameter sensitivity analysis can quickly create very large parameter vectors and system matrices. Using the chain rule one can dramatically cut down the sizes of vectors and matrices involved, it is possible to just calculate the sensitivity of state values to changes in the component values and then later retrieve individual model parameter values.

For example, if one calculates the sensitivity of a state with respect to a resistor value,  $\frac{\partial x}{\partial R}$ , one could then extract specific parameter sensitivities at a later point on the fly by using the chain rule:



$$\frac{\partial \mathbf{x}}{\partial l} = \frac{\partial \mathbf{x}}{\partial \mathbf{R}} \frac{\partial \mathbf{R}}{\partial l}, \quad (4.23)$$

$$\frac{\partial \mathbf{x}}{\partial w} = \frac{\partial \mathbf{x}}{\partial \mathbf{R}} \frac{\partial \mathbf{R}}{\partial w}, \quad (4.24)$$

$$\frac{\partial \mathbf{x}}{\partial \rho} = \frac{\partial \mathbf{x}}{\partial \mathbf{R}} \frac{\partial \mathbf{R}}{\partial \rho}. \quad (4.25)$$

Equation (4.23) shows how the chain rule is applied to calculate the sensitivity of a state to the length parameter within a particular resistor expression, (4.25) for the resistivity parameter and (4.24) for the width parameter. This approach can be taken when extracting parameter sensitivities for any other component model, a chain rule example for a capacitor model parameter is shown below:

$$\frac{\partial \mathbf{x}}{\partial \mathbf{d}} = \frac{\partial \mathbf{x}}{\partial \mathbf{C}} \frac{\partial \mathbf{C}}{\partial \mathbf{d}}$$

The conclusion is that one should take care in defining parameters for which sensitivities need to be calculated. Any dependence between parameters should be avoided, and the list of parameters should be minimal. Ideally, one should reduce the list of parameters to a set of basic parameters that can be used to describe all desired sensitivities in the end. In what follows, this aspect will not be reconsidered, we will always assume that a set of parameters is given and that all sensitivities with respect to these parameters must be calculated.

#### 4.4.2 Observation function sensitivity

It is often required to observe the combined effect or meaning of each individual circuit state, in other words the realization of a circuit design solution, and the effect of the sensitivity of the overall design to a change in circuit parameters. A frequently occurring concern in the electronics industry is the energy consumption of their devices, circuits, subcircuit and individual component efficiency. A well known standard expression for the power consumption of a resistor is power = current  $\times$  voltage and after applying Ohm's law, we have power = (voltage)<sup>2</sup>/resistance. We can express this as a function of the state vector  $\mathbf{x}(t, \mathbf{p})$ , as follows<sup>2</sup>:

<sup>2</sup>We henceforth change our notation, and use  $G_{\text{obs}}$  for the observable function, which in turn is a function of some  $\mathbf{F}$  that depends on the solution  $\mathbf{x}$  and the parameters  $\mathbf{p}$

$$\mathbf{F}(\mathbf{x}(t, \mathbf{p})) = \frac{1}{R} f^2(\mathbf{x}(t, \mathbf{p})), \quad (4.26)$$

where  $f(\mathbf{x}(t, \mathbf{p}))$  selects voltage states and finds the potential difference across a specific resistor with value  $R$ . The total power used by the resistor over a period of time, the energy use, is found by integrating (4.26) over that same time period:

$$\mathbf{G}_{\text{obs}}(\mathbf{x}, \mathbf{p}) \equiv \int_0^T \mathbf{F}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) dt. \quad (4.27)$$

Equation (4.27) is an observation function that observes the energy use over a selected subcircuit area. The calculation and prediction of the sensitivity of energy consumption to minor parameter changes is invaluable, electronic components are expected to have a stable power supply.

The observation function sensitivity is calculated by differentiating (4.27) with respect to the parameter vector, giving (around some nominal parameter)

$$\frac{d\mathbf{G}_{\text{obs}}}{d\mathbf{p}} = \int_0^T \left( \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial \mathbf{p}} + \frac{\partial \mathbf{F}}{\partial \mathbf{p}} \right) dt, \quad (4.28)$$

or, using the previously introduced notation  $\hat{\mathbf{x}} = \frac{\partial \mathbf{x}}{\partial \mathbf{p}}$ ,

$$\frac{d\mathbf{G}_{\text{obs}}}{d\mathbf{p}} = \int_0^T \left( \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \cdot \hat{\mathbf{x}} + \frac{\partial \mathbf{F}}{\partial \mathbf{p}} \right) dt. \quad (4.29)$$

Note that the state sensitivity expression is present in (4.28), this is stated more clearly in (4.29) and confirms that the more complicated observation functions are built upon and are dependent upon the state sensitivity vector. This dependence on the state sensitivity vector introduces a constant initial cost, before any observation function sensitivities can be evaluated, the state sensitivities must first all be calculated. This is entirely analogous to what we explained in Section 4.2 for the general abstract case.

#### A simple illustration of the backward adjoint idea

The adjoint method introduced in Section 4.2 may look like a mathematical trick. However, if we consider a simple linear circuit, the idea of using the adjoint comes out very

naturally. To demonstrate this, consider a linear circuit that can be described (for example by employing the MNA method) by (note that, in this case, no dynamics is involved)

$$\mathbf{A}\mathbf{x} = \mathbf{b}.$$

Here, the vector  $\mathbf{x}$  contains node voltages and branch currents. If the system contains parameters, denoted by the vector  $\mathbf{p}$ , we find that

$$\mathbf{A} \frac{\partial \mathbf{x}}{\partial \mathbf{p}} + \frac{\partial \mathbf{A}}{\partial \mathbf{p}} = \frac{\partial \mathbf{b}}{\partial \mathbf{p}}.$$

Hence, we find that

$$\frac{\partial \mathbf{x}}{\partial \mathbf{p}} = -\mathbf{A}^{-1} \left( \frac{\partial \mathbf{A}}{\partial \mathbf{p}} - \frac{\partial \mathbf{b}}{\partial \mathbf{p}} \right).$$

Very often, however, the output function is a linear combination of the components of  $\mathbf{x}$ , i.e.

$$\phi = \mathbf{d}^T \mathbf{x}$$

for some vector  $\mathbf{d}$ . This is clearly a function of the solution  $\mathbf{x}$ . We now wish to find

$$\frac{\partial \phi}{\partial \mathbf{p}} = \mathbf{d}^T \frac{\partial \mathbf{x}}{\partial \mathbf{p}} = -\mathbf{d}^T \mathbf{A}^{-1} \left( \frac{\partial \mathbf{A}}{\partial \mathbf{p}} - \frac{\partial \mathbf{b}}{\partial \mathbf{p}} \right).$$

From this expression we see that it would be convenient to calculate a vector  $\mathbf{z}$  such that

$$\mathbf{z}^T = \mathbf{d}^T \mathbf{A}^{-1},$$

implying that

$$\mathbf{z} = \mathbf{A}^{-T} \mathbf{d}.$$

If we have the LU-decomposition of  $\mathbf{A}$  available, then

$$\mathbf{z} = \mathbf{L}^{-T} \mathbf{U}^{-T} \mathbf{d},$$

and having found this vector  $\mathbf{z}$ , the sensitivity is easily found as

$$\frac{\partial \phi}{\partial \mathbf{p}} = -\mathbf{z}^T \left( \frac{\partial \mathbf{A}}{\partial \mathbf{p}} - \frac{\partial \mathbf{b}}{\partial \mathbf{p}} \right).$$

Thus, the occurrence of the adjoint operator is quite natural in the context of sensitivities for observed quantities, and should not come as a surprise.

## 4.5 Applying the direct forward method to circuit sensitivity

The direct forward method for calculating the state sensitivities reuses information that is already available from an initial circuit simulation. However, it depends on how

much time we wish to invest in calculating the actual sensitivities. We can use the method based upon expression (4.10) or its dynamic counterpart, but to do this we must calculate exactly the matrix  $\partial\mathbf{F}/\partial\mathbf{x}$  and the right hand side vector/matrix  $\partial\mathbf{F}/\partial\mathbf{p}$ . This may be quite a cumbersome task, especially since we must do this at all time steps as described in Section 4.2. For this reason, one often resorts to a numerical approach. Then, the main requirement is the circuit state solutions, which can be obtained, for example, by using an Euler-backward time integration of the circuit equations,

$$\frac{1}{\Delta t}[\mathbf{q}^{n+1} - \mathbf{q}^n] + \mathbf{j}^{n+1} - \mathbf{s}^{n+1} = 0, \quad (4.30)$$

where the quantities at  $n$  and  $n+1$  are discrete approximations of the continuous quantities at the time points  $t^n$  and  $t^{n+1}$ , respectively. A Newton-Raphson solution of (4.30) involves the coefficient matrix  $\mathbf{Y} = \frac{1}{\Delta t}\mathbf{C} + \mathbf{G}$ , where  $\mathbf{C} = \partial\mathbf{q}/\partial\mathbf{x}$  and  $\mathbf{G} = \partial\mathbf{j}/\partial\mathbf{x}$ . The LU-decomposition of the matrix  $\mathbf{Y} = \mathbf{L}\mathbf{U}$  carried out while solving for the state solutions to this circuit system should also be saved as it can also be reused in the calculation of the state sensitivities.

### Calculation of state sensitivities

The direct forward method is so called because a direct forward differentiation is carried out on the circuit equations as part of the method. Here we first differentiate (4.30) with respect to the parameter vector  $\mathbf{p}$ ,

$$0 = \frac{1}{\Delta t} \left[ \frac{\partial\mathbf{q}^{n+1}}{\partial\mathbf{p}} - \frac{\partial\mathbf{q}^n}{\partial\mathbf{p}} \right] + \frac{1}{\Delta t} [\mathbf{C}\hat{\mathbf{x}}^{n+1} - \mathbf{C}\hat{\mathbf{x}}^n] + \mathbf{G}\hat{\mathbf{x}}^{n+1} + \frac{\partial\mathbf{j}^{n+1}}{\partial\mathbf{p}} - \frac{\partial\mathbf{s}^{n+1}}{\partial\mathbf{p}},$$

which, after applying a re-arrangement of terms, leads to

$$\underbrace{((1/\Delta t)\mathbf{C} + \mathbf{G})}_{\mathbf{Y}} \hat{\mathbf{x}}^{n+1} = - \underbrace{\frac{1}{\Delta t} \left[ \frac{\partial\mathbf{q}^{n+1}}{\partial\mathbf{p}} - \frac{\partial\mathbf{q}^n}{\partial\mathbf{p}} \right] + (1/\Delta t)\mathbf{C}\hat{\mathbf{x}}^n - \frac{\partial\mathbf{j}}{\partial\mathbf{p}} + \frac{\partial\mathbf{s}}{\partial\mathbf{p}}}_{\mathbf{f}}.$$

This is an equation for the state sensitivity where  $\hat{\mathbf{x}}^{n+1}(\mathbf{p}) \approx \hat{\mathbf{x}}(t_{n+1}, \mathbf{p})$ .

Once the initial DC value  $\hat{\mathbf{x}}^0$  is calculated the only unknown is the state sensitivity  $\hat{\mathbf{x}}^{n+1}$ . The initial sensitivity of  $\hat{\mathbf{x}}_{\text{DC}}(\mathbf{p})$  with respect to  $\mathbf{p}$  is similar to  $\hat{\mathbf{x}}(t, \mathbf{p})$ . Reusing the solutions and the LU-decomposition from (4.30), the sensitivity  $\hat{\mathbf{x}}^{n+1}(\mathbf{p}) \approx \hat{\mathbf{x}}(t_{n+1}, \mathbf{p})$  may be calculated by recursion [22, 40]

$$\hat{\mathbf{x}}^{n+1}(\mathbf{p}) = \mathbf{Y}^{-1}\mathbf{f}, \text{ in which} \quad (4.31)$$

$$\mathbf{f} = -\frac{1}{\Delta t} \left[ \frac{\partial \mathbf{q}^{n+1}}{\partial \mathbf{p}} - \frac{\partial \mathbf{q}^n}{\partial \mathbf{p}} \right] - \frac{\partial \mathbf{j}^{n+1}}{\partial \mathbf{p}} + \frac{\partial \mathbf{s}^{n+1}}{\partial \mathbf{p}} + \frac{1}{\Delta t} \mathbf{C}\hat{\mathbf{x}}^n(\mathbf{p}). \quad (4.32)$$

The last term in vector  $\mathbf{f}$  requires  $\mathcal{O}(\text{PN}^2)$  operations, in addition each other term needs  $\mathcal{O}(\text{PN})$  evaluations. Solving the system requires an additional  $\mathcal{O}(\text{PN}^2)$  operations.

One final remark is that, from the point of view of reducing use of storage, it is optimal to calculate the sensitivity immediately after the Newton iterative process for the solution at time  $t_{n+1}$  has converged. The LU-decomposition needed is then still available (assuming a direct solution is employed for the linear systems), and it is a simple exercise to calculate the desired sensitivities.

### Function sensitivities

Once an expression for the observation function sensitivity has been derived, in our case we previously derived a function observing the sensitivity of the circuit power usage (4.33). It is evaluated while substituting the state sensitivity values calculated for the time period being observed. The initial cost of evaluating the state sensitivities is not avoided, but if more than one observation function is formulated these state sensitivity values can be reused with the restriction that one must only observe the next observation function in the same time period with the same parameter and component values, which is quite likely.

As an example, one may want to test the sensitivity of the amplification function (or gain) of an amplifier, and for the same time period and parameter values, test the individual component power performance or even just the total power consumption. Until the need arises to change the test signals, parameter values and the time frames, this reuse does maximize the productivity of the initial cost, but frequently it is desired to test a range of conditions and parameters to fine tune a circuit design. If one wanted to modify the amplification of an amplifier, the parameters would change and one would then need to recalculate all of the state sensitivities before re-evaluating all of the desired observations again. For a design life cycle, this constant re-evaluation becomes very expensive.

If we take a closer look at the sensitivity of each observation function,

$$\frac{d\mathbf{G}_{\text{obs}}}{d\mathbf{p}} = \int_0^T \left( \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \hat{\mathbf{x}} + \frac{\partial \mathbf{F}}{\partial \mathbf{p}} \right) dt, \quad (4.33)$$

we find that if the only step taken is the substitution of  $\hat{\mathbf{x}}$  and the cost of determining  $\frac{\partial \mathbf{F}}{\partial \mathbf{x}}$  is cheap, then the main concern for calculating observation function sensitivities, as in

(4.33), is the efficient initial calculation of  $\hat{\mathbf{x}}(t, \mathbf{p})$ , or even the efficient calculation of the matrix-vector product  $\frac{\partial \mathbf{F}}{\partial \mathbf{x}} \hat{\mathbf{x}}$ .

Note that, from (4.31), we can derive an expression to judge the cost of this inner product using the direct forward method:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} \hat{\mathbf{x}} = \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \mathbf{Y}^{-1} \mathbf{f} = [\mathbf{Y}^{-T} \left[ \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right]^T]^T \mathbf{f} \quad (4.34)$$

This inner product can be calculated in  $\mathcal{O}(\min(F, m)n^2 + Fmn)$  operations.

For challenging industrial problems where the number of parameters in a model is large, and when more than one function is observed, this brute force approach becomes very expensive very quickly. Especially as this operation takes place at each time point during integration.

## 4.6 Application of the backward adjoint method to circuit sensitivity

By applying a Lagrangian multiplier approach, the method of finding sensitivities reformulates the observation function sensitivity equation into a form which is faster to evaluate. No extra work is needed to obtain the state sensitivities  $\hat{\mathbf{x}}$  found in (4.33), as these are implicitly present in the new formulation. We have developed this concept in the general abstract case in Section 4.2, and will now apply it to the specific case of the circuit equations.

### Calculation of function sensitivities

In the previous section we encountered the product  $\frac{\partial \mathbf{F}}{\partial \mathbf{x}} \hat{\mathbf{x}}$  and concluded that its cost is quite high; this is mainly caused by the high cost of evaluating the state sensitivities. Using an approach utilizing Lagrangian multipliers the Backward Adjoint Method [17] can be used to eliminate this product, or better put it replaced that product with a cheaper expression.

The starting point is the circuit equation, which is multiplied by  $\lambda(t)$  and then integrated. These steps are detailed in the following. The circuit equation reads:

$$\frac{d}{dt} [\mathbf{q}(\mathbf{x}, \mathbf{p})] + \mathbf{j}(\mathbf{x}, \mathbf{p}) - \mathbf{s}(t, \mathbf{p}) = 0.$$

The first step is now to differentiate this equation with respect to  $\mathbf{p}$  and then integrate with respect to time over a specified period. Using the chain rule, we easily find (remember that  $\mathbf{G}$  and  $\mathbf{C}$  are functions of  $\mathbf{x}$ )

$$\frac{d}{d\mathbf{p}}\mathbf{j}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) = \frac{\partial \mathbf{j}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{p}} + \frac{\partial \mathbf{j}}{\partial \mathbf{p}} = \mathbf{G} \frac{d\mathbf{x}}{d\mathbf{p}} + \frac{\partial \mathbf{j}}{\partial \mathbf{p}},$$

$$\frac{d}{d\mathbf{p}}\mathbf{q}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) = \frac{\partial \mathbf{q}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{p}} + \frac{\partial \mathbf{q}}{\partial \mathbf{p}} = \mathbf{C} \frac{d\mathbf{x}}{d\mathbf{p}} + \frac{\partial \mathbf{q}}{\partial \mathbf{p}},$$

where, as is commonly denoted,  $\mathbf{G} = \frac{\partial \mathbf{j}}{\partial \mathbf{x}}$  and  $\mathbf{C} = \frac{\partial \mathbf{q}}{\partial \mathbf{x}}$ . Now multiplying by  $\lambda^T(t)$  and integrating gives

$$\begin{aligned} 0 &= \int_0^T \lambda^T(t) \left[ \frac{d}{dt} \frac{d\mathbf{q}}{d\mathbf{p}} + \frac{d\mathbf{j}}{d\mathbf{p}} - \frac{d\mathbf{s}}{d\mathbf{p}} \right] dt, \\ 0 &= \int_0^T \lambda^T(t) \left[ \frac{d}{dt} \frac{d\mathbf{q}}{d\mathbf{p}} \right] dt + \int_0^T \lambda^T(t) \left[ \frac{d\mathbf{j}}{d\mathbf{p}} - \frac{d\mathbf{s}}{d\mathbf{p}} \right] dt. \end{aligned}$$

Using integration by parts, we find for the first integral that

$$\int_0^T \lambda^T(t) \left[ \frac{d}{dt} \frac{d\mathbf{q}}{d\mathbf{p}} \right] dt = \left[ \lambda^T(t) \frac{d\mathbf{q}}{d\mathbf{p}} \right]_0^T - \int_0^T \frac{d\lambda^T(t)}{dt} \frac{d\mathbf{q}}{d\mathbf{p}} dt. \quad (4.35)$$

Recombining this with the second integral, we have

$$0 = \left[ \lambda^T(t) \frac{d}{d\mathbf{p}} \mathbf{q}(\mathbf{x}(t, \mathbf{p})) \right]_0^T + \int_0^T \left[ - \left( \frac{d\lambda^T}{dt} \mathbf{C} - \lambda^T \mathbf{G} \right) \hat{\mathbf{x}} - \frac{d\lambda^T}{dt} \frac{\partial \mathbf{q}}{\partial \mathbf{p}} + \lambda^T \left( \frac{\partial \mathbf{j}}{\partial \mathbf{p}} - \frac{\partial \mathbf{s}}{\partial \mathbf{p}} \right) \right] dt.$$

The state sensitivity  $\hat{\mathbf{x}}$  is clearly part of this equation, but it will be seen soon that we can avoid having to evaluate it explicitly. To this end, we rearrange the foregoing equation in order to isolate the state sensitivity:

$$\int_0^T \left[ \left( \frac{d\lambda^T}{dt} \mathbf{C} - \lambda^T \mathbf{G} \right) \hat{\mathbf{x}} \right] = \left[ \lambda^T(t) \frac{d\mathbf{q}}{d\mathbf{p}} \right]_0^T + \int_0^T \left[ -\frac{d\lambda^T}{dt} \frac{\partial \mathbf{q}}{\partial \mathbf{p}} + \lambda^T \left( \frac{\partial \mathbf{j}}{\partial \mathbf{p}} - \frac{\partial \mathbf{s}}{\partial \mathbf{p}} \right) \right] dt \quad (4.36)$$

We can now begin to eliminate the costly state sensitivity calculation. Notice that (4.33) and (4.36) both have a term containing  $\hat{\mathbf{x}}$ , we can eliminate the costly matrix-vector product  $\frac{\partial \mathbf{F}}{\partial \mathbf{x}} \hat{\mathbf{x}}$  in (4.33) by using the following so-called *Backward Adjoint Equation*:

$$\mathbf{C}^T(\mathbf{p})\dot{\lambda}(t) - \mathbf{G}^T(\mathbf{p})\lambda(t) = - \left( \frac{\partial \mathbf{F}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})}{\partial \mathbf{x}(t, \mathbf{p})} \right)^T. \quad (4.37)$$

After the substitution step using (4.37) our original observation function (4.33) now takes the following form,

$$\begin{aligned} \frac{d}{d\mathbf{p}} \mathbf{G}_{\text{obs}} &= \int_0^T \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \hat{\mathbf{x}} + \frac{\partial \mathbf{F}}{\partial \mathbf{p}} dt \\ &= \int_0^T \left[ -\frac{d\lambda^T}{dt} \mathbf{C} + \lambda^T \mathbf{G} \right] \hat{\mathbf{x}} + \frac{\partial \mathbf{F}}{\partial \mathbf{p}} dt \\ &= -\left[ \lambda^T(t) \frac{d\mathbf{q}}{d\mathbf{p}} \right]_{t=0}^T + \int_0^T \frac{d\lambda^T}{dt} \frac{\partial \mathbf{q}}{\partial \mathbf{p}} - \lambda^T \left( \frac{\partial \mathbf{j}}{\partial \mathbf{p}} - \frac{\partial \mathbf{s}}{\partial \mathbf{p}} \right) + \frac{\partial \mathbf{F}}{\partial \mathbf{p}} dt. \end{aligned} \quad (4.38)$$

We now have an integral term and a boundary term to calculate, the  $\hat{\mathbf{x}}_{\text{DC}}$  value (at  $t = 0$ ) is already known and easy to find but the calculation of the boundary value at  $t = T$  would mean calculating all intermediate values and so the new found efficiency would be lost.

A good choice of initial values for the solution of the equations for  $\lambda$  eliminates this problem, we choose,

$$\lambda(T) = 0. \quad (4.39)$$

This then immediately explains the reason why we have a backward equation: we cannot specify the initial condition at  $t = 0$ , but only a condition at  $t = T$ . Hence, we need



to integrate the adjoint equation in a backward way. Going back to our example in Subsection 4.4.2, this is similar to the use of an LU decomposition, solving both a forward and a backward system.

We can now solve for correct values of  $\lambda$ , and our expression for the observation function sensitivities simplifies further to

$$\begin{aligned} \frac{d}{d\mathbf{p}} \mathbf{G}_{\text{obs}}(\mathbf{x}(\mathbf{p}), \mathbf{p}) &= \lambda^T(0) \left[ \frac{\partial \mathbf{q}}{\partial \mathbf{x}}(0) \hat{\mathbf{x}}_{\text{DC}} + \frac{\partial \mathbf{q}}{\partial \mathbf{p}}(0) \right] + \\ &\int_0^T \left[ \frac{d\lambda^T}{dt} \frac{\partial \mathbf{q}}{\partial \mathbf{p}} - \lambda^T \left( \frac{\partial \mathbf{j}}{\partial \mathbf{p}} - \frac{\partial \mathbf{s}}{\partial \mathbf{p}} \right) + \frac{\partial \mathbf{F}}{\partial \mathbf{p}} \right] dt. \end{aligned} \quad (4.40)$$

Thus, just as in the case of the general nonlinear DAE in Section 4.2, we can find the observation function sensitivities by solving first the Backward Adjoint Equation (4.37) for the Lagrangian multiplier  $\lambda(t)$ , and then apply the formula (4.40) to find the desired sensitivities. In other words, we avoid the calculation of the state sensitivities at each time point, this cost being replaced by having to solve an additional DAE backward in time. The cost of calculating  $\hat{\mathbf{x}}_{\text{DC}}$  can be neglected, it is readily available after the DC solution has been calculated. Furthermore, the derivatives  $\frac{\partial \mathbf{q}}{\partial \mathbf{p}}$ ,  $\frac{\partial \mathbf{j}}{\partial \mathbf{p}}$ ,  $\frac{\partial \mathbf{s}}{\partial \mathbf{p}}$  and  $\frac{\partial \mathbf{F}}{\partial \mathbf{p}}$  should be readily available as the parameter dependence of the functions  $\mathbf{q}$ ,  $\mathbf{j}$ ,  $\mathbf{s}$  and  $\mathbf{F}$  is known explicitly.

### Cost analysis

The main burden of the backward adjoint sensitivity equations still is the  $W = \mathcal{O}(n^\alpha)$  ( $\alpha = 3$  for full systems, for sparse systems  $1 \leq \alpha \leq 2$ ) work needed for the LU-decompositions when integrating backward in time for the adjoint equations for  $\lambda(t)$ .

The costs can be summarized as follows:

1. Each time integration step of the backward adjoint equations (4.37) requires  $W + \mathcal{O}(Fn^2)$  operations.
2. The initial condition at  $t = 0$  in (4.40) has a cost  $W + \mathcal{O}(\min(F, m)n^2 + Fm)$ , this is done once only.
3. Each time integration step in (4.40) is now only  $\mathcal{O}(mn + Fm)$ .

### Stability considerations

Clearly, if we wish to adopt the backward adjoint formalism to calculate sensitivities for our electronic circuits, it is of the utmost importance that the adjoint equations are

stable whenever the original circuit equations are. Otherwise, numerical solution of the adjoint equation is severely hampered.

This is certainly not a trivial issue, as has been demonstrated for a simple example in [17]. In that case, an augmented formulation of the adjoint system must be used to circumvent the instability. This implies a larger system must be used, thereby making the solution process more expensive. Fortunately, the same paper also shows that for systems of the form we are using, this problem does not occur due to the fact that the original equations constitute a semi-explicit differential-algebraic equation of index-1. Hence, stability is guaranteed whenever the original problem is stable. This also becomes clear when reformulating the adjoint problem into a forward problem, as is done in the next chapter in Section 5.2. A final confirmation of this fact is found in [57], where it is shown that for "well-formulated" DAE systems, stability of the adjoint equation and its discretization is maintained.

## 4.7 Summary

In this chapter we discussed several ways to calculate sensitivities with respect to parameters. In principle, there are two main methods that can be used. The direct forward method is simple and straightforward, but becomes computationally expensive when the number of parameters is large. The main advantage of the backward adjoint method is that it outperforms the direct forward method when the number of parameters is large. From the analysis in this chapter it is clear that in this case a dramatic speed up in the evaluation of the integrand will be observed.

Despite the advantages of the backward adjoint method over the direct forward method, it still requires the solution of a differential algebraic system backward in time. For industrial problems, this cost is absolutely not negligible, and so any alternatives that guarantee a fast yet accurate calculation of sensitivities are very welcome. In the next section we introduce a first idea by using an application of Model Order Reduction to enable us to reduce the cost. It turns out that this makes the adjoint method an even more attractive option. In subsequent chapters, the idea will be refined.



## Chapter 5

# The Backward Reduced Adjoint Method

In this chapter we introduce the first of our ideas to improve the performance of the , leading to a method that we named Backward Reduced Adjoint Method (BRAM). It consists of applying the Proper Orthogonal Decomposition method to the backward adjoint problem, the basis being based on the forward problem. In this way, the need for solving the full backward equation is eliminated, only a much reduced backward problem needs to be solved. In order to be able to apply POD, it must be shown that the POD bases for the forward and backward adjoint problems are similar. This is done by applying various arguments, one of which is the application of

### 5.1 Introduction

In the previous chapter, we discussed a number of methods that are used for finding the sensitivities of the state vector and derived observation functions with respect to the parameters in the problem. For the specific case of electronic circuits where the DAE is in a special form, a dedicated derivation was provided. We concluded that, when many parameters are involved, the backward adjoint method is the preferred technique. Indeed, when coping with electronic circuits in the electronics industry, often a huge number of parameters is involved (tens of thousands or even more), and designers are extremely keen on finding all of the corresponding sensitivities.

Despite the fact that the backward adjoint method is the method of choice, it is clear that the additional cost of having to solve yet another differential algebraic system of the same size as the original circuit problem does not sound very attractive. Designers expect that, whenever the original forward problem has been solved, the sensitivities

will be readily available. Of course, one could try to convince them that this is not feasible, on the other hand it is worthwhile to explore alternatives that may lead to the desired sensitivities in a more efficient way and avoid the additional DAE solution. In doing this, it helps that designers are often not interested in the state sensitivities, but merely in the sensitivity of several derived observation quantities such as, for example, consumed power.

To show the performance of the backward adjoint method on some industrial examples, within the COMSON project (see Section 1.2) some experiments were carried out. One of the main issues, as one can expect, is the solution of the linear systems. In principle, one could save the LU-decompositions of the final Newton steps in the forward solution, as the Jacobians will be the same in the backward problem. But, clearly, this would mean an enormous amount of additional storage, practically infeasible. Thus, the implementation used made no use of storing the LU-decomposition, but instead recalculated it for the backward adjoint equation. The results obtained for four industrial examples are summarized in the following table:

Circuit	# R	# C	# MOS	trans. sim.	# R,C param.	# MOS param.	trans. + sens.
cross1	1566	9476	508	65 sec	11042		104 sec
cross2	1566	9476	508	65 sec		508	110 sec
x – sect	13629	11916	34607	5210 sec		34607	11111 sec
clock	15199	144067	4869	167 min	159266		424 min

These results clearly indicate that solving the backward adjoint problem is just as expensive as the forward problem, in fact the time is usually doubled. For more information, see [4]. Positive is, of course, that we are able to calculate a huge number of sensitivities at the same time.

In recent years, a number of approaches have been proposed in the literature to use results from the field of model order reduction and carry these over to the problem of sensitivity analysis in order to drastically reduce the time needed for the additional backward simulation. Early approaches consisted of constructing an auxiliary basis [47, 54]. The main motivation of this work is to avoid the computation of the derivative of the original basis used in the simulation. The advantage of these methods was that the sensitivity information is obtained from the generation of only two orthogonal bases, regardless of the number of parameters present in the problem. Unfortunately, the methods have a severe limitation. The range of application is frequency-domain sensitivity analysis, implying that sensitivity analysis in the presence of nonlinear elements remains impractical. A similar limitation is attached to the method described in [29], where a Lanczos-based model order reduction technique was used to calculate the frequency-domain sensitivity of the small-signal response of linear circuits.

From the foregoing, it is clear that several techniques have been developed successfully for the frequency domain, but that the real problem is found when doing sensitivity in the time domain in the presence of nonlinear components. In [1], a tech-

nique was presented for 1-port systems, whereas in [2] the method was extended to the case of multiport systems. The proposed algorithm is based on the Arnoldi method and, therefore, enables systems reduced by orthogonal projection techniques (such as PRIMA [63]) to have their sensitivity evaluated with incremental computational cost. In other words, the accuracy can be influenced by employing more iterations of the reduction process. This paper also demonstrates that passivity is retained. A severe drawback of the method is that it assumes a decomposition of the circuit is known into linear subnetworks and one large network containing all nonlinear components. Although we did not implement the method to assert its efficiency, we are afraid it will not be able to address the large circuit problems we envisaged in our research. In fact, the examples given in the paper are all very small, and only contain transmission lines and not general nonlinear circuits. Furthermore, only sensitivities of linear subnetworks can be obtained, whereas designers are interested in the sensitivity of the entire network including the nonlinear components.

Finally, we wish to mention a more recent development that is presented in [44,74], and which is of a rather general nature. The authors present an analysis of the perturbation-induced errors in the solution of dynamical systems, and estimate errors for a POD-based model order reduction applied to both ODE systems [44], and also differential algebraic systems [74]. These papers contain a very detailed analysis of the errors occurring, and present a technique that can be used to obtain the direction in parameter space that provides maximum growth in the output functional. This direction vector can be calculated with a small number of solutions of the backward adjoint model. In this sense, the emphasis in the paper is not so much on the combination of MOR and the backward adjoint method. However, the analysis can certainly be very useful in investigating the various errors that are associated with such combinations.

In view of the foregoing, we concluded that there is not really a good combination of the backward adjoint method with model order reduction techniques that works for general nonlinear circuits in the time domain. In the mean time, the proper orthogonal decomposition method had been developed much further, and matured. Researchers also started to use it for reducing nonlinear circuits. Hence, we developed a new method that is based on a combination of the backward adjoint method and the POD reduction technique, hoping that this would be feasible for general nonlinear circuits. Indeed, the latter was a strict requirement at the start of the research, as designers in the electronics industry need to be able to cope with such situations continually.

Within this chapter we will cover the derivation of the BRAM method, including giving some of our initial thoughts and justifications of why. From the widely available choice of Model Order Reduction methods and techniques, POD was our first and almost only choice. We explain what it is about the properties of the POD method that we find useful, how we can use it to find a relationship connecting both the forward and backward problems that makes it the most suitable and almost only choice of method to aid the task of reducing the overall cost of a circuit sensitivity analysis within the framework of the backward adjoint method.

We include in our discussion of the backward reduced adjoint method pointing out

further important properties of the systems themselves. We point out the structural similarities of both the forward and reverse differential algebraic equations including the identical circuit topology of the system circuits and the links to Tellegen's treatment of adjoint circuits based on his famous theorem. This is a feature of our systems that is very special to electronic circuits<sup>1</sup>. We show that the snapshot solution matrices of both the forward and backward systems share a common full space basis, and that they do have a common dominant subspace which we take advantage of by reducing the computational cost of the reverse system by applying a projection matrix, constructed from reused data found in the initial circuit simulation, to its system matrices.

We finish with a discussion pointing out the cost advantages of our new method and then discuss some areas for further improvement and development such as the sensitivity of the projector itself to parameter changes and also the influence of full space data and operations on a solution unfolding in the truncated subspace. This will then lead to refinements of the backward reduced adjoint method in the subsequent chapter.

## 5.2 The structure of the backward adjoint equation

Here we begin by revisiting the original backward adjoint equation as derived in Chapter 4. Our aim is to apply the POD model order reduction method to obtain the sensitivities via the backward adjoint equation, but we need convincing arguments to be able to do so. These are found by recalling that the backward adjoint method for sensitivity analysis depends heavily upon having its backward adjoint equations initial conditions set at the time point  $t = T$ , and that it has to be solved back towards time point  $t = 0$ . We see that this is reflected in the equation when it first appears in the derivation, it is found in the following backward time form (we have omitted the dependence of  $\mathbf{C}$  and  $\mathbf{G}$  on the solution  $\mathbf{x}$ , but it is clear that for nonlinear problems this plays an important role):

$$\mathbf{C}^T(\mathbf{p})\dot{\lambda}(t) - \mathbf{G}^T(\mathbf{p})\lambda(t) = - \left( \frac{\partial \mathbf{F}(\mathbf{x}(t, \mathbf{p}))}{\partial \mathbf{x}(t, \mathbf{p})} \right)^T. \quad (5.1)$$

We get a step closer to making the similarities with the forward circuit equation system clearer if we let  $\tilde{t} = T - t$  where  $\tilde{t}$  now is forward time, and then rearrange the equation into the following:

$$\mathbf{G}^T(\mathbf{p})\lambda(\tilde{t}) + \mathbf{C}^T(\mathbf{p})\dot{\lambda}(\tilde{t}) = \left( \frac{\partial \mathbf{F}(\mathbf{x}(\tilde{t}, \mathbf{p}))}{\partial \mathbf{x}(\tilde{t}, \mathbf{p})} \right)^T. \quad (5.2)$$

<sup>1</sup>It is possible to abstract the resulting theory beyond electronic circuits, and formulate a class of problems that has similar properties, but we have not done this here.

For linear systems  $\mathbf{C}$ ,  $\mathbf{G}$  are symmetric matrices (and without dependence on  $\mathbf{x}$ ), and hence equivalent to their transposes. When applied to the above this reveals that, if we express the backward adjoint equation in the forward form, we find that the system shown actually has an identical node and branch circuit structure as the forward system. In this case, we can thus write the modified forward time expression:

$$\mathbf{G}(\mathbf{p})\lambda(\tilde{t}) + \mathbf{C}(\mathbf{p})\dot{\lambda}(\tilde{t}) = \left( \frac{\partial \mathbf{F}(\mathbf{x}(\tilde{t}, \mathbf{p}))}{\partial \mathbf{x}(\tilde{t}, \mathbf{p})} \right)^T. \quad (5.3)$$

We place alongside here the original forward circuit equations for comparison, and just by taking a look it is clear we have the same circuit topology, except that they have different right hand sides:

$$\mathbf{G}(\mathbf{p})\mathbf{x}(t, \mathbf{p}) + \mathbf{C}(\mathbf{p})\dot{\mathbf{x}}(t, \mathbf{p}) = \mathbf{s}(t, \mathbf{p}). \quad (5.4)$$

The foregoing change of time variable clearly shows that there is an intimate relationship between the original and adjoint systems. In [10, 11, 57] this relation is also observed. Related to this is the analysis in [16], where observability of the original system is shown to be related to controllability of the dual system, and solvability of the two problems is linked. From this work, it can be concluded that the forward and backward problems have a similar structure of solutions.

For linear problems, this conclusion is immediate, whereas for the nonlinear case it needs some additional arguments. One could argue that the forward problem can be solved, and solutions  $\mathbf{x}$  can be stored at all time points. If we then calculate the derivative matrices  $\mathbf{C}$  and  $\mathbf{G}$  at these time points, substituting the solutions found, then it is clear that the solution  $\mathbf{x}$  satisfies a linearized forward equation of a form similar to (5.4). Thus, also in the nonlinear case, there is a similarity between the forward and backward problem. In this case, the matrices  $\mathbf{G}$  and  $\mathbf{C}$  may be non-symmetric, at least in the subcircuit areas where nonlinear components are used.

The foregoing gives a clear indication that the POD bases for the forward and backward problem may have a similar structure. This has some interesting consequences, not only because it enables us to formulate a method that combines the backward adjoint equation with the method of proper orthogonal decomposition, but also because it strongly points to applications of Tellegen's theorem. Therefore, we will first explore the latter in the following sections, before presenting the backward reduced adjoint method.



### 5.3 Tellegen's Theorem

The discussion in the previous section has revealed a strong relation between the forward and backward adjoint problems. Apparently, the topological structure of the corresponding circuits is the same, and electronic engineers then immediately think of applying Tellegen's theorem [64, 78]. An interesting coincidence is that Bernard D.H. Tellegen in 1923 joined the Philips Research Laboratories in Eindhoven, which is also the place where most of our work on the subject of this thesis was carried out!

**Theorem 5.1** *Consider an electronic circuit consisting of  $B$  branches, and having potential differences  $V_b$  and electrical currents  $I_b$ , where  $b = 1, \dots, B$ , associated with each branch. If the branch voltages  $V_b$  satisfy the constraints given by Kirchhoff's voltage law and the branch currents  $I_b$  satisfy the constraint given by Kirchhoff's current law, then,*

$$\sum_{b=1}^B V_b I_b = \mathbf{V}^T \mathbf{I} = 0. \quad (5.5)$$

The proof of this remarkable fact is relatively straightforward. If the node potentials are denoted by  $\phi$ , then we have that

$$\mathbf{V} = \mathbf{A}^T \phi,$$

$\mathbf{A}$  being the so-called incidence matrix of the circuit that we have also encountered in Chapter 2. Kirchhoff's current law then states that

$$\mathbf{A} \mathbf{I} = 0.$$

But then we immediately observe that

$$\mathbf{V}^T \mathbf{I} = \phi^T \mathbf{A} \mathbf{I} = 0.$$

This simple theorem is very general, and because it is based upon two fundamental conservation laws it can be demonstrated to be valid beyond the linear circuits discussed here. It is valid for any electrical circuit or other network having any type of component on its branches, be they linear, non-linear, active or passive. In electronics this includes the entire range of voltage and current sources, resistors, diodes and capacitors. Indeed, as we can see from the proof of the theorem, nowhere the branch constitutive equations are used.

Tellegen's theorem can also be lifted up and extended beyond its application to just one circuit network and give a relationship linking two topologically identical circuits, even if they contain different components on each branch. This can sometimes lead to very interesting observations, such as for example reciprocity relations between two networks. The extension we have in mind is the following theorem:

**Theorem 5.2** Consider two electronic circuit networks that are topologically identical, in other words sharing the same incidence matrix  $A$ . Denote the vectors of branch voltages by  $\mathbf{V}_{B1}$  and  $\mathbf{V}_{B2}$ , respectively, and the vectors of branch currents by  $\mathbf{I}_{B1}$  and  $\mathbf{I}_{B2}$ , respectively. Then we have the following identities for these voltage and current vectors:

$$\begin{aligned}\mathbf{V}_{B1}^T \mathbf{I}_{B1} &= 0, \\ \mathbf{V}_{B2}^T \mathbf{I}_{B2} &= 0, \\ \mathbf{V}_{B1}^T \mathbf{I}_{B2} &= 0, \\ \mathbf{V}_{B2}^T \mathbf{I}_{B1} &= 0.\end{aligned}$$

Clearly, this theorem relating two topologically identical networks is very interesting for our research, as we already concluded that the forward problem and the backward adjoint problem for electronic circuits have the same topological structure.

Another useful extension is dependent upon the fact that Kirchhoff's voltage and current laws are not violated by a set of branch voltages and currents if linear operations are applied to them. A linear operation on a zero sum still gives a zero sum result.

**Theorem 5.3** Given two linear operations  $\mathbf{f}_1, \mathbf{f}_2$  acting upon the branch voltages and branch currents of a circuit, the modified values  $\mathbf{V}_f, \mathbf{I}_f$  preserve the zero sum. This is because KVL and KCL are immune to linear operations.

$$\sum_{b=1}^B \mathbf{f}_1(\mathbf{V}_b) \mathbf{f}_2(\mathbf{I}_b) = \mathbf{V}_f^T \mathbf{I}_f = 0. \quad (5.6)$$

This is also valid when mixing branch values for two topologically equivalent circuits.

Tellegen's theorem has been generalized in several ways, see for example [65].

## 5.4 Tellegen's theorem and adjoint sensitivities

In view of the discussion in the previous section, especially about the topological resemblance between the forward and backward problems, it is not surprising that Tellegen's theorem can be connected to the calculation of sensitivities. See, for example, the work described in [36]. The presence of Tellegen's theorem within the inner workings of the backward adjoint method explains the apparent similarities we have found between the adjoint equation (5.3) and the original circuit equation. This should indeed not be

a great surprise as these properties were used in the original derivation. We have already pointed out that they share the same structure, and thus have the same incidence matrix. This adjoint pair of systems is also evaluated over the same time period over the same basis. In this section, we will therefore have a closer look at the application of Tellegen's theorem for the purpose of calculating sensitivities <sup>2</sup>.

Here we present first a discussion of how Tellegen's theorem(s) can be used to calculate sensitivities. To this end, consider a linear time-invariant network  $N$ , which we assume to be a 2-port network. Let  $\hat{N}$  be a 2-port network which is topologically equivalent to  $N$ , i.e. the graphs of the two networks are identical. Furthermore, let  $V_e$  and  $I_e$  denote the voltage and current, respectively, associated with an element  $e$  in the network  $N$ , and  $\psi_e$  and  $\lambda_e$  denote similarly for the corresponding element  $e$  in  $\hat{N}$ . Finally,  $V_i$  and  $I_i$ ,  $i = 1, 2$ , denote the voltage and current variables associated with the 2 ports of  $N$ , and similarly we have  $\psi_i$ ,  $\lambda_i$ ,  $i = 1, 2$  for the network  $\hat{N}$ . The configurations are shown in Figure 5.1.

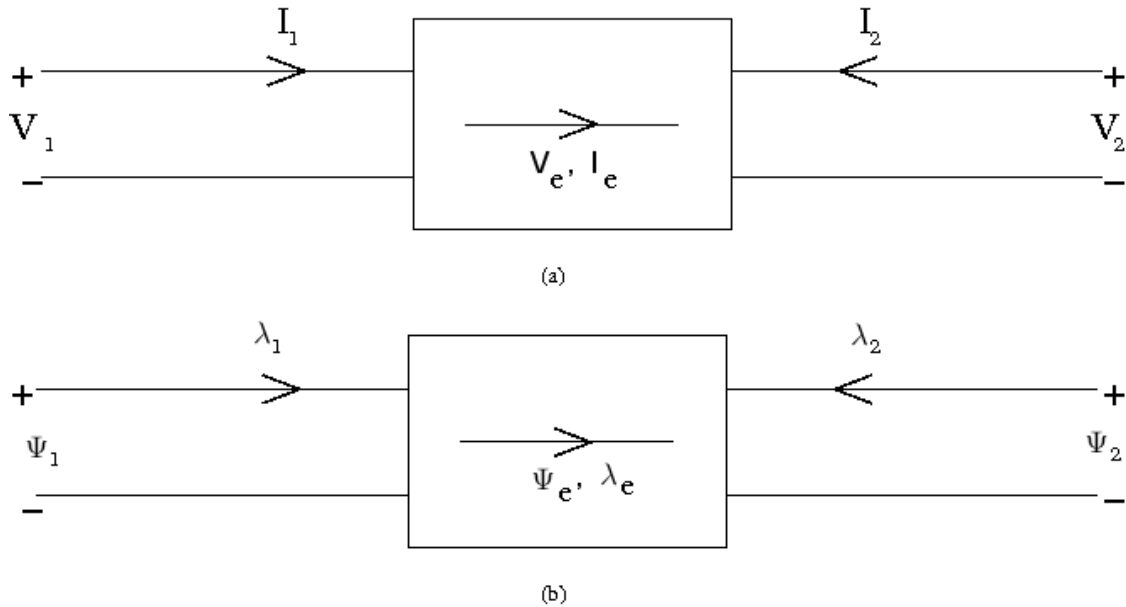


Figure 5.1: 2-port system

We now apply Tellegen's theorem to both networks, and find

$$V_1 \lambda_1 + V_2 \lambda_2 = \sum_e V_e \lambda_e, \quad (5.7)$$

<sup>2</sup>In a very recent paper, an approach is presented to calculate second-order sensitivities, based upon a variation of Tellegen's theorem; see [89]

$$I_1\psi_1 + I_2\psi_2 = \sum_e I_e\psi_e. \quad (5.8)$$

Now suppose that we are going to disturb network  $N$ , but keep the network  $\hat{N}$  the same as before. Then we can again apply Tellegen's theorem, now to the perturbed network and  $\hat{N}$ , to give

$$(V_1 + \Delta V_1)\lambda_1 + (V_2 + \Delta V_2)\lambda_2 = \sum_e (V_e + \Delta V_e)\lambda_e, \quad (5.9)$$

$$(I_1 + \Delta I_1)\psi_1 + (I_2 + \Delta I_2)\psi_2 = \sum_e (I_e + \Delta I_e)\psi_e, \quad (5.10)$$

where  $\Delta V$  and  $\Delta I$  represent the changes in the voltages and currents as a result of the perturbations in  $N$ . Subtracting now (5.7) from (5.9) and (5.8) from (5.10), we get

$$\Delta V_1\lambda_1 + \Delta V_2\lambda_2 = \sum_e \Delta V_e\lambda_e,$$

$$\Delta I_1\psi_1 + \Delta I_2\psi_2 = \sum_e \Delta I_e\psi_e.$$

Subtracting the latter two equations we obtain

$$(\Delta V_1\lambda_1 - \Delta I_1\psi_1) + (\Delta V_2\lambda_2 - \Delta I_2\psi_2) = \sum_e (\Delta V_e\lambda_e - \Delta I_e\psi_e) \quad (5.11)$$

Notice that, up to now, we did not say anything about the construction of the network  $\hat{N}$ , so we still have freedom in doing this. We wish to define the corresponding element  $e$  of  $\hat{N}$  for every element  $e$  in the original network  $N$  in such a way that each of the terms in the right hand side of (5.11) reduces to a function of the voltage and current variables and the change in value of the corresponding network element.

To show how this is done, we consider the simple case of a linear resistive element, for which we have

$$V_R = RI_R.$$

Let us now change the resistance to  $R + \Delta R$ , then

$$V_R + \Delta V_R = (R + \Delta R)(I_R + \Delta I_R),$$

and, on neglecting second-order terms,

$$V_R + \Delta V_R = R I_R + R \Delta I_R + I_R \Delta R.$$

Subtracting the original relation for the resistor, we thus obtain

$$\Delta V_R = R \Delta I_R + I_R \Delta R.$$

Using this expression in (5.11), the terms corresponding to the resistive elements of  $N$  can be written as

$$\sum_R [R \lambda_R - \psi_R] \Delta I_R + I_R \lambda_R \Delta R. \quad (5.12)$$

Thus, if we choose

$$\psi_R = R \lambda_R, \quad (5.13)$$

the term (5.12) reduces to

$$\sum_R I_R \lambda_R \Delta R,$$

which involves only the network variables in the original, unperturbed, network  $N$  and the second network  $\hat{N}$ , as well as the changes in resistance values. This immediately shows how sensitive the original circuit is to changes in resistor values.

Equation (5.13) defines the corresponding elements in the network  $\hat{N}$ . Proceeding in a similar manner, we can determine other elements of  $\hat{N}$  corresponding to other types of network elements, such as inductances, capacitances, controlled sources etc. Indeed, using a notation based on element related incidence matrices (for details, see Chapter 2 in [83]), we know that

$$\mathbf{q}(t, \mathbf{x}) = \sum_e A_e \mathbf{q}_e(t, B_e^T \mathbf{x}),$$

$$\mathbf{j}(t, \mathbf{x}) = \sum_e A_e \mathbf{j}_e(t, B_e^T \mathbf{x}),$$

so that

$$\mathbf{C}(t, \mathbf{x}) = \sum_e A_e \mathbf{C}_e(t, B_e^T \mathbf{x}) B_e^T,$$

$$\mathbf{G}(t, \mathbf{x}) = \sum_e A_e \mathbf{G}_e(t, B_e^T \mathbf{x}) B_e^T.$$

Thus, the network  $\hat{\mathbf{N}}$  can also be defined element-wise.

A table defining adjoint elements corresponding to different types of network elements can be found in [77]. The approach was first presented in [24], which contains a nice derivation and also a table of adjoint descriptions and sensitivities for various types of elements. In [14], the method is used to calculate delay time sensitivities.

## 5.5 Tellegen's theorem mirrored in vector space

Our initial inspiration to apply a projector, created by applying Proper Orthogonal Decomposition to the forward problem, to the backward adjoint equation is the striking similarity of the structure of both systems and because they are evaluated over the same time period. Hence, they most probably also share a set of common basis vectors. In this section, we will substantiate this further.

An interesting point to consider is what happens to the basis in the reverse problem,

$$\mathbf{G}^T(\mathbf{p})\lambda(\tilde{t}) + \mathbf{C}^T(\mathbf{p})\dot{\lambda}(\tilde{t}) = \left( \frac{\partial \mathbf{F}(\mathbf{x}(\tilde{t}, \mathbf{p}))}{\partial \mathbf{x}(\tilde{t}, \mathbf{p})} \right)^T. \quad (5.14)$$

To examine this we create here, for illustration purposes, an observation function that observes the energy consumption of a resistor between voltage nodes 1 and 2 of a 5 state electronic system.

$$\mathbf{V}_R = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t, \mathbf{p}) \\ x_2(t, \mathbf{p}) \\ x_3(t, \mathbf{p}) \\ x_4(t, \mathbf{p}) \\ x_5(t, \mathbf{p}) \end{bmatrix} \quad (5.15)$$

$$\mathbf{V}_R = [x_2(t, \mathbf{p}) - x_1(t, \mathbf{p})] \quad (5.16)$$

$$\text{Power} = V_R^2/R \quad (5.17)$$

$$\mathbf{F}(\mathbf{x}(t, \mathbf{p})) = [x_2^2(t, \mathbf{p}) - 2x_1(t, \mathbf{p})x_2(t, \mathbf{p}) + x_1^2(t, \mathbf{p})] / R_1 \quad (5.18)$$

$$\mathbf{G}_{\text{obs}}(\mathbf{x}(\mathbf{p})) = \int_0^T \mathbf{F}(\mathbf{x}(t, \mathbf{p})) dt \quad (5.19)$$

The observation function  $\mathbf{G}_{\text{obs}}$  integrates the power consumption over a period of time used by our resistor. When applying the backward adjoint method, the right hand side of our backward adjoint equation (5.14) is set to the differential with respect to the state vector of  $\mathbf{F}$ :

$$\mathbf{G}^T(\mathbf{p})\lambda(\tilde{t}) + \mathbf{C}^T(\mathbf{p})\lambda(\tilde{t}) = \left( \frac{1}{R_1} \begin{bmatrix} 2 & -2 & 0 & 0 & 0 \\ 2 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(\tilde{t}, \mathbf{p}) \\ x_2(\tilde{t}, \mathbf{p}) \\ x_3(\tilde{t}, \mathbf{p}) \\ x_4(\tilde{t}, \mathbf{p}) \\ x_5(\tilde{t}, \mathbf{p}) \end{bmatrix} \right)^T \quad (5.20)$$

The right hand side of (5.20) shows the input vector to the backward adjoint system at a particular point in time is formed by applying a linear operator onto the state vector of the forward system at the same point in time.

This also holds more generally. An observation function is very rarely active, it does not introduce new excitations to a system and at most is a linear operation of the existing state solutions of the forward system. Keeping in mind Tellegen's theorem and other observations we have made so far, we can state the following properties with confidence about these two systems that are each others adjoints:

- The two systems can be treated as two circuits with an identical topology.
- The two circuits share the same incidence matrix and branch component distribution.
- Tellegen's Theorem and its extensions can be applied to the pair of circuits.

- The observation function within the backward adjoint method feeds back to the backward adjoint equation at a point in time the state vector, or a linearly operated state vector as the right hand side, from the same point in time.
- No new excitation or basis changing/extending information is provided to the reverse system.

If we consider the pair of equations at a particular instant in time, not only do they have interchangeable current and branch values that satisfy Theorem 5.3, they also share the same instant in time and their state solutions are a linear function of each other, and so the same set of basis functions, at that point in time.

If we extend this Tellegen pairing to each snapshot in the set of all system solutions,  $W = \{x_1, x_2, \dots, x_{NS}\}$ , for the forward system and,  $R = \{\lambda_1, \lambda_2, \dots, \lambda_{NS}\}$  where  $NS$  is the number of snapshots taken in the time period of evaluation we can say both systems share the same full basis vectors.

The other reasons why POD is the most suited method, is that we already have the circuit state snapshots from the initial forward solve of the circuit, these are already stored for use in evaluating the right hand side of the backward adjoint equation, being able to reuse the already stored data once more is an attractive prospect, especially if it contributes to speeding up the calculation. These advantages and the above analysis gives us the confidence to apply a POD projection matrix created from the forward system, to the reverse system. This will then be the basis for the methods developed in this thesis, starting within the following section.

## 5.6 The backward reduced adjoint method

Having clarified the relation between POD bases for the forward and backward adjoint problems, we are now ready to formulate our first method to accelerate the calculation of sensitivities in the general nonlinear setting. We formulate the circuit equations for the circuit under analysis and choose the signal inputs, the period of time and the component parameters for which we want to analyze the sensitivity:

$$\frac{d}{dt} [\mathbf{q}(\mathbf{x}(t), \mathbf{p})] + \mathbf{j}(\mathbf{x}(t), \mathbf{p}) = \mathbf{s}(t, \mathbf{p}). \quad (5.21)$$

We then solve (5.21) (it goes without saying that we need to use the specified initial conditions) over the required time period and store each state in the snapshot matrix,  $W = \{x_{t=1}, \dots, x_{t=T}\}$ . Applying the singular value decomposition to this snapshot matrix we obtain the POD basis and also a distribution of singular values each associated with a single POD basis element listed in order of magnitude.



Once we have the complete decomposition, we build the projection matrix  $V$  by using the method we have already described in detail earlier. In short, we truncate the POD basis using the distribution of singular values as a measure of the dominance of the information represented by the remaining non-truncated basis functions. The remaining subspace contains the most dominant basis functions. We must be careful to keep a balance between the required or desired magnitude of the system reduction and the acceptable error tolerance set for a particular problem. If we over-truncate, although we may have a smaller system to solve, once our system results are projected back into full space we may find they no longer accurately represent the system. An initial good choice is to take the first few dominant singular values that account for 99% of the total diagonal sum. In practice, this may often be relaxed quite considerably, for example to 95% or even less.

At this point we apply the backward adjoint method in the usual way, we choose an observation function and as before we derive the form of the sensitivity equation and also obtain the backward adjoint equation for our particular problem. The circuit sensitivities we are looking for are given once the observation function is evaluated, using the calculated values of  $\lambda$ .

It is at this point that we begin the model order reduction step by using the projection matrix  $V$ , and reducing the backward adjoint equation using the Galerkin method of projection. The solution  $\lambda \in \mathbb{R}^n$  of the backward adjoint equation in full space, we can approximate it by using a smaller vector  $\mathbf{k} \in \mathbb{R}^k$  where  $V\mathbf{k} \approx \lambda$ :

$$\underbrace{\mathbf{C}^T(\mathbf{p})V\dot{\mathbf{k}}(\tilde{t}) + \mathbf{G}^T(\mathbf{p})V\mathbf{k}(\tilde{t}) - \left( \frac{\partial \mathbf{F}(\mathbf{x}(\tilde{t}, \mathbf{p}))}{\partial \mathbf{x}(\tilde{t}, \mathbf{p})} \right)^T}_{\mathbf{r}(\mathbf{k}) \text{ Residual}} \sim 0. \quad (5.22)$$

Equation (5.22) is now an overdetermined system, with more equations than unknowns. It is also not equal to zero, except, of course, if  $V$  spans the entire space.

We are interested in finding  $\mathbf{k}(t)$  such that the remaining residual  $\mathbf{r}(\mathbf{k})$  is orthogonal to the subspace spanned by the approximating basis vectors chosen in the truncation step. As we have shown the standard way to do this is to apply a Galerkin projection giving the following constraint,

$$V^T \mathbf{r}(\mathbf{k}) = 0. \quad (5.23)$$

Applying the constraint set in (5.23) to our overdetermined backward adjoint equation (5.22), we obtain the following set of equations:

$$\mathbf{V}^T \mathbf{C}^T(\mathbf{p}) \mathbf{V} \dot{\mathbf{k}}(\bar{t}) + \mathbf{V}^T \mathbf{G}^T(\mathbf{p}) \mathbf{V} \mathbf{k}(\bar{t}) = \mathbf{V}^T \left( \frac{\partial \mathbf{F}(\mathbf{x}(\bar{t}, \mathbf{p}))}{\partial \mathbf{x}(\bar{t}, \mathbf{p})} \right)^T. \quad (5.24)$$

Introducing the notation  $\hat{\mathbf{C}}^T, \hat{\mathbf{G}}^T$  to represent the reduced system matrices we have the following reduced system of equations:

$$\hat{\mathbf{C}}^T(\mathbf{p}) \dot{\mathbf{k}}(\bar{t}) + \hat{\mathbf{G}}^T(\mathbf{p}) \mathbf{k}(\bar{t}) = \mathbf{V}^T \left( \frac{\partial \mathbf{F}(\mathbf{x}(\bar{t}, \mathbf{p}))}{\partial \mathbf{x}(\bar{t}, \mathbf{p})} \right)^T. \quad (5.25)$$

The function on the right hand side of (5.25) is evaluated in full space, using the state solution  $\mathbf{x}(\bar{t}, \mathbf{p})$ , before being projected to the truncated subspace. This is a system that is also solvable by Newton methods if the matrix pencil  $\{\hat{\mathbf{C}}, \hat{\mathbf{G}}\}$  is nonsingular.

The method described in the foregoing is the first version of our backward reduced adjoint method, shortly indicated by BRAM. It was introduced and presented at the SCEE 2006 conference in Sinaia, see [46]. In the next section, we will discuss some features of the method, but will also indicate that there is a need for a reformulated approach in order to facilitate a more sound analysis of the errors involved in the reduction process.

## 5.7 Analysis of the backward reduced adjoint method

The backward reduced adjoint method introduced in the foregoing section constituted the first idea of rendering the calculation of sensitivities using the backward adjoint method more efficient. Indeed, it avoids the solution of the backward adjoint equation; instead, a POD basis is formed for solutions of the forward problem, and this basis is used to expand the solution of the backward adjoint problem. All it takes is the storage of the snapshots at all, or a subset, time points, followed by a singular value decomposition. Solutions of the backward adjoint problem are then found in the form  $\lambda = \mathbf{V} \mathbf{k}$ , where  $\mathbf{V}$  is the projection matrix associated with the truncated POD basis. It then remains to solve the reduced backward adjoint problem.

Having formulated the procedure for calculating the approximate sensitivities, it is now essential to evaluate the new technique as far as cost and accuracy is concerned. This will be done in the following subsections.

### 5.7.1 The cost of the backward reduced adjoint method

We have managed to preserve the advantages of the original backward adjoint method, and have enhanced the method by the application of a model order reduction technique

to one of the most costly steps by re-using existing data. We identified the main burden of the backward adjoint sensitivity equation as  $W = \mathcal{O}(N^3)$ , we have now been able to reduce that to  $W = \mathcal{O}(N_k^3)$  where  $N_k$  represents the dimension of the reduced system. Clearly, this amounts to quite a reduction in computation cost if  $N_k \ll N$ .

There is also additional cost associated with the use of POD, of course. This is associated with the singular value decomposition of the matrix of snapshots. This cost is more difficult to assess, as it depends on the number of time points chosen to store snapshots. It is not advisable to store the solution  $x$  at all time points, as this may lead to an excessively large matrix. Instead, we could use snapshots only at a prescribed sequence of time points, or use a more sophisticated adaptive procedure to assess when snapshots should be added. Clearly, this is a point of research to be addressed separately, and is one of the points we will mention in our conclusions in Chapter 9.

The calculation of the dominant singular values and corresponding POD basis vectors can also be done in a more efficient way by exploiting recent developments in numerical linear algebra, more specifically concerning the calculation of specific eigenvalues. For more information, we refer to [8, 41, 42].

### 5.7.2 System subspace restriction concerns

The matrices  $\hat{C}^T, \hat{G}^T$  are the system matrices of the reduced system, restricting the system and also the evolution of the  $\lambda$  solutions to the truncated subspace which we will denote by  $S$ . A representation of the evolution of the solution  $\lambda$  is shown in the following figure.

If we pay closer attention to the right hand side of our reduced backward adjoint system (5.25), we notice the following.

1. We are using the system state solution  $x$  in the full space
2. We are evaluating the right hand side in full space
3. We then project the result onto the subspace before continuing to solve for  $\lambda$ .

The points above introduce some uncertainty in the development of the solution  $\lambda$ , or more specifically its reduced order approximation, as it is unknown how a function will behave in full space when interacting with the restricted solution flow in the truncated subspace. This is illustrated in the above figure. Hence, it is important to investigate this in more detail, as will be done in the next subsection.

It is worth mentioning here that this problem is addressed in detail in [68]. The above features imply that additional errors are made, and this severely complicates the error analysis of the POD method. The paper mentioned provides a study of the different components of the error.

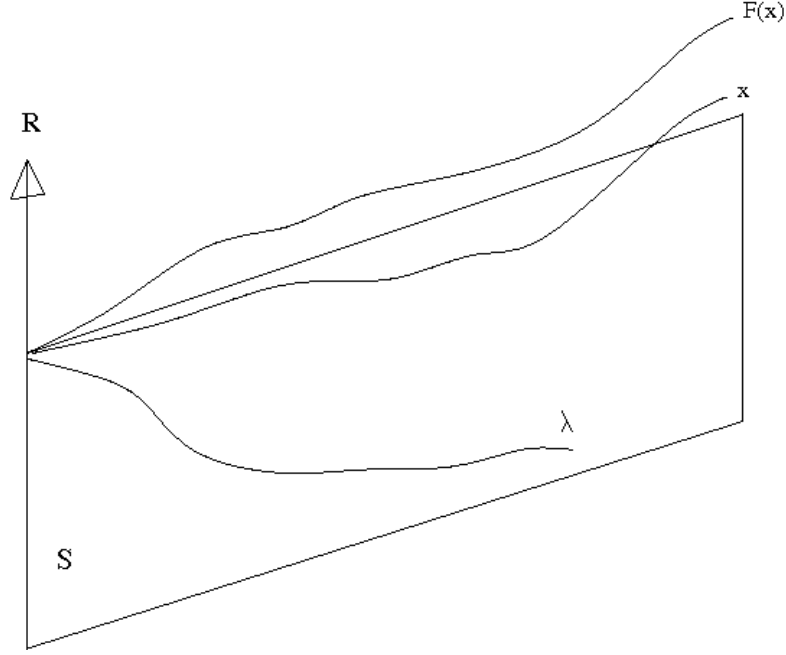


Figure 5.2: Backward Adjoint Equation and Subspace

### 5.7.3 The effect of the projection matrix on sensitivity values

Although the calculation of individual state sensitivities is done implicitly in the BRAM method, taking a closer look at the effect a projection matrix could have on state sensitivities reveals an extra term which we need to be careful about. This extra term shows that the sensitivity is dependent on the sensitivity of the projection matrix to changes in its parameter values, which can also be thought of as the sensitivity of the most dominant system basis functions.

$$\mathbf{x}(t, \mathbf{p}) \approx \mathbf{P}(\mathbf{p})\tilde{\mathbf{x}}(t, \mathbf{p}) \quad (5.26)$$

$$\frac{\partial \mathbf{x}}{\partial \mathbf{p}}(t, \mathbf{p}) \approx \frac{\partial \mathbf{P}(\mathbf{p})}{\partial \mathbf{p}} \cdot \tilde{\mathbf{x}}(t, \mathbf{p}) + \mathbf{P}(\mathbf{p}) \cdot \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{p}}(t, \mathbf{p}) \quad (5.27)$$

One can see in (5.27) that there is now an extra term involved in the projection of the reduced state sensitivity back to the original space, namely the dependence of the POD basis on the set of parameters  $\mathbf{p}$ . As long as this term is relatively small, and we can show that it varies slowly as a function of changing parameter values, then we are allowed to apply POD to the backward adjoint method using a projection matrix based

upon a fixed setting of the parameters (usually the nominal value of parameters). If this is not the case, then the second term in the above expression will have a significant effect, and hence the accuracy of our approximation may deteriorate quickly.

## 5.8 Summary

In this chapter, we have introduced the backward reduced adjoint method (BRAM), in which sensitivities are approximated by solving a reduced (by projection) form of the backward adjoint equation. We have demonstrated that the use of the POD basis obtained from the solution of the forward problem is adequate, as this basis is similar to the one for the backward problem. This is verified in several ways, and also guaranteed by Tellegen's theorem.

A point of concern with this first idea of reducing the backward adjoint equation is that the accuracy of the approximate sensitivities depends on the sensitivity of the POD basis on the parameters. Another complication is that a full analysis of the error is rather complicated, especially in the case of nonlinear components. It is not clear at this stage how the various errors contribute to the total error made in calculating the sensitivities. For these reasons, we will propose in the next chapters to change the formulation of the BRAM technique, and also perform a thorough investigation into the behavior of the POD basis as a function of parameters in a subsequent chapter.

## Chapter 6

# Backward Reduced Adjoint Method II

In this chapter we introduce a modified method, BRAM II, that is also based on the use of proper orthogonal decomposition. It introduces an additional forward problem to be solved, but has the advantage of guaranteeing a more accurate sensitivity analysis. Its accuracy is dependent only on the sensitivity of the POD basis with respect to the parameters, which is an issue that will be discussed in the next chapter.

### 6.1 Method overview

The BRAM II method is very similar to the BRAM method introduced in the foregoing chapter, except for the following point. We take the projection matrix obtained using the snapshot data from the forward system solution, and apply this projection to the original forward system which we then solve again over the same time period. The backward adjoint equation is then reduced in the same way, but the observation function is evaluated on the states of the reduced forward solution. This action restricts all system operations and solutions to evolve on the same subspace, and thereby takes away the concerns we expressed in Chapter 5.

A very attractive feature of this new approach is that the adjoint method formulation itself guarantees that the sensitivity result obtained is exactly the same as the sensitivity result obtained from the direct forward method. In other words, there is no approximation anymore going from the forward to the backward problem. This is in contrast with the original BRAM method, where we do have an approximation error in going from the forward to the backward problem and a resulting complex structure of the total error.

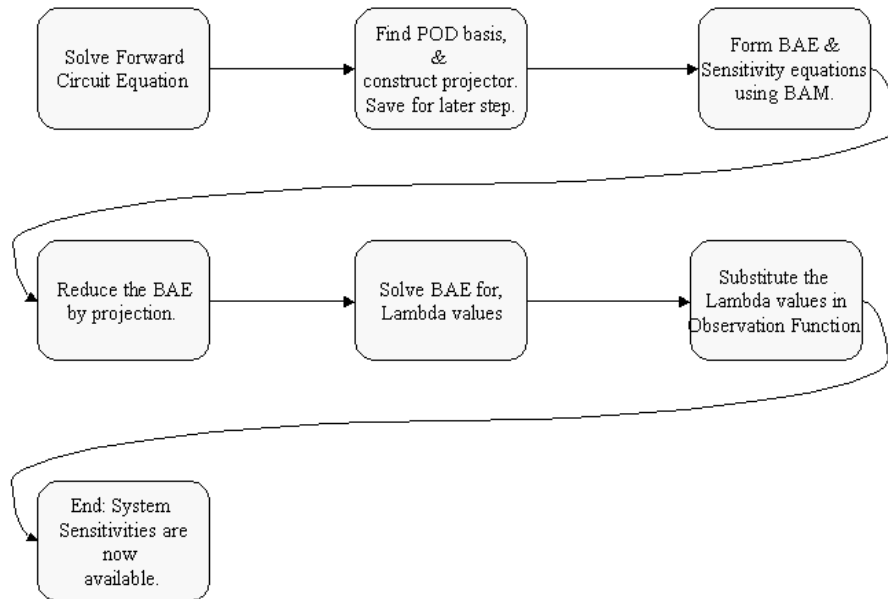


Figure 6.1: Flow chart for BRAM

The foregoing does not mean that there is no error in the calculated sensitivities. However, this error can be assessed entirely by studying approximation properties of the forward problem. In fact, the error is the same as the error made in the POD method. We will discuss this in more detail later in this chapter. Let us now first illustrate the differences between the methods BRAM and BRAM II in a flow chart, as given in Figures 6.1 and 6.2.

From these flow charts it is clear that the BRAM II method involves some additional steps as compared to the original BRAM method. This will have an effect on the cost of the method, but this is compensated by a better insight into the errors made and the final approximation.

## 6.2 Setting up BRAM II

In a previous chapter we followed through the steps of the backward adjoint method to derive the expressions for the sensitivity equation in terms of the unknown vector  $\lambda$ , and the backward adjoint differential algebraic equations. Here we will go through the same steps, but before we do that we will take the forward circuit equation reduction step, obtain the projection matrix and then derive the steps again, but this time restrict the problems to the reduced subspace from the very beginning.

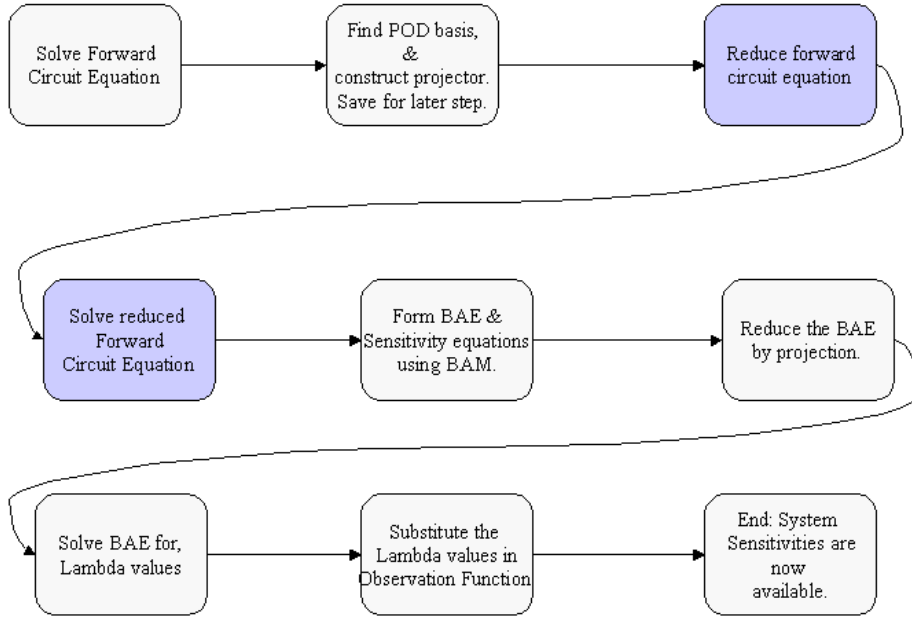


Figure 6.2: Flow chart for BRAM II

Over the required time period  $(0, T)$  we solve the circuit equations,

$$\frac{d}{dt} [\mathbf{q}(\mathbf{x}(t), \mathbf{p}), \mathbf{p}] + \mathbf{j}(\mathbf{x}(t), \mathbf{p}, \mathbf{p}) = \mathbf{s}(t, \mathbf{p}), \quad (6.1)$$

and store the snapshots in the snapshot matrix  $W = \{\mathbf{x}(t = 0), \dots, \mathbf{x}(t = T)\}$ . To simplify the analysis, we will assume that the solution is stored at all time points. Taking the singular value decomposition,  $W = V^T S U$ , we obtain the POD basis  $V$ . Observing the decaying singular values, we determine the most dominant subspace containing the basis vectors associated with the strongest singular values and truncate the POD basis accordingly to obtain the projection matrix  $V$ . In the following chapters, we will show the decay of these singular vectors for a number of industrial examples. It turns out that, for our industrial examples, the dimension of the space can be reduced considerably.

Applying this projection matrix to the forward equation by approximating the solution,  $\mathbf{x} \approx V\mathbf{z}$ , where  $\mathbf{z}$  is the reduced subspace solution that would evolve in the subspace spanned by the the basis in  $V$  when solving the reduced system. Substituting the approximation in the forward problem gives a system,



$$\frac{d}{dt} [\mathbf{q}(\mathbf{V}\mathbf{z}(t, \mathbf{p}), \mathbf{p})] + \mathbf{j}(\mathbf{V}\mathbf{z}(t, \mathbf{p}), \mathbf{p}) \sim \mathbf{s}(t, \mathbf{p}). \quad (6.2)$$

Applying the Galerkin projection we have the following reduced system:

$$\mathbf{V}^T \frac{d}{dt} [\mathbf{q}(\mathbf{V}\mathbf{z}(t, \mathbf{p}), \mathbf{p})] + \mathbf{V}^T \mathbf{j}(\mathbf{V}\mathbf{z}(t, \mathbf{p}), \mathbf{p}) = \mathbf{V}^T \mathbf{s}(t, \mathbf{p}). \quad (6.3)$$

We solve this reduced equation (6.3) and record the results, keeping them in the snapshot matrix  $\mathbf{Z} = \{\mathbf{z}(t = 1), \dots, \mathbf{z}(t = T)\}$ . These results are solutions that have evolved entirely in the restricted subspace and are expressed in the basis of the reduced space. We will later need to evaluate the observation function using these state solutions, we can express these solutions in full space by a change of basis,  $\bar{\mathbf{x}} = \mathbf{V}\mathbf{z}$ , where  $\bar{\mathbf{x}}$  is the exact reduced solution expressed in the coefficients of full space, but restricted to the truncated subspace.

We now carry on as we would have done in the backward adjoint method, but we place the whole process in the reduced space by using the exact subspace solutions, and the substitution of  $\bar{\mathbf{x}} = \mathbf{V}\mathbf{z}$ . To start the calculation of the sensitivities, we proceed as before. The starting point is the reduced circuit equation (6.3). We equate this equation to zero (by putting all terms in the left hand side), multiply it by  $\lambda(t)$ , and then integrate it over a time period:

$$\begin{aligned} 0 &= \int_0^T \lambda^T(t) \left[ \mathbf{V}^T \frac{d}{dt} \frac{d\mathbf{q}}{d\mathbf{p}} + \mathbf{V}^T \frac{d\mathbf{j}}{d\mathbf{p}} - \mathbf{V}^T \frac{d\mathbf{s}}{d\mathbf{p}} \right] dt, \\ 0 &= \int_0^T \lambda^T(t) \left[ \mathbf{V}^T \frac{d}{dt} \frac{d\mathbf{q}}{d\mathbf{p}} \right] dt + \int_0^T \lambda^T(t) \left[ \mathbf{V}^T \frac{d\mathbf{j}}{d\mathbf{p}} - \mathbf{V}^T \frac{d\mathbf{s}}{d\mathbf{p}} \right] dt. \end{aligned}$$

The left integral is evaluated by parts and on completing all differentials we arrive at

$$\begin{aligned} 0 &= \left[ \lambda^T(t) \mathbf{V}^T \frac{d}{d\mathbf{p}} \mathbf{q}(\bar{\mathbf{x}}(t, \mathbf{p})) \right] \Big|_0^T + \\ &\int_0^T \left[ - \left( \frac{d\lambda^T}{dt} \mathbf{V}^T \mathbf{C}\mathbf{V} - \lambda^T \mathbf{V}^T \mathbf{G}\mathbf{V} \right) \hat{\mathbf{x}} - \frac{d\lambda^T}{dt} \mathbf{V}^T \frac{\partial \mathbf{q}}{\partial \mathbf{p}} + \lambda^T \mathbf{V}^T \left( \frac{\partial \mathbf{j}}{\partial \mathbf{p}} - \frac{\partial \mathbf{s}}{\partial \mathbf{p}} \right) \right] dt. \end{aligned} \quad (6.4)$$

We then rearrange (6.4) to isolate the state sensitivity:

$$\left[ \lambda^\top(t) \mathbf{V}^\top \frac{d}{dt} \mathbf{q}(\bar{\mathbf{x}}(t, \mathbf{p})) \right] \Big|_0^\top + \int_0^\top \left[ -\frac{d\lambda^\top}{dt} \mathbf{V}^\top \frac{\partial \mathbf{q}}{\partial \mathbf{p}} + \lambda^\top \mathbf{V}^\top \left( \frac{\partial \mathbf{j}}{\partial \mathbf{p}} - \frac{\partial \mathbf{s}}{\partial \mathbf{p}} \right) \right] dt = \int_0^\top \left[ \left( \frac{d\lambda^\top}{dt} \mathbf{V}^\top \mathbf{C} \mathbf{V} - \lambda^\top \mathbf{V}^\top \mathbf{G} \mathbf{V} \right) \hat{\mathbf{x}} \right] dt = \quad (6.5)$$

We finally have expressed the backward adjoint equation for the reduced subspace and now have,

$$\mathbf{V}^\top \mathbf{C}^\top(\mathbf{p}) \mathbf{V} \dot{\lambda}_r(t) - \mathbf{V}^\top \mathbf{G}^\top(\mathbf{p}) \mathbf{V} \lambda_r(t) = -\mathbf{V}^\top \left( \frac{\partial \mathbf{F}(\bar{\mathbf{x}}(t, \mathbf{p}))}{\partial \bar{\mathbf{x}}(t, \mathbf{p})} \right)^\top. \quad (6.6)$$

Rewriting (6.6) in its forward form by setting  $\tilde{t} = T - t$  gives us the familiar structure,

$$\mathbf{V}^\top \mathbf{C}^\top(\mathbf{p}) \mathbf{V} \dot{\lambda}_r(\tilde{t}) + \mathbf{V}^\top \mathbf{G}^\top(\mathbf{p}) \mathbf{V} \lambda_r(\tilde{t}) = \mathbf{V}^\top \left( \frac{\partial \mathbf{F}(\bar{\mathbf{x}}(t, \mathbf{p}))}{\partial \bar{\mathbf{x}}(t, \mathbf{p})} \right)^\top. \quad (6.7)$$

The  $\lambda_r$  values are the solutions of the backward adjoint equation in the truncated space, they are restricted to the reduced basis. The right hand side observation function evaluation is done on forward state solutions that are themselves restricted to the truncated space. The total system is restricted to the reduced space:

$$\bar{\lambda} = \mathbf{V}^\top \lambda_r.$$

Once we have the  $\bar{\lambda}$  and  $\bar{\mathbf{x}}$  solutions, we can substitute them in to the expression for the system observation function sensitivity:

$$\begin{aligned} \frac{d}{d\mathbf{p}} \mathbf{G}_{\text{obs}}(\bar{\mathbf{x}}(\mathbf{p}), \mathbf{p}) &= \bar{\lambda}^\top(0) \left[ \frac{\partial \mathbf{q}}{\partial \bar{\mathbf{x}}}(0) \cdot \hat{\mathbf{x}}_{\text{DC}} + \frac{\partial \mathbf{q}}{\partial \mathbf{p}}(\mathbf{0}) \right] + \\ &\int_0^\top \frac{d\bar{\lambda}^\top}{dt} \frac{\partial \mathbf{q}}{\partial \mathbf{p}} - \bar{\lambda}^\top \left( \frac{\partial \mathbf{j}}{\partial \mathbf{p}} - \frac{\partial \mathbf{s}}{\partial \mathbf{p}} \right) + \frac{\partial \mathbf{F}}{\partial \mathbf{p}} dt. \end{aligned} \quad (6.8)$$

Figure 6.3 illustrates how we have manipulated the solution flow of both forward and backward systems, and the observation function evaluation on to the dominant truncated subspace. We have shown that it is possible to apply the backward adjoint method

in combination with the proper orthogonal decomposition reduction step to calculate system sensitivities, we retain the advantages of model order reduction and are able to improve upon our BRAM method by restricting the system fully to the truncated subspace.

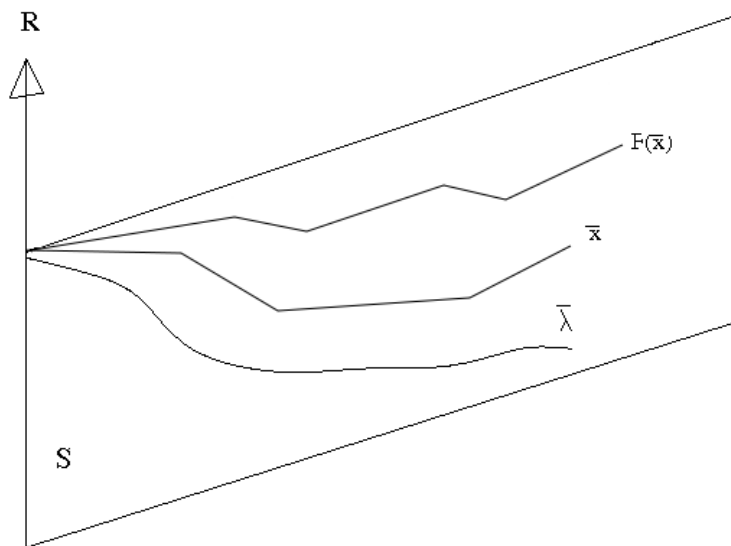


Figure 6.3: Backward Adjoint Equation Restricted to Subspace

What we have done is rather than applying POD to the forward problem and using the basis immediately for the approximate solution of the backward adjoint equation, we have introduced first a reduced order model for the forward problem. This is an essential additional step, which we think is vital in obtaining a good insight in the errors made during the entire BRAM process. It does not necessarily mean that the original version of the backward reduced adjoint method will not work in practice. From a mathematical point of view, it is better to take care and develop the methods and theory in a more incremental way.

### 6.3 Analysis of the BRAM II method

The method developed in the previous section is similar to the method described in Chapter 5. For linear problems, the results obtained are the same, but if the circuit contains nonlinear components there may be differences between the sensitivities obtained.

Our intention in developing this new version of the backward reduced adjoint method was to clarify the various errors made in the approximation of sensitivities. For the original method, the total error consists of several contributions that are not easily assessed. For BRAM II, however, the situation is much more clear due to the fact that the calculated backward adjoint sensitivities are exactly equal to the sensitivities of the reduced forward problem. In other words, no error is made after the original circuit has been put into its reduced formulation. This follows directly from the derivation given in the foregoing section.

Thus, for the new method, the error is equal to the difference between the sensitivities of the original problem and the reduced problem. This can be measured by evaluating

$$\mathbf{e} = \frac{\partial \mathbf{x}}{\partial \mathbf{p}} - \frac{\partial V \mathbf{z}}{\partial \mathbf{p}}.$$

Clearly, this can also be written as

$$\mathbf{e} = \left( \frac{\partial \mathbf{x}}{\partial \mathbf{p}} - V \frac{\partial \mathbf{z}}{\partial \mathbf{p}} \right) - \frac{\partial V}{\partial \mathbf{p}} \mathbf{z}.$$

This implies that the error consists of two main parts:

- the error in approximating  $\mathbf{x}$  by  $V\mathbf{z}$  for a fixed  $\mathbf{p}$
- the rate of change of the POD basis, represented by the projection matrix  $V$ , as a function of the parameters  $\mathbf{p}$

The former of these contributions can be controlled by the POD process, and can be reduced by taking more singular values and corresponding basis vectors into account. In the limit of  $V$  spanning the entire space, this error term reduces to zero. In principle, of course, it is not just the error of the POD approximation, but the parameter dependence of the POD error that must be considered. However, it is clear that these errors are very much related, and the latter error can also be influenced by extending the POD basis used for the approximations. We discuss this in more detail in the next section.

Much more interesting is the second contribution to the error. In practice, one will always use a certain nominal parameter setting for which the solutions  $\mathbf{x}$  and  $\mathbf{z}$  are calculated, and the matrix  $V$  will be based upon the POD basis for that nominal setting. However, for other settings of the parameter set  $\mathbf{p}$ , it may well be that the POD basis changes considerably, and this would lead to a large contribution of the term containing  $\frac{\partial V}{\partial \mathbf{p}}$ . Hence, the crucial question is: what is the parameter dependence of the projection matrix  $V$  for our applications in the electronics industry? This is an important question, which we will answer in the next chapter.

The much clearer picture for BRAM II as far as the error is concerned, comes with a cost. This cost is the additional forward solution of a reduced equation. Maybe this can be

avoided, but we have not made an attempt to do this as our research concentrated on the question a few lines above, i.e. the dependence of the POD basis on the parameters. Therefore, we leave the question of avoiding the extra forward solve as a future research problem, and it is mentioned as such in Chapter 9.

## 6.4 The Galerkin projection and POD error estimation

In the discussion so far we have stepped over the problems that can occur while constructing (6.3). This is, however, an extremely important point that is still topic of ongoing discussions in the POD world. It is certainly not natural that the reduced problem is well-defined. Especially for the complex nonlinear transistor models [72] that are common in the electronics industry, it is rather difficult to assess what the effects are of the substitution  $\mathbf{x} = \mathbf{V}\mathbf{z}$ . These models are constructed in a largely heuristic way, using a combination of physical and electronic arguments as well as a wealth of experimental results summarized in fitted data. Besides this, the cost for evaluating the nonlinear reduced models may even be larger than for the original models!

Apart from the foregoing problems, which are almost impossible to address due to the complexity and high degree of heuristics of the models used, there is also the problem of assessing the accuracy of the reduced order model. Fortunately, in recent years, the theory has been advanced quite a lot. Let us start with an ODE system of the form

$$\mathbf{x}'(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{f}(t, \mathbf{x}(t)), \quad (6.9)$$

with continuous  $\mathbf{f} : [0, T] \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  and with initial condition  $\mathbf{x}(0) = \mathbf{x}_0$ . Let  $\psi_1, \psi_2, \dots, \psi_l$  be a POD basis of rank  $l$ , and use the Galerkin ansatz

$$\mathbf{x}^l(t) := \sum_{j=1}^l (\mathbf{x}^l(t)^T \psi_j) \psi_j = \sum_{j=1}^l \mathbf{x}_j^l(t) \psi_j.$$

Then the Galerkin projection leads to the system

$$\psi_i^T \mathbf{x}'(t) = \psi_i^T \mathbf{A} \mathbf{x}^l(t) + \psi_i^T \mathbf{f}(t, \mathbf{x}^l(t)), \quad (6.10)$$

with initial condition

$$\psi_i^T \mathbf{x}^l(0) = \psi_i^T \mathbf{x}_0,$$

for  $i = 1, 2, \dots, l$ .

If  $\mathbf{f} \equiv 0$ , then it can be shown that

$$\int_0^T \|\mathbf{x}(t) - \mathbf{x}^l(t)\|^2 dt \leq C \left( \sum_{i=l+1}^{\infty} \sigma_i + \int_0^T \|\mathbf{x}'(t) - \mathcal{P}^l \mathbf{x}'(t)\|^2 dt \right),$$

where  $\mathcal{P}^l$  denotes the projector onto the span of the POD basis vectors  $\psi_1, \psi_2, \dots, \psi_l$ , and the  $\sigma_i$  are the singular values associated with the POD basis.

In view of the fact that a term involving the derivative  $\mathbf{x}'(t)$  and its projection appears in the error estimate, some researchers prefer to use a modified POD method:

$$\min \int_0^T \|\mathbf{x}(t) - \mathcal{P}^l \mathbf{x}(t)\|^2 + \|\mathbf{x}'(t) - \mathcal{P}^l \mathbf{x}'(t)\|^2 dt \quad \text{s.t.} \quad (\psi_i, \psi_j) = \delta_{ij}.$$

In this case, the error estimate is

$$\int_0^T \|\mathbf{x}(t) - \mathbf{x}^l(t)\|^2 dt \leq C \sum_{i=l+1}^{\infty} \lambda_i.$$

For more information and a precise derivation, we refer to [39, 43, 51, 68, 85]. In some of these papers, the nonlinear case is addressed too. For example, [68] presents a very detailed study of the errors involved in solving a nonlinear ODE initial value problem using POD and Galerkin projection under rather general (but difficult to verify) conditions on the nonlinearities.

For DAE systems, the POD literature is not well developed yet. In [82, 83], nonlinear systems of DAEs are considered for a number of specific IC models. Here, both Galerkin and Petrov-Galerkin projections are taken into consideration.

On the other hand, we feel that it is not necessary to develop a separate theory for differential-algebraic systems. The reason for this is that the theory of canonical projectors [56] can be used to split the DAE into an ODE and a number of algebraic systems. First the system for the "differential variables" can be solved. After that, the "algebraic variables" can be calculated as they depend on the "differential variables" and their time derivatives up to a certain order. We used this argument already in Section 4.1 when constructing the adjoint equation in the DAE case.

Now let us consider the system

$$A\mathbf{x}' + B\mathbf{x} = \mathbf{q}, \tag{6.11}$$

with  $A, B \in \mathbb{R}^{n \times n}$  matrices and  $\mathbf{x}, \mathbf{q} \in \mathbb{R}^n$  vectors. We assume that the matrix  $A$  is singular, so that the system is differential-algebraic. We introduce the projector  $Q$  onto

ker  $A$ , and the complementary projector  $P$ . They are characterized by

$$AQ = 0, \quad Q^2 = Q, \quad Q + P = I. \quad (6.12)$$

Thus, it also holds:

$$AP = A, \quad P^2 = P, \quad PQ = QP = 0. \quad (6.13)$$

By using these properties, we find the following identities:

$$(A + BQ)P = A, \quad (A + BQ)Q = BQ. \quad (6.14)$$

Then, recalling (6.12), we can rewrite (6.11) as:

$$(A + BQ)(Px' + Qx) + BPx = \mathbf{q}. \quad (6.15)$$

If the matrix  $A + BQ$  is non-singular, we can write:

$$Px' + Qx = (A + BQ)^{-1}(\mathbf{q} - BPx). \quad (6.16)$$

Since we have the decomposition of the identity  $I = P + Q$ , and  $P, Q$  are orthogonal projectors, this equation is equivalent to the two equations obtained after left-multiplication by  $P$  and  $Q$ :

$$Px' = P(A + BQ)^{-1}(\mathbf{q} - BPx), \quad (6.17)$$

$$Qx = Q(A + BQ)^{-1}(\mathbf{q} - BPx). \quad (6.18)$$

The first equation is an ordinary differential equation for  $\mathbf{y} := Px$ , the second is an algebraic equation which expresses  $\mathbf{z} := Qx$  in terms of  $\mathbf{y}$ . In this case we say that system (6.11) has tractability index 1. We call  $\mathbf{y}$  differential component of  $\mathbf{x}$ , and  $\mathbf{z}$  algebraic component of  $\mathbf{x}$ . The previous equations can be written as:

$$\mathbf{y}' = P(A + BQ)^{-1}(\mathbf{q} - BP\mathbf{y}), \quad (6.19)$$

$$\mathbf{z} = Q(A + BQ)^{-1}(\mathbf{q} - BP\mathbf{y}). \quad (6.20)$$

If the matrix  $A + BQ$  is singular, the previous procedure can be iterated, but this is not necessary for our index-1 circuits.

Having found a decomposition of a general DAE system in terms of differential and algebraic equations, we now apply the foregoing to circuits in order to see what the projectors are in that specific case. For simplicity, we consider a linear RLC electric network, that is, a network which connects linear capacitors, inductors and resistors, and independent voltage and current sources,  $\mathbf{v}(t) \in \mathbb{R}^{n_v}$  and  $\mathbf{i}(t) \in \mathbb{R}^{n_i}$ . The unknowns which describe the network are the node potentials  $\mathbf{u}(t) \in \mathbb{R}^n$ , and the currents  $\mathbf{j}_L(t) \in \mathbb{R}^{n_L}$  and  $\mathbf{j}_V(t) \in \mathbb{R}^{n_v}$  through inductors and voltage sources, respectively.

Following the formalism of Modified Nodal Analysis (MNA), we introduce: the incidence matrices  $\mathbf{A}_C \in \mathbb{R}^{n \times n_C}$ ,  $\mathbf{A}_L \in \mathbb{R}^{n \times n_L}$  and  $\mathbf{A}_R \in \mathbb{R}^{n \times n_G}$ , which describe the branch-node relationships for capacitors, inductors and resistors; the incidence matrices

$\mathbf{A}_V \in \mathbb{R}^{n \times n_V}$  and  $\mathbf{A}_I \in \mathbb{R}^{n \times n_I}$ , which describe this relationship for voltage and current sources, respectively. Then the DAE network equation for the unknown  $\mathbf{x} = (\mathbf{u}, \mathbf{j}_L, \mathbf{j}_V)^\top$  is given by

$$\begin{pmatrix} \mathbf{A}_C \mathbf{C} \mathbf{A}_C^\top & 0 & 0 \\ 0 & \mathbf{L} & 0 \\ 0 & 0 & 0 \end{pmatrix} \frac{d\mathbf{x}}{dt} + \begin{pmatrix} \mathbf{A}_R \mathbf{G} \mathbf{A}_R^\top & \mathbf{A}_L & \mathbf{A}_V \\ -\mathbf{A}_L^\top & 0 & 0 \\ -\mathbf{A}_V^\top & 0 & 0 \end{pmatrix} \mathbf{x} + \begin{pmatrix} \mathbf{A}_I \mathbf{t} \\ 0 \\ \mathbf{v} \end{pmatrix} = 0, \quad (6.21a)$$

with consistent initial data

$$\mathbf{x}(t_0) = \mathbf{x}_0. \quad (6.21b)$$

Here,  $\mathbf{C} \in \mathbb{R}^{n_C \times n_C}$ ,  $\mathbf{L} \in \mathbb{R}^{n_L \times n_L}$  and  $\mathbf{G} \in \mathbb{R}^{n_G \times n_G}$  are the capacitance, inductance and conductance matrices, which are assumed to be symmetric and positive-definite.

We apply to (6.21a) the procedure described in the previous section. We pose:

$$\mathbf{A}_0 = \begin{pmatrix} \mathbf{A}_C \mathbf{C} \mathbf{A}_C^\top & 0 & 0 \\ 0 & \mathbf{L} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{B}_0 = \begin{pmatrix} \mathbf{A}_R \mathbf{G} \mathbf{A}_R^\top & \mathbf{A}_L & \mathbf{A}_V \\ -\mathbf{A}_L^\top & 0 & 0 \\ -\mathbf{A}_V^\top & 0 & 0 \end{pmatrix},$$

and

$$\mathbf{q} = \begin{pmatrix} \mathbf{A}_I \mathbf{t} \\ 0 \\ \mathbf{v} \end{pmatrix}.$$

We denote by  $\mathbf{Q}_C$  the projector onto the kernel of  $\mathbf{A}_C^\top$ , and set  $\mathbf{P}_C = \mathbf{Id} - \mathbf{Q}_C$ , such that  $\mathbf{P}_C \mathbf{Q}_C = \mathbf{Q}_C \mathbf{P}_C = \mathbf{0}$ . Then we can write:

$$\mathbf{Q}_0 = \begin{pmatrix} \mathbf{Q}_C & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \mathbf{Id} \end{pmatrix}, \quad \mathbf{P}_0 = \begin{pmatrix} \mathbf{P}_C & 0 & 0 \\ 0 & \mathbf{Id} & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

and we find

$$\mathbf{A}_1 = \mathbf{A}_0 + \mathbf{B}_0 \mathbf{Q}_0 = \begin{pmatrix} \mathbf{A}_C \mathbf{C} \mathbf{A}_C^\top + \mathbf{A}_R \mathbf{G} \mathbf{A}_R^\top \mathbf{Q}_C & 0 & \mathbf{A}_V \\ -\mathbf{A}_L^\top \mathbf{Q}_C & \mathbf{L} & 0 \\ -\mathbf{A}_V^\top \mathbf{Q}_C & 0 & 0 \end{pmatrix},$$

$$\mathbf{B}_1 = \mathbf{B}_0 \mathbf{P}_0 = \begin{pmatrix} \mathbf{A}_R \mathbf{G} \mathbf{A}_R^\top \mathbf{P}_C & \mathbf{A}_L & 0 \\ -\mathbf{A}_L^\top \mathbf{P}_C & 0 & 0 \\ -\mathbf{A}_V^\top \mathbf{P}_C & 0 & 0 \end{pmatrix}.$$

A vector  $\mathbf{x}$  belongs to the kernel of  $\mathbf{A}_1$  if and only if:

$$(\mathbf{A}_C \mathbf{C} \mathbf{A}_C^\top + \mathbf{A}_R \mathbf{G} \mathbf{A}_R^\top \mathbf{Q}_C) \mathbf{u} + \mathbf{A}_V \mathbf{j}_V = 0, \quad (6.22)$$

$$-\mathbf{A}_L^\top \mathbf{Q}_C \mathbf{u} + \mathbf{L} \mathbf{j}_L = 0, \quad (6.23)$$

$$-\mathbf{A}_V^\top \mathbf{Q}_C \mathbf{u} = 0. \quad (6.24)$$

Multiplying the first equation by  $\mathbf{u}^\top \mathbf{Q}_C^\top$  from the left, and using the third equation, we find:

$$\mathbf{u}^\top \mathbf{Q}_C^\top \mathbf{A}_R \mathbf{G} \mathbf{A}_R^\top \mathbf{Q}_C \mathbf{u} = 0,$$



which implies

$$\mathbf{A}_R^\top \mathbf{Q}_C \mathbf{u} = 0.$$

In this way, recalling the definition of  $\mathbf{Q}_C$ , we have found

$$\mathbf{Q}_C \mathbf{u} \in \ker \mathbf{A}_C^\top \cap \ker \mathbf{A}_V^\top \cap \ker \mathbf{A}_R^\top \equiv \ker(\mathbf{A}_C, \mathbf{A}_V, \mathbf{A}_R)^\top. \quad (6.25)$$

Then the first equation reduces to:

$$\mathbf{A}_C \mathbf{C} \mathbf{A}_C^\top \mathbf{P}_C \mathbf{u} + \mathbf{A}_V \mathbf{J}_V = 0.$$

Multiplying this equation by  $\mathbf{Q}_C^\top$  from the left, we find

$$\mathbf{Q}_C^\top \mathbf{A}_V \mathbf{J}_V = 0,$$

which implies

$$\mathbf{J}_V \in \ker(\mathbf{Q}_C^\top \mathbf{A}_V). \quad (6.26)$$

Once we have determined  $\mathbf{Q}_C \mathbf{u}$ ,  $\mathbf{J}_V$  from (6.25), (6.26), we can find  $\mathbf{P}_C \mathbf{u}$ ,  $\mathbf{J}_L$  from the remaining equations:

$$\begin{aligned} \mathbf{P}_C^\top \mathbf{A}_C \mathbf{C} \mathbf{A}_C^\top \mathbf{P}_C \mathbf{u} + \mathbf{P}_C^\top \mathbf{A}_V \mathbf{J}_V &= 0, \\ -\mathbf{A}_L^\top \mathbf{Q}_C \mathbf{u} + \mathbf{L} \mathbf{J}_L &= 0, \end{aligned}$$

that is

$$\mathbf{P}_C \mathbf{u} = -(\mathbf{P}_C^\top \mathbf{A}_C \mathbf{C} \mathbf{A}_C^\top \mathbf{P}_C + \mathbf{Q}_C^\top \mathbf{Q}_C)^{-1} \mathbf{P}_C^\top \mathbf{A}_V \mathbf{J}_V, \quad (6.27)$$

$$\mathbf{J}_L = \mathbf{L}^{-1} \mathbf{A}_L^\top \mathbf{Q}_C \mathbf{u}. \quad (6.28)$$

In conclusion, the kernel of  $\mathbf{A}_1$  is:

$$\begin{aligned} \ker \mathbf{A}_1 = & \left\{ \begin{pmatrix} \mathbf{u} \\ \mathbf{J}_L \\ \mathbf{J}_V \end{pmatrix} \in \mathbb{R}^{n+n_L+n_V} \mid \right. \\ & \mathbf{Q}_C \mathbf{u} \in \ker(\mathbf{A}_C, \mathbf{A}_V, \mathbf{A}_R)^\top, \quad \mathbf{J}_V \in \ker(\mathbf{Q}_C^\top \mathbf{A}_V), \\ & \mathbf{J}_L = \mathbf{L}^{-1} \mathbf{A}_L^\top \mathbf{Q}_C \mathbf{u}, \\ & \left. \mathbf{P}_C \mathbf{u} = -(\mathbf{P}_C^\top \mathbf{A}_C \mathbf{C} \mathbf{A}_C^\top \mathbf{P}_C + \mathbf{Q}_C^\top \mathbf{Q}_C)^{-1} \mathbf{P}_C^\top \mathbf{A}_V \mathbf{J}_V \right\}. \end{aligned}$$

Notice that the condition  $\mathbf{J}_V \in \ker(\mathbf{Q}_C^\top \mathbf{A}_V)$  is equivalent to

$$(\mathbf{Q}_C \mathbf{u}) \cdot (\mathbf{A}_V \mathbf{J}_V) = 0 \quad \forall \mathbf{u} \in \mathbb{R}^n,$$

which implies

$$\mathbf{J}_V \in \ker(\mathbf{Q}_C^\top \mathbf{A}_V) \iff \mathbf{A}_V \mathbf{J}_V \in (\ker \mathbf{A}_C^\top)^\perp \equiv \text{im} \mathbf{A}_C. \quad (6.29)$$

If we have the conditions:

$$\ker(\mathbf{A}_C, \mathbf{A}_V, \mathbf{A}_R)^\top = \{0\}, \quad (6.30)$$

$$\ker(\mathbf{Q}_C^\top \mathbf{A}_V) = \{0\}, \quad (6.31)$$

then we find that  $\mathbf{x} \in \ker \mathbf{A}_1$  only if  $\mathbf{Q}_C \mathbf{u} = 0$ ,  $\mathbf{j}_V = 0$ , which, using (6.27), (6.28), imply  $\mathbf{P}_C \mathbf{u} = 0$ ,  $\mathbf{j}_L = 0$ . Thus the conditions (6.30), (6.31) are equivalent to the index-1 condition  $\det \mathbf{A}_1 \neq 0$ . In this case, the differential variable  $\mathbf{y}$  and the algebraic variable  $\mathbf{z}$  are given by

$$\mathbf{y} = \mathbf{P}_0 \mathbf{x} = \begin{pmatrix} \mathbf{P}_C \mathbf{u} \\ \mathbf{j}_L \\ 0 \end{pmatrix}, \quad \mathbf{z} = \mathbf{Q}_0 \mathbf{x} = \begin{pmatrix} \mathbf{Q}_C \mathbf{u} \\ 0 \\ \mathbf{j}_V \end{pmatrix}. \quad (6.32)$$

We notice that, by using (6.29), condition (6.31) is equivalent to

$$\text{im} \mathbf{A}_C \cap \text{im} \mathbf{A}_V = \{0\}. \quad (6.33)$$

Having found the projectors in the linear case, we note that for the nonlinear case we can use the same projectors, as the incidence matrices will always be the same for a given circuit. Hence, we can use the same definition of the differential and algebraic variables as in (6.32). But then we can use the POD error estimations as derived at the beginning of this section.

## 6.5 Experimental validation of the backward reduced adjoint method

Clearly, after having developed two different backward reduced adjoint methods, we would like to perform experiments with these methods. On the one hand to verify that these methods can indeed be used to obtain accurate approximations of the sensitivities desired by designers. On the other hand, to verify the computational advantage over existing methods. And, finally, to assess the differences in accuracy and computational complexity between the two methods.

The aforementioned task is, unfortunately, far from being trivial. For examples of an academic nature, it is possible to use Matlab to obtain the desired results and insights, and we have done so in Chapter 8. However, for truly industrial examples, it is essential to perform the desired analyses in industrial codes. Companies like Infineon and NXP have their own in-house software for circuit simulation, namely Titan [23] and Pstar [34]. These codes are extremely sophisticated, and have many different features. They also contain a full library of models for the nonlinear devices, such as MOS [72] and bipolar transistors. All of these features are usually not present in academically available tools, such as Spice [50, 60] (Simulation Program with Integrated Circuit Emphasis). This is rather unfortunate, as it is clear that the experiments indicated in the above must be carried out in an industrial context, having the right device models available. And, even more important, we should have access to the software in order to make the necessary changes to the code so that both BRAM and BRAM II can be implemented and tested.

Long and detailed discussions at NXP Semiconductors, where the main part of the work in this thesis was carried out, led to the conclusion that an implementation into the circuit simulator would only be feasible by involving local experts in the circuit simulator

Pstar. The time associated with this effort was estimated to be too high, and therefore declined. The alternative of implementing the changes independently was strongly advised against, as the software contains too many features that need to be taken into account.

In view of the foregoing, we had to re-consider how to obtain good indications of the performance of the BRAM and BRAM II methods. In view of the discussion given before in this chapter, we concentrated our attention on the BRAM II method and, more specifically, on the approximation properties of POD for some challenging industrial circuits and the parameter dependence of the POD basis. This will give sufficient insight into the accuracy of the BRAM II method, as argued in section 6.3. It is hoped that the promising results presented in Chapters 7 and 8 will convince developers of Pstar, Titan, Spice and Spectre to implement the BRAM II method.

## **6.6 Summary**

In this chapter, we have introduced a new type of backward adjoint method for calculating sensitivities based upon the POD method applied to the forward problem. The main advantage of this method is that the error structure is less complex than for the original BRAM method. The accuracy of BRAM II can be influenced by the choice of POD basis, and otherwise depends on the sensitivity of this POD basis with respect to the parameters. The latter is, therefore, addressed thoroughly in the following two chapters.

## Chapter 7

# Parameter Dependence of POD basis

We have so far shown the development of sensitivity analysis methods in which we combine the Proper Orthogonal Decomposition method with backward adjoint methods. We have demonstrated how to construct a projection matrix from the POD basis obtained by analyzing the response of an electronic circuit system and then how to apply the projection to obtain a reduced adjoint system. The second backward adjoint reduced method delivers the benefits of having a smaller, less complex system to solve in combination with the extra benefits contributed by the application of the backward adjoint method.

By restricting the total backward adjoint system to the truncated subspace we have also shown that the results obtained from the reduced, by the Galerkin projection, brute force forward sensitivity calculation are identical to the second backward reduced adjoint method which gives the advantages of the adjoint approach to these sensitivity problems and system reduction by projection. It is in BRAM II that we have eliminated the potential for undesired errors in our reverse backward adjoint equation.

The POD basis is the center of our method's added advantages, and having eliminated all other concerns in previous chapters, in this chapter we will focus our discussion on the subject of parameter dependence of the complete set of basis functions of a circuit system, the dependence of the projection matrix which is built up from the dominant system basis functions and the sensitivity value dependence on the parameter when obtained via a model order reduced system.

## 7.1 State sensitivity approximation complications

When solving a circuit system that has been reduced by a projection matrix  $V(\mathbf{p})$  for the system state vectors, each state solution that we find is an approximation to the original full state vector. It is the choice of basis functions we take from the full system basis to construct the projection matrix that determines how accurate this approximate solution is.

Here is an expression of the full state solution, and its approximation:

$$\mathbf{x}(t, \mathbf{p}) \approx V(\mathbf{p})\tilde{\mathbf{x}}(t, \mathbf{p}). \quad (7.1)$$

To explore the influence that the parameter vector  $\mathbf{p}$  has on the projection matrix  $V$ , it is at this point interesting to take the partial derivative of (7.1) with respect to the parameter vector  $\mathbf{p}$ . This is the step taken to find the state sensitivity in the brute force forward method, but this time we do this also to the state approximation on the right hand side:

$$\hat{\mathbf{x}} = \frac{\partial \mathbf{x}(t, \mathbf{p})}{\partial \mathbf{p}} \approx V(\mathbf{p}) \frac{\partial \tilde{\mathbf{x}}(t, \mathbf{p})}{\partial \mathbf{p}} + \frac{\partial V(\mathbf{p})}{\partial \mathbf{p}} \tilde{\mathbf{x}}(t, \mathbf{p}). \quad (7.2)$$

Equation (7.2) demonstrates that the original full space state sensitivity does have a truncated space approximation, but the result we have derived shows that the approximation is not a simple projection, it is a combination of two terms. The first term is the projection of the reduced state sensitivity, which is a straight forward projection of the truncated space sensitivity,

$$V(\mathbf{p}) \frac{\partial \tilde{\mathbf{x}}(t, \mathbf{p})}{\partial \mathbf{p}}. \quad (7.3)$$

The error associated with this term can be made smaller by increasing the number of basis vectors used in the projection. The second term shown is one that is found to also contain the projection matrix, but in this term we find that the projection matrix itself is sensitive to the parameter vector, this leads to an additional effect on the error made. This term is not easily decreased. In fact, even if we take a full basis, this term may be a significant source of errors made. This has to do with the fact that we are using a projection that is based on one fixed set of parameters, usually the nominal ones. Hence, it is important to investigate the behavior of the following term:

$$\frac{\partial V(\mathbf{p})}{\partial \mathbf{p}} \tilde{\mathbf{x}}(t, \mathbf{p}). \quad (7.4)$$

## 7.2 Projection matrices and basis comparison

Projection matrices are constructed from the dominant basis functions chosen after a singular value analysis of a circuit system response. If we make a small change in the parameter of a component in the system we will observe a slightly different response, and also find that the basis functions used to construct the new system solutions also change. We expect that a small change in the physical parameters of a linear component will not affect the basis as much as a change in a non-linear exponentially defined physical parameter. A topologically significant parameter changing event such as a disconnection of a component would have a far greater impact on the system solutions, a parameter event such as this can be seen as having a large sensitive effect on the projection matrix.

In our analysis, and also the expected use of the backward adjoint reduced method, we do not expect parameters to vary much. Often, the area of application is in circuit tuning where gradient-based nonlinear optimization algorithms are used. Usually, first a nominal simulation of the circuit is performed, followed by the optimization procedure. Nevertheless, we have observed cases where the initial parameter settings are dramatically changed during the optimization process. However, a violent parameter failure would be detected in the semiconductor industry as part of yield loss, and not the responsibility of the design and development stage.

We will assume and also restrict parameter changes to relatively small variations, in the order of around 10-15 percent. This is a typical number that is encountered in most circuit optimizations, the larger factors mentioned before are rare. In fact, one could say that in those cases where this occurs, designers have not done their work well enough!

We will apply a principal component analysis to measure the change in the dominant basis functions as we vary parameters and also interchange basis functions and system solutions from simulations of different parameters.

In what follows, we will need the concept of angle between two subspaces. This can be defined as follows. For two subspaces  $\mathbf{F}$  and  $\mathbf{G}$  of a unitary space  $\mathbf{E}^m$  with  $p = \dim\mathbf{F} \geq \dim\mathbf{G} = q \geq 1$  the principal angles  $\theta_1, \dots, \theta_q$  are recursively defined by,

$$\cos(\theta_k) = \underbrace{\max_{\mathbf{u} \in \mathbf{F}}}_{\mathbf{u} \in \mathbf{F}} \underbrace{\max_{\mathbf{v} \in \mathbf{G}}}_{\mathbf{v} \in \mathbf{G}} \mathbf{u}^H \mathbf{v} = \mathbf{u}_k^H \mathbf{v}_k, \|\mathbf{u}\|_2 = 1, \|\mathbf{v}\|_2 = 1, \quad (7.5)$$

subject to  $\mathbf{u}_j^H \mathbf{u}_k = 0, \mathbf{v}_j^H \mathbf{v}_k = 0, j = 1, \dots, k-1$ . The vectors  $\mathbf{u}_1, \dots, \mathbf{u}_q, \mathbf{v}_1, \dots, \mathbf{v}_q$  maximizing (7.5) are named principal vectors, where  $\mathbf{v}_1, \dots, \mathbf{v}_q$  form a unitary basis of  $\mathbf{G}$  and vectors  $\mathbf{u}_{q+1}, \dots, \mathbf{u}_p$  can be found such that  $\mathbf{u}_1, \dots, \mathbf{u}_p$  form a unitary basis.

Assume now that the columns of  $\mathbf{Q}_F$  and  $\mathbf{Q}_G$  form a unitary basis for  $\mathbf{F}$  and  $\mathbf{G}$ , respectively, this representing the full space basis vectors of a system solutions. We then put,

$$\mathbf{M} = \mathbf{Q}_F^H \mathbf{Q}_G \in \mathbf{C}^{p \times q}, \quad (7.6)$$

construct the singular value decomposition, and build the POD basis of  $\mathbf{M}$ . This action will yield the unitary matrices,

$$\tilde{\mathbf{Y}} \in \mathbf{C}^{p \times p}, \mathbf{Z} \in \mathbf{C}^{q \times q}, \quad (7.7)$$

and

$$\tilde{\mathbf{C}} = \begin{pmatrix} \mathbf{C} \\ \mathbf{0} \end{pmatrix} \in \mathbf{C}^{p \times q}, \quad (7.8)$$

with  $\mathbf{C} = \text{diag}\{\sigma_1, \dots, \sigma_q\}$  such that,

$$\mathbf{M} = \tilde{\mathbf{Y}} \tilde{\mathbf{C}} \mathbf{Z}. \quad (7.9)$$

Note that we have  $\tilde{\mathbf{Y}}^H \tilde{\mathbf{Y}} = \tilde{\mathbf{Y}} \tilde{\mathbf{Y}}^H = \mathbf{I}_p$ , and  $\tilde{\mathbf{Z}}^H \tilde{\mathbf{Z}} = \tilde{\mathbf{Z}} \tilde{\mathbf{Z}}^H = \mathbf{I}_q$ . The foregoing expression can be rewritten as

$$\mathbf{M} = \mathbf{Y} \mathbf{C} \mathbf{Z}, \quad (7.10)$$

where  $\mathbf{Y}$  are the first  $q$  columns of  $\tilde{\mathbf{Y}}$  with  $\mathbf{Y} \in \mathbf{C}^{p \times q}$ ,  $\mathbf{C} \in \mathbf{C}^{q \times q}$  and  $\mathbf{Z} \in \mathbf{C}^{q \times q}$ .

Considering the above, the principal angles and vectors of the two systems can be calculated from,

$$\cos \theta_k = \sigma_k, (k = 1, \dots, q) \quad (7.11)$$

$$(\mathbf{v}_1, \dots, \mathbf{v}_q) = \mathbf{V} = \mathbf{Q}_G \cdot \mathbf{Z}, \quad (7.12)$$

$$(\mathbf{u}_1, \dots, \mathbf{u}_q) = \mathbf{U} = \mathbf{Q}_F \cdot \mathbf{Y}. \quad (7.13)$$

We now have a method in which we can compare the principal angles of two subspaces, and this method will be used to compare POD bases that are obtained for a number of industrial examples in order to assess the dependence of such bases on the parameters  $\mathbf{p}$ . Before doing that for one of our industrial test cases, we investigate a special case of the foregoing procedure in order to verify that the method yields appropriate results.

### 7.3 Principal angles of two identical subspaces

To put the method described in the foregoing through a dry run, let's assume that  $\mathbf{Q}_F$  and  $\mathbf{Q}_G$  both span the same subspace (which, however, can be smaller than the dimension of the full space). Our mathematical intuition tells us that the principal angles should all be equal to zero!

In general when comparing two subspaces of two systems we have to deal with the case where we have a mismatch in the number of basis functions,

$$\mathbf{Q}_F \in \mathbf{C}^{m \times m}, \mathbf{Q}_G \in \mathbf{C}^{m \times q}, \quad (7.14)$$

with  $\text{colspan } \mathbf{Q}_F = \text{colspan } \mathbf{Q}_G$ .

Thus, we will have to distinguish two different cases, and this will be done in the following. It will be seen that, in both cases, the result predicted by our mathematical intuition is indeed obtained.

#### 7.3.1 The simple case $q = m$

Here we compare the two vector spaces  $\mathbf{Q}_F$  and  $\mathbf{Q}_G$ , and in this case assume they are of equal size:

$$\mathbf{Q}_F \in \mathbf{C}^{m \times m}, \mathbf{Q}_G \in \mathbf{C}^{m \times m}. \quad (7.15)$$

The columns of  $\mathbf{Q}_F$  and  $\mathbf{Q}_G$  each build a unitary basis,  $\mathbf{Q}_F$  and  $\mathbf{Q}_G$  are unitary matrices, because of this fact we can now take the following steps. The SVD of  $\mathbf{M}$  is found in the following valid way,

$$\mathbf{M} = \mathbf{Q}_F^H \mathbf{Q}_G = \mathbf{Q}_F^H \mathbf{I}_m \mathbf{Q}_G. \quad (7.16)$$

This immediately demonstrates that the singular values are all equal to 1, so that

$$\cos\theta_1 = \dots = \cos\theta_m = 1, \quad (7.17)$$



implying that all of the principal angles are equal to zero:

$$\theta_1 = \dots = \theta_m = 0 \quad (7.18)$$

In this special case, the SVD in full reads,

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}, \quad (7.19)$$

$$\mathbf{V} = \mathbf{Q}_G \mathbf{Q}_G^H = \mathbf{I}_m, \quad (7.20)$$

$$\mathbf{U} = \mathbf{Q}_F \mathbf{Q}_F^H = \mathbf{I}_m. \quad (7.21)$$

### 7.3.2 The general case $q \leq m$

Next we consider the case that  $q \leq m$ ; because of the assumption that the column spans are identical we can write,

$$\mathbf{Q}_F = \mathbf{Q}_G \mathbf{Q}_G^H \mathbf{Q}_F, \quad (7.22)$$

$$\mathbf{Q}_G = \mathbf{Q}_F \mathbf{Q}_F^H \mathbf{Q}_G. \quad (7.23)$$

Because of the following relations,

$$(\mathbf{Q}_F^H \mathbf{Q}_G)(\mathbf{Q}_F^H \mathbf{Q}_G)^H = \mathbf{Q}_F^H \underbrace{\mathbf{Q}_G^H \mathbf{Q}_G^H}_{\mathbf{I}_q} \mathbf{Q}_F = \mathbf{Q}_F^H \mathbf{Q}_F = \mathbf{I}_q, \quad (7.24)$$

$$(\mathbf{Q}_F^H \mathbf{Q}_G)^H (\mathbf{Q}_F^H \mathbf{Q}_G) = \mathbf{Q}_G^H \underbrace{\mathbf{Q}_F^H \mathbf{Q}_F^H}_{\mathbf{I}_q} \mathbf{Q}_G = \mathbf{Q}_G^H \mathbf{Q}_G = \mathbf{I}_q, \quad (7.25)$$

we also have that  $\mathbf{Q}_F^H \mathbf{Q}_G \in \mathbb{C}^{q \times q}$  is unitary. Thus, in this case, we can write down the following singular value decomposition:

$$\mathbf{M} = \mathbf{Q}_F^H \mathbf{Q}_G = \underbrace{\mathbf{Q}_F^H \mathbf{Q}_G}_Y \cdot \underbrace{\mathbf{I}_q}_C \cdot \underbrace{\mathbf{I}_q^H}_{Z^H} \quad (7.26)$$

Using this decomposition we can say that, for the general case, the principal angles are given by,

$$\cos(\theta_1) = \dots = \cos(\theta_q) = 1 \Rightarrow \theta_1 = \dots = \theta_q = 0, \quad (7.27)$$

and the principal vectors are given by,

$$\mathbf{V} = \mathbf{Q}_G \mathbf{I}_q = \mathbf{Q}_G, \quad (7.28)$$

$$\mathbf{U} = \mathbf{Q}_F \mathbf{Q}_F^H \mathbf{Q}_G = \mathbf{Q}_G. \quad (7.29)$$

This shows that, in all cases, the principal angles are equal to zero. This strengthens our confidence in the method proposed for measuring the angles between the subspaces generated by two different POD bases. In the following section, we will use the method for an industrial test case.

## 7.4 Battery example

Having derived a procedure to compare subspaces generated by different sets of principal component vectors generated by applying the proper orthogonal decomposition method, it is time to start considering the variation of the POD basis with respect to parameter changes. As the proof of the pudding is in the eating, we will now consider a challenging industrial circuit for which designers wish to perform an optimization cycle.

The motivation for considering this example is as follows. If you pick up any portable electronic equipment, anything from mobile phones to laptops and modern electronic cars, there is a high chance that within their casing you will find that a lithium-ion battery has been chosen as part of the design. Lithium-ion batteries have many advantages and to name just a few, they can be constructed and shaped to fit any device, they have a high energy to weight ratio, and have low self discharge rate.

These advantages are available in exchange for some manageable risks. Lithium in its pure form is a highly reactive element which makes for a potentially very unstable battery. Lithium-ion batteries are known to become very dangerous and go very wrong very quickly if mistreated, these batteries are responsible for the well known stories in the media of exploding consumer electronics.

Lithium-ion batteries are sensitive to temperature changes and overheating, from external and internal current flows, overcharging and over-discharging. Complicated battery circuit charger circuits have the primary function of watching over and regulating the battery charging process in order to keep it safe. Once this primary target is achieved, then changes in the charging circuit can be designed to increase efficiency in terms of charging time, power output.

The charging circuit is critical to the safety of users. An unexpected variation in a component parameter in the manufacturing stage has the potential to dramatically affect the output, this is one good example of where a full parameter sensitivity analysis of a circuit is very useful.

We have had access to a computer simulated model of a Li-ion charger at NXP semiconductors, and decided to use this circuit to demonstrate the effect of parameter changes on the system basis functions and also demonstrate the feasibility of the backward reduced adjoint method. We would like to see a low sensitivity to parameter changes for the projection matrix.

We ran a simulation between 0ms and 200ms and collected 3500 snapshot circuit states, and we chose a single capacitor for parameter variation by choosing to vary its area. The initial area value was set at 30, and the circuit was re-simulated for area values 32,34,36,38 and 40. We stored the snapshot states for each simulation.

For each parameter value,  $p_v = \{30, 32, 34, 36, 38, 40\}$ , we calculated an initial singular value decomposition, denoted by

$$U_{p_v} S_{p_v} V_{p_v} = W_{p_v} \quad (7.30)$$

We then plotted the singular value distribution of all parameter responses over the time period. Figure 7.1 shows that the first 100 singular values are enough for a good reconstruction, which as a by-product also shows a high potential for the application of the backward reduced adjoint method as the dimension of the problem can be reduced by roughly a factor of 35. The plot also shows that a change in parameter does have an effect on the system response.

We now take a look at the effect of the parameter changes in the rotation of the principal angles, a full basis analysis (i.e. no reduction) is taken on each set of data. We take the parameter value of 30 as our reference parameter and compare the system principal angles, with itself and the response of the rest of the parameters. The results are displayed in Figure 7.2.

When comparing the reference system response with itself as expected there is no rotation (here, the results of Section 7.3 apply), and at this scale it seems there is no rotation at all. Looking at the log scale, see Figure 7.3, we find that there is a small variation in the principal angles when varying the parameter values. This is, however, within machine precision.

Referring back to the minimum number of 100 basis vectors needed to create a good reconstruction of the system response, we now truncate the basis to the first 100 dominant vectors and then compare their rotation. The results are displayed in Figure 7.4. The apparent jump to 90 degree rotations near the cut off point is due to the matrix diagonal zero padding introduced in the general case for principal vector analysis. These large 90 degree rotations are not due to principal vectors influenced by parameter changes and

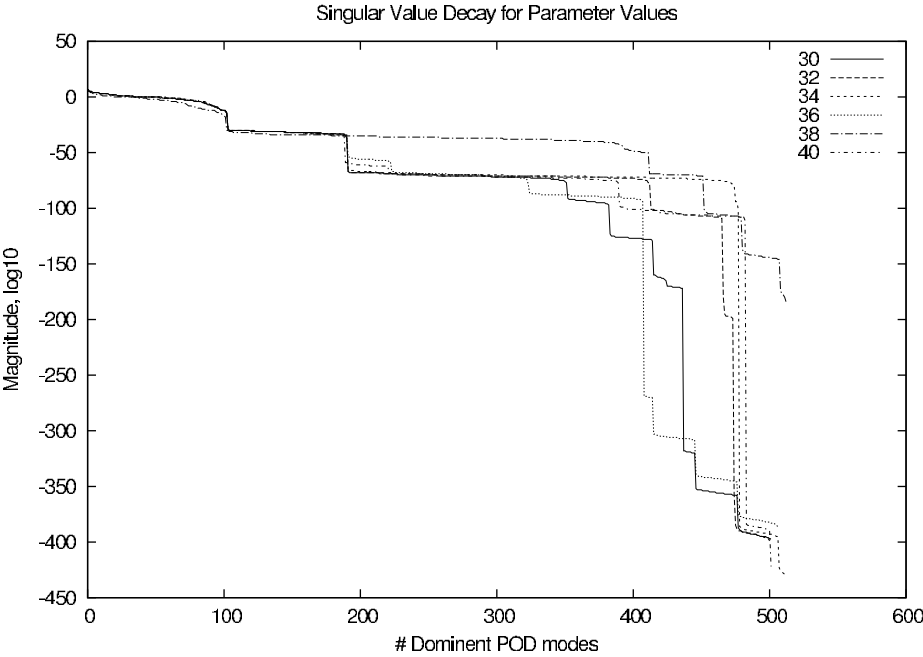


Figure 7.1: Singular values of battery example

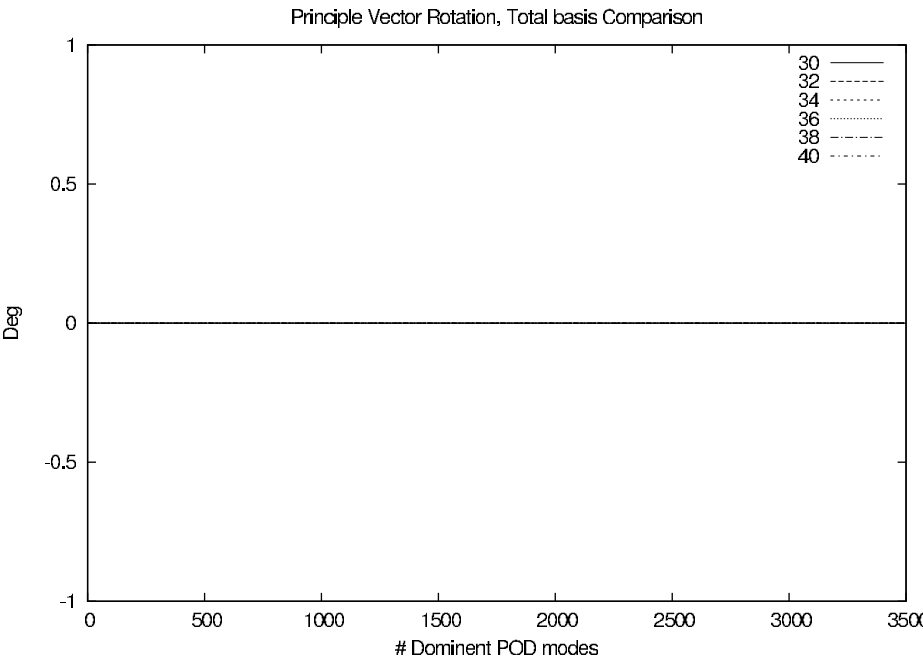


Figure 7.2: Effect of parameter changes

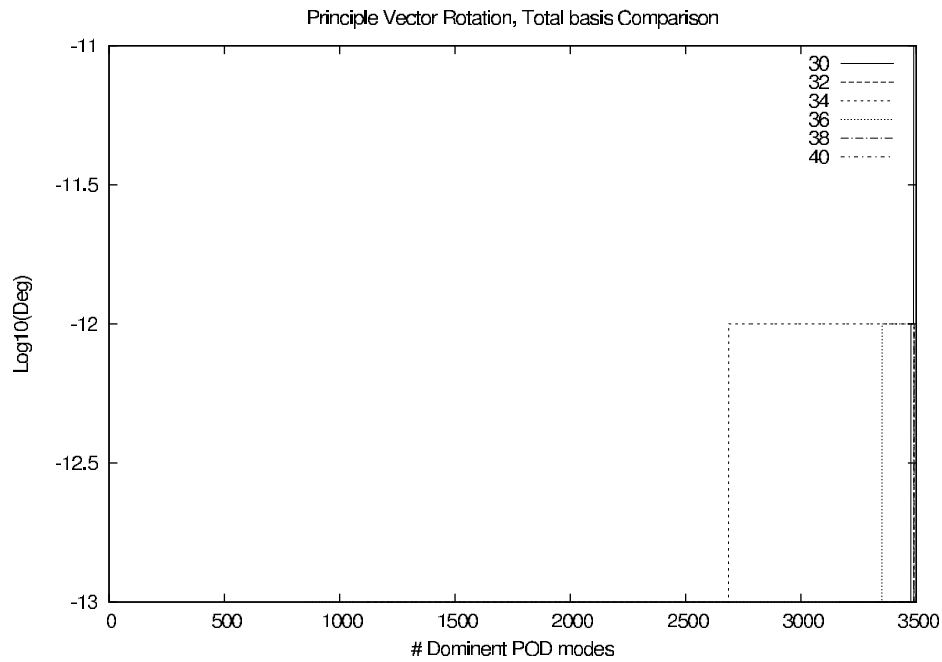


Figure 7.3: Comparison of reference system with itself

should not be taken into account.

Figure 7.5 shows the same data, but now with a logarithmic scale.

We repeat the same experiments, but now for the first 50 dominant basis vectors. Note that this is, in principle, not sufficient to accurately represent the sensitivities, as we neglect too many singular vectors that influence the solution. The idea behind performing this experiment is that we are interested in the most dominant singular vectors. Namely, it may well be that the spaces of 100 singular vectors spans the same subspace, but that the first 50 are gradually changed, and that some of these vectors gradually become the less dominant vectors for different parameter values. Figure 7.6 and Figure 7.7 show the results of these experiments, the latter displaying the results again in a logarithmic scale.

From this initial analysis we have found that for a complicated industrial circuit a change in parameter values does not affect the total set of basis vectors by a disruptive amount, and also not at the point of truncation which is the projector matrix construction step. This sets the parameter sensitivity term to a small amount in the reduced sensitivity analysis equation.

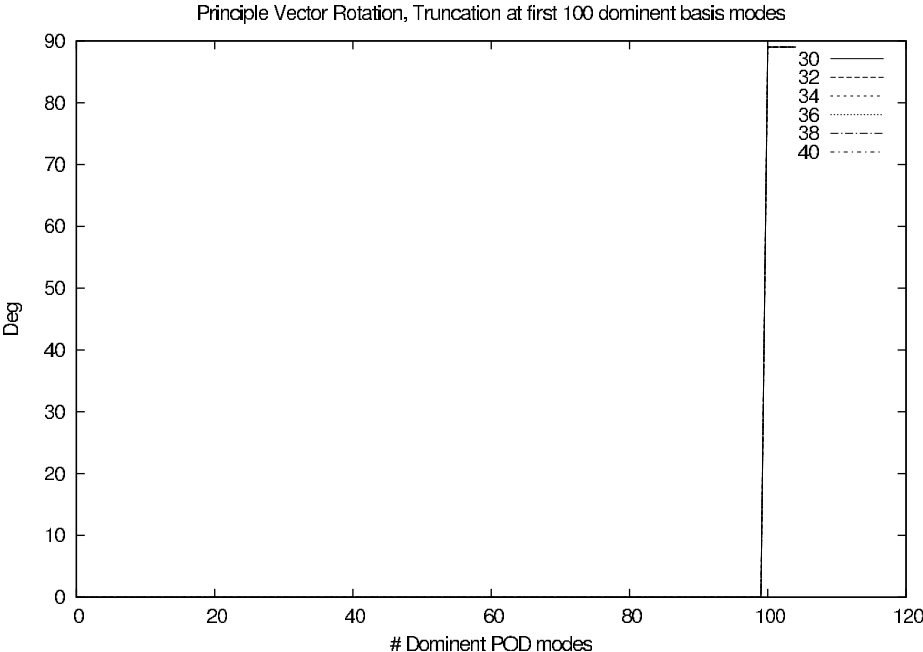


Figure 7.4: PVR100vRD

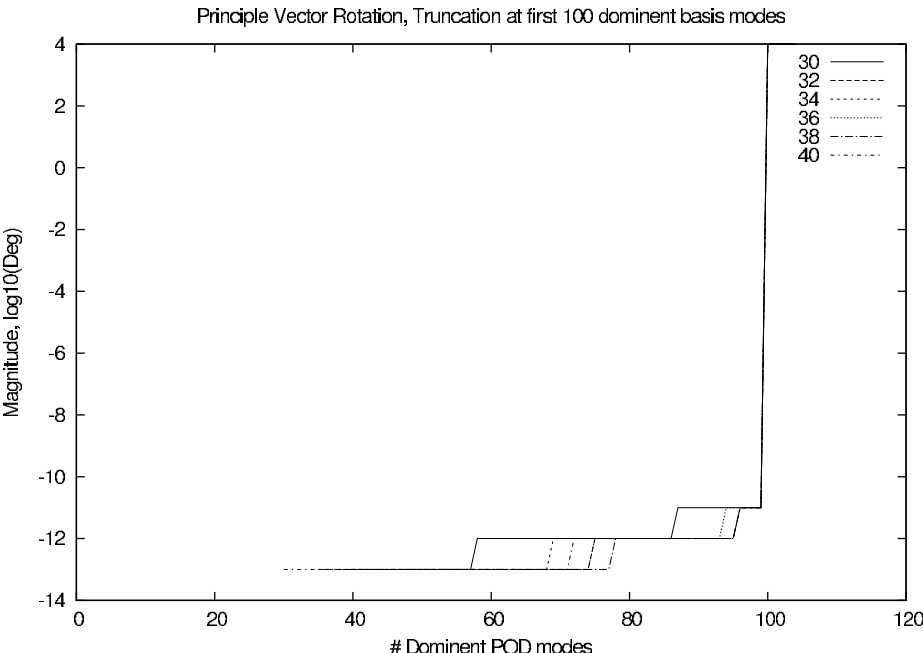


Figure 7.5: Results of Figure 7.4 on logarithmic scale

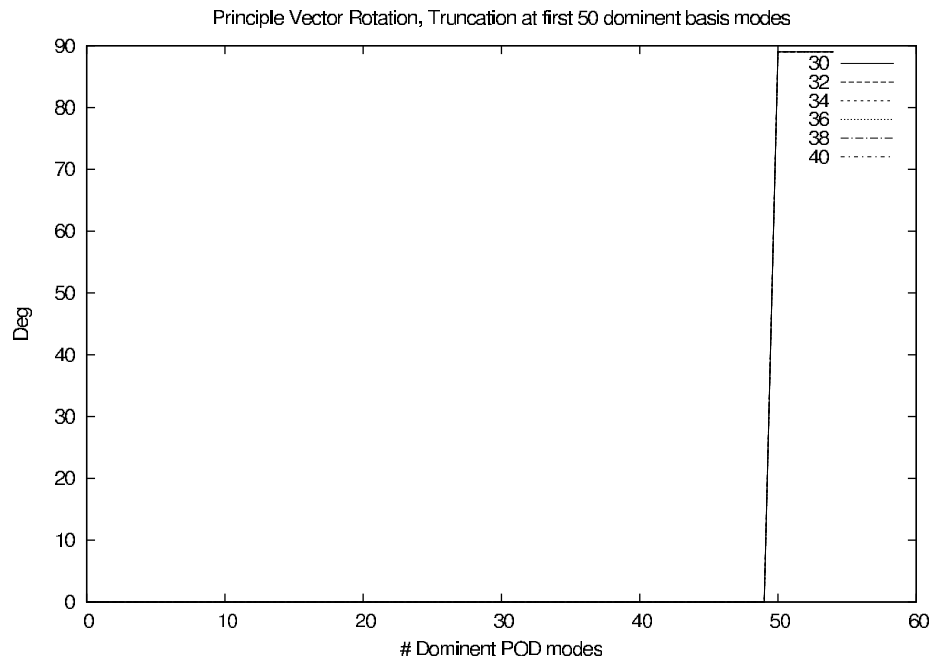


Figure 7.6: Results for truncation at 50 vectors

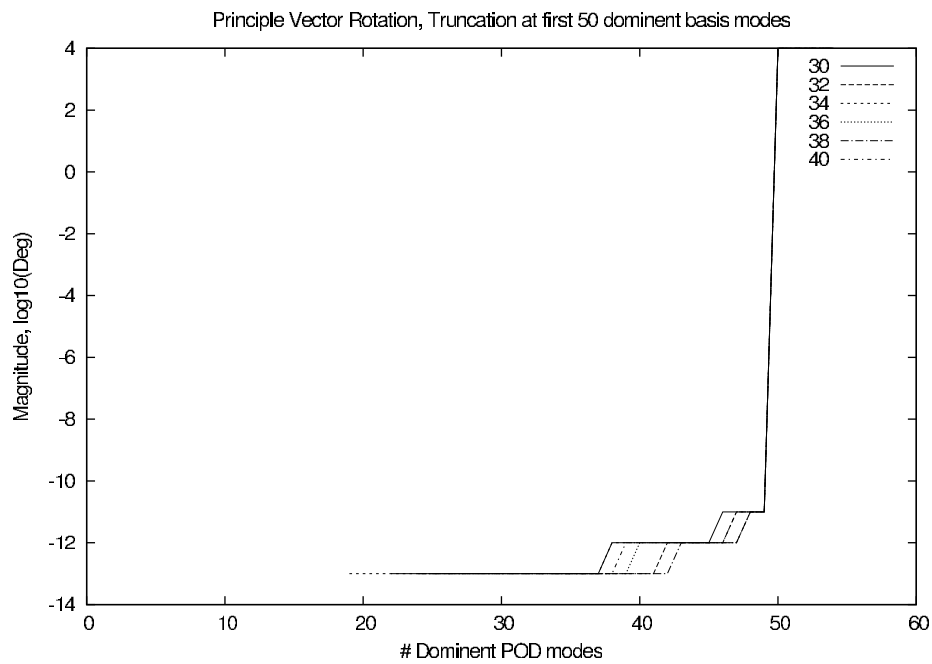


Figure 7.7: Results of Figure 7.6 on logarithmic scale

## 7.5 Summary

In this chapter, we analyzed in more detail the parameter dependence of the POD basis, as this is the main contribution to the error in the BRAM II method. First, we presented a method for determining the angle between two subspaces generated for two different parameter settings. This method was shown to give the expected result in case the two subspaces are equal, i.e. when the bases span the same space. Next, a detailed analysis was provided for a challenging industrial circuit, showing that the subspaces generated for different parameter values are not dramatically different, even for changes of the parameters of up to 30 percent.





## Chapter 8

# Academic & Industrial Examples

Having verified the relative parameter independence of the POD basis, in this chapter we present some additional examples in order to verify whether the backward adjoint methods have potential to yield accurate sensitivity results also for challenging state-of-the-art industrial problems. Before doing that, we first present a simplified example and work this in detail to show the equivalence of sensitivity results obtained for the forward and backward methods. The main aim of this chapter is to analyze the performance of the POD method, with special emphasis on its sensitivity to parameter changes.

### 8.1 A simple test example

We start by considering a simple test circuit for which we can actually perform calculations by hand. This will provide insight in how the computations are done, and will also lead to more insight in why the different sensitivity calculations lead to the same results. In Figure 8.1 we display the circuit under consideration.

We choose to observe the sensitivity of the energy used by the resistor  $R_2$  to changes in the parameter value  $R_2$ . The observation function is given by

$$G_{\text{obs}} = \int_0^T F(t) dt, \quad (8.1)$$

where

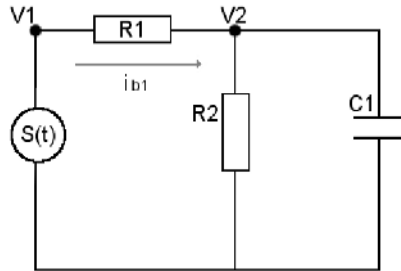


Figure 8.1: Simple circuit for comparison of sensitivities

$$F(t) = \frac{V_2(t)^2}{R_2}. \quad (8.2)$$

We will now calculate the sensitivity in two ways, using the direct forward method and the backward adjoint method, and will demonstrate that the two results are exactly the same. The example shows that, even though it is a rather simple problem, the calculations can be quite involved.

### 8.1.1 Sensitivity - direct forward method

We begin by calculating the direct forward sensitivity. The sensitivity of the observation function with respect to  $R_2$  is given by

$$\frac{dG}{dR_2} = \int_0^T \frac{dF(t)}{dR_2} \cdot dt \quad (8.3)$$

On expanding the integral, we find that

$$\frac{dF(t)}{dR_2} = \frac{1}{R_2} \frac{\partial V_2^2(t)}{\partial R_2} + V_2^2(t) \frac{-1}{R_2^2}$$

Thus, we find the following expression for the direct forward sensitivity:

$$\frac{dG_{\text{obs}}}{dR_2} = \underbrace{\int_0^T \frac{2V_2(t)}{R_2} \frac{\partial V_2(t)}{\partial R_2} dt}_{\text{int 1}} - \underbrace{\int_0^T \frac{V_2^2(t)}{R_2^2} dt}_{\text{int 2}} \quad (8.4)$$

Here we depart temporarily from the forward method, and first calculate the result for the backward adjoint method.

### 8.1.2 Sensitivity - backward adjoint method

We recall the following expression for the sensitivity of the observation function with respect to a parameter  $P$  as found by using the backward adjoint method:

$$\begin{aligned} \frac{dG}{dP} = & \lambda^*(0) \left[ \frac{\partial q(0)}{\partial x} \cdot \hat{x}_{DC} + \frac{\partial q(0)}{\partial P} \right] \\ & + \int_0^T \frac{\partial \lambda^*(t)}{\partial t} \cdot \frac{\partial q}{\partial P} - \lambda^*(t) \left( \frac{\partial j}{\partial P} + \frac{\partial S}{\partial P} \right) dt \\ & + \int_0^T \frac{\partial F(t)}{\partial P} dt, \end{aligned} \quad (8.5)$$

where

$$\hat{x} = \frac{\partial x}{\partial P}. \quad (8.6)$$

As we have chosen  $P = R_2$ , we find

$$\begin{aligned} \frac{dG}{dR_2} = & \lambda^*(0) \left[ \frac{\partial q(0)}{\partial x} \cdot \hat{x}_{DC} + \frac{\partial q(0)}{\partial R_2} \right] \\ & + \int_0^T \frac{\partial \lambda^*(t)}{\partial t} \cdot \frac{\partial q}{\partial R_2} - \lambda^*(t) \left( \frac{\partial j}{\partial R_2} + \frac{\partial S}{\partial R_2} \right) dt \\ & + \int_0^T \frac{\partial F(t)}{\partial R_2} dt. \end{aligned} \quad (8.7)$$

For the given circuit, it can be shown that some of these terms will equal zero. The term  $\frac{\partial q(0)}{\partial x} \hat{x}_{DC}$  is equal to zero, owing to the fact that the DC solution is zero. It is also immediate that  $\frac{\partial q}{\partial R_2} = 0$  and  $\frac{\partial S}{\partial R_2} = 0$ . Thus, the sensitivity expression reduces to

$$\frac{dG_{\text{obs}}}{dR_2} = \int_0^T -\lambda^*(t) \left( \frac{\partial j}{\partial R_2} \right) dt + \int_0^T \frac{\partial F(t)}{\partial R_2} dt. \quad (8.8)$$

Again referring back to the circuit equations, the terms in the first integral can be simplified. The partial derivative w.r.t  $R_2$  of the  $j$  matrix is as follows,

$$j = \begin{pmatrix} \frac{x_1}{R_1} - \frac{x_2}{R_1} - x_3 \\ -\frac{x_1}{R_1} + \frac{x_2}{R_1} + \frac{x_2}{R_2} \\ x_1 \end{pmatrix} \quad (8.9)$$

$$\frac{\partial j}{\partial R_2} = \begin{pmatrix} 0 \\ -\frac{x_2}{R_2^2} \\ 0 \end{pmatrix} \quad (8.10)$$

The partial derivative term in the second integral is expanded as follows,

$$\frac{\partial F(t)}{\partial R_2} = \frac{-V_2^2(t)}{R_2^2} \quad (8.11)$$

Collecting all of these results, it turns out that (8.8) can be written as,

$$\frac{dG}{dR_2} = \int_0^T \lambda_2(t) \frac{V_2(t)}{R_2^2} dt - \int_0^T \frac{V_2^2(t)}{R_2^2} dt. \quad (8.12)$$

### 8.1.3 Comparison of the direct forward and backward sensitivity

Having determined expressions for the sensitivity both for the forward and backward methods, we can now start comparing these results. For convenience, we summarize

the results obtained previously. The sensitivity calculated using the forward method leads to the expression

$$\frac{dG}{dR_2} = \underbrace{\int_0^T \frac{2V_2(t)}{R_2} \frac{\partial V_2(t)}{\partial R_2} dt}_{\text{int 1}} - \underbrace{\int_0^T \frac{V_2^2(t)}{R_2^2} dt}_{\text{int 2}}, \quad (8.13)$$

whereas the sensitivity calculated using the backward approach is given by

$$\frac{dG}{dR_2} = \underbrace{\int_0^T \lambda_2(t) \frac{V_2(t)}{R_2^2} dt}_{\text{int 1}} - \underbrace{\int_0^T \frac{V_2^2(t)}{R_2^2} dt}_{\text{int 2}}. \quad (8.14)$$

At first sight, these expressions do not seem to be equal. The second integral is the same, but the first one looks quite different. Matters would be simple if we would have

$$\lambda_2(t) = 2R_2 \frac{\partial V_2}{\partial R_2},$$

but this is, unfortunately, not the case. Thus, we will need to work the integrals, and compare after having done that. Thus, in the following subsection, we evaluate the integrals.

### Working the forward expression

For the circuit example at  $0 < t < T$ , we have the following:

$$\lambda_2(t) = \beta \left[ \frac{T e^{\alpha(t-T)}}{\alpha^2} + \frac{e^{\alpha(t-2T)}}{2\alpha^3} - \frac{t}{\alpha^2} - \frac{e^{\alpha t}}{2\alpha^3} \right], \quad (8.15)$$

$$V_2(t) = x_2(t) = \frac{A}{C_1 R_1} \left[ \frac{t}{\alpha} - \frac{1}{\alpha^2} + \frac{e^{-\alpha t}}{\alpha^2} \right], \quad (8.16)$$

$$\alpha = \left[ \left( \frac{1}{C_1} \right) \left( \frac{1}{R_1} + \frac{1}{R_2} \right) \right], \quad (8.17)$$

$$\beta = \frac{-2A}{C_1^2 R_2 R_1}, \quad (8.18)$$

$$\frac{\partial V_2(t)}{\partial R_2} = \frac{\partial V_2(t)}{\partial \alpha} \frac{\partial \alpha}{\partial R_2},$$

$$\frac{\partial V_2(t)}{\partial R_2} = \frac{-A}{C_1^2 R_1^2 R_1} \left[ \frac{-t}{\alpha^2} + \frac{2}{\alpha^3} + \frac{te^{-\alpha t}}{\alpha^2} - \frac{2e^{-\alpha t}}{\alpha^3} \right]. \quad (8.19)$$

Using these expressions, the desired integrals can be calculated. First we have that

$$\int_0^T \frac{2V_2(t)}{R_2} \frac{\partial V_2(t)}{\partial R_2} dt = \beta \int_0^T \left[ \frac{t}{\alpha} - \frac{1}{\alpha^2} - \frac{e^{-\alpha t}}{\alpha^2} \right] \left[ \frac{-t}{\alpha^2} + \frac{2}{\alpha^3} + \frac{te^{-\alpha t}}{\alpha^2} - \frac{2e^{-\alpha t}}{\alpha^3} \right] dt \quad (8.20)$$

Using integration by parts a number of times, we find that this equation is equal to the following:

$$\begin{aligned} & \beta \left[ \left[ \frac{t}{\alpha} - \frac{1}{\alpha^2} + \frac{e^{-\alpha t}}{\alpha^2} \right] \left[ \frac{-t^2}{2\alpha^2} + \frac{2t}{\alpha^3} + \frac{te^{-\alpha t}}{\alpha^3} + \frac{e^{-\alpha t}}{\alpha^4} - \frac{2e^{-\alpha t}}{\alpha^4} \right] \right. \\ & - \left[ \frac{1}{\alpha} - \frac{e^{-\alpha t}}{\alpha} \right] \left[ \frac{-t^3}{6\alpha^2} + \frac{2t^2}{2\alpha^3} - \frac{te^{-\alpha t}}{\alpha^4} - \frac{e^{-\alpha t}}{\alpha^5} - \frac{e^{-\alpha t}}{\alpha^5} - \frac{2e^{-\alpha t}}{\alpha^5} \right] \\ & + \left[ \frac{t^3 e^{-\alpha t}}{6\alpha^3} + \frac{3t^2 e^{-\alpha t}}{6\alpha^4} + \frac{6te^{-6\alpha t}}{\alpha^5} + \frac{6e^{-\alpha t}}{6\alpha^6} \right. \\ & \left. \left. - \frac{t^2 e^{-\alpha t}}{\alpha^4} - \frac{2te^{-\alpha t}}{\alpha^5} - \frac{2e^{-\alpha t}}{\alpha^6} + \frac{te^{-2\alpha t}}{2\alpha^5} + \frac{e^{-2\alpha t}}{4\alpha^6} + \frac{e^{-2\alpha t}}{2\alpha^6} + \frac{3e^{-2\alpha t}}{2\alpha^6} \right] \right]_0^T. \end{aligned} \quad (8.21)$$

At  $t = 0$  (8.21) quickly reduces to the following,

$$\beta \left[ \frac{5}{4\alpha^6} \right] \quad (8.22)$$

At  $t = T$  (8.21) becomes,

$$\begin{aligned}
& \beta \left[ \frac{-T^3}{2\alpha^3} + \frac{2T^2}{\alpha^4} + \frac{T^2 e^{-\alpha T}}{\alpha^4} + \frac{3Te^{-\alpha T}}{\alpha^5} + \frac{T^2}{2\alpha^4} - \frac{2T}{\alpha^5} - \frac{Te^{-\alpha T}}{\alpha^5} - \frac{3e^{-\alpha T}}{\alpha^6} \right. \\
& \quad - \frac{T^2 e^{-\alpha T}}{2\alpha^4} + \frac{2Te^{-\alpha T}}{\alpha^5} + \frac{Te^{-2\alpha T}}{\alpha^5} + \frac{3e^{-2\alpha T}}{\alpha^6} + \frac{T^3}{6\alpha^3} - \frac{2T^2}{2\alpha^4} + \frac{Te^{-\alpha T}}{\alpha^5} \\
& \quad + \frac{4e^{-\alpha T}}{\alpha^6} - \frac{T^3 e^{-\alpha T}}{6\alpha^3} + \frac{2T^2 e^{-\alpha T}}{2\alpha^4} - \frac{Te^{-2\alpha T}}{\alpha^5} - \frac{4e^{-\alpha T}}{\alpha^6} + \frac{T^3 e^{-\alpha T}}{6\alpha^3} + \frac{3T^2 e^{-\alpha T}}{6\alpha^4} \\
& \quad + \frac{6Te^{-\alpha T}}{6\alpha^5} + \frac{6e^{-\alpha T}}{6\alpha^6} - \frac{T^2 e^{-\alpha T}}{\alpha^4} - \frac{2Te^{-\alpha T}}{\alpha^5} - \frac{2e^{-\alpha T}}{\alpha^6} + \frac{Te^{-2\alpha T}}{2\alpha^5} + \frac{e^{-2\alpha T}}{4\alpha^6} \\
& \quad \left. + \frac{2e^{-\alpha T}}{\alpha^6} \right] \quad (8.23)
\end{aligned}$$

Combining these results, we finally obtain the following expression for the sensitivity obtained from the direct forward method:

$$\begin{aligned}
& \int_0^T \frac{2V_2(t)}{R_2} \frac{\partial V_2(t)}{\partial R_2} dt = \\
& \beta \left[ \frac{T^3}{\alpha^3} \left[ \frac{-2}{6} + \frac{e^{-\alpha T}}{6} \right] + \frac{T^2}{\alpha^4} \left[ \frac{3}{2} + e^{-\alpha T} \right] + \frac{T}{\alpha^5} \left[ -2 + 4e^{-\alpha T} + \frac{e^{-2\alpha T}}{2} \right] \right. \\
& \quad \left. + \frac{1}{\alpha^6} \left[ \frac{-5}{4} + \frac{5e^{-2\alpha T}}{4} \right] \right] \quad (8.24)
\end{aligned}$$

### Working the backward expression

In a similar way, we can evaluate the integral obtained using the backward adjoint method. Substituting the expressions found at the beginning of the previous subsection, we find

$$\int_0^T \lambda_2(t) \frac{V_2(t)}{R_2} dt$$



$$= \beta \int_0^T \left[ \frac{T e^{\alpha(t-T)}}{\alpha^2} + \frac{e^{\alpha(t-2T)}}{2\alpha^3} - \frac{t}{\alpha^2} - \frac{e^{\alpha t}}{2\alpha^3} \right] \left[ \frac{t}{\alpha} - \frac{1}{\alpha^2} + \frac{e^{-\alpha t}}{\alpha^2} \right] dt \quad (8.25)$$

Again applying two rounds of partial integration, we find

$$\begin{aligned} & \beta \left[ \frac{t}{\alpha} - \frac{1}{\alpha^2} + \frac{e^{-\alpha t}}{\alpha^2} \right] \left[ \frac{T e^{\alpha(t-T)}}{\alpha^3} + \frac{e^{\alpha(t-2T)}}{2\alpha^4} - \frac{t^2}{2\alpha^2} + \frac{e^{-\alpha t}}{2\alpha^4} \right] \\ & - \left[ \frac{1}{\alpha} - \frac{e^{-\alpha t}}{\alpha} \right] \left[ \frac{T e^{\alpha(t-T)}}{\alpha^4} + \frac{e^{\alpha(t-2T)}}{2\alpha^5} - \frac{t^3}{6\alpha^2} - \frac{e^{-\alpha t}}{2\alpha^5} \right] \\ & + \left[ \frac{T e^{-\alpha T}}{\alpha^4} + \frac{t e^{-2\alpha T}}{2\alpha^5} + \frac{t^3 e^{-\alpha t}}{6\alpha^3} + \frac{3t^2 e^{-\alpha t}}{6\alpha^4} + \frac{6t e^{-\alpha t}}{6\alpha^5} + \frac{6e^{-\alpha t}}{6\alpha^6} + \frac{e^{-2\alpha t}}{4\alpha^6} \right] \Big|_0^T \end{aligned} \quad (8.26)$$

### 8.1.4 The final comparison

At this stage it is not immediately clear that (8.27) is equal to (8.21). To show that it is, one needs to fully evaluate (8.27) taking into account the boundary conditions. At  $t = 0$ , (8.27) quickly reduces to,

$$\beta \left[ \frac{5}{4\alpha^6} \right], \quad (8.27)$$

whereas at  $t = T$  (8.27) becomes,

$$\begin{aligned} & \beta \left[ \frac{T}{\alpha} - \frac{1}{\alpha^2} + \frac{e^{-\alpha T}}{\alpha^2} \right] \left[ \frac{T}{\alpha^3} + \frac{e^{-\alpha T}}{2\alpha^4} - \frac{T^2}{2\alpha^2} + \frac{e^{-\alpha T}}{2\alpha^4} \right] \\ & - \left[ \frac{1}{\alpha} - \frac{e^{-\alpha T}}{\alpha} \right] \left[ \frac{T}{\alpha^4} + \frac{e^{-\alpha T}}{2\alpha^5} - \frac{T^3}{6\alpha^2} - \frac{e^{-\alpha T}}{2\alpha^5} \right] \\ & + \left[ \frac{T^2 e^{-\alpha T}}{\alpha^4} + \frac{T e^{-2\alpha T}}{2\alpha^5} + \frac{T^3 e^{-\alpha T}}{6\alpha^3} + \frac{3T^2 e^{-\alpha T}}{6\alpha^4} + \frac{6T e^{-\alpha T}}{6\alpha^5} + \frac{6e^{-\alpha T}}{6\alpha^6} + \frac{e^{-2\alpha T}}{4\alpha^6} \right] \end{aligned} \quad (8.28)$$

which expands to,

$$\begin{aligned}
& \beta \left[ \frac{T^2}{\alpha^4} + \frac{T e^{-\alpha T}}{\alpha^5} - \frac{T^3}{2\alpha^3} - \frac{T}{\alpha^5} - \frac{e^{-\alpha T}}{\alpha^6} + \frac{T^2}{2\alpha^4} + \frac{T e^{-\alpha T}}{\alpha^5} \right] \\
& + \frac{e^{-2\alpha T}}{\alpha^6} - \frac{T^2 e^{-\alpha T}}{2\alpha^4} - \frac{T}{\alpha^5} + \frac{T^3}{6\alpha^3} + \frac{T e^{-\alpha T}}{\alpha^5} - \frac{T^3 e^{-\alpha T}}{6\alpha^3} + \frac{T^2 e^{-\alpha T}}{\alpha^4} \\
& + \frac{T e^{-2\alpha T}}{2\alpha^5} + \frac{T^3 e^{-\alpha T}}{6\alpha^3} + \frac{3T^2 e^{-\alpha T}}{6\alpha^4} + \frac{6T e^{-\alpha T}}{6\alpha^5} + \frac{6e^{-\alpha T}}{6\alpha^6} + \frac{e^{-2\alpha T}}{4\alpha^6} ]
\end{aligned} \tag{8.29}$$

Thus, collecting terms, we find that

$$\begin{aligned}
& \int_0^T \lambda_2(t) \frac{V_2(t)}{R_2^2} dt = \\
& \beta \left[ \frac{T^3}{\alpha^3} \left[ \frac{-2}{6} + \frac{e^{-\alpha T}}{6} \right] + \frac{T^2}{\alpha^4} \left[ \frac{3}{2} + e^{-\alpha T} \right] \right] \\
& + \beta \left[ \frac{T}{\alpha^5} \left[ -2 + 4e^{-\alpha T} + \frac{e^{-2\alpha T}}{2} \right] + \frac{1}{\alpha^6} \left[ \frac{-5}{4} + \frac{5e^{-2\alpha T}}{4} \right] \right]
\end{aligned} \tag{8.30}$$

which immediately shows that both expressions for sensitivity derived from the backward adjoint method and the direct forward method are equal.

### 8.1.5 The remaining integral

For completeness we will expand the second integral that appears both in (8.13) and in (8.14).

$$- \int_0^T \frac{V_2^2(t)}{R_2^2} dt = \tag{8.31}$$

$$= \gamma \int_0^T \left[ \frac{t}{\alpha} - \frac{1}{\alpha^2} + \frac{e^{-\alpha t}}{\alpha^2} \right] \left[ \frac{t}{\alpha} - \frac{1}{\alpha^2} + \frac{e^{-\alpha t}}{\alpha^2} \right] dt \tag{8.32}$$

where,

$$\gamma = \frac{-A^2}{C_1^2 R_1^2 R_2^2}$$

$$= \gamma \int_0^T \left[ \frac{t^2}{\alpha^2} - \frac{t}{\alpha^3} + \frac{te^{-\alpha t}}{\alpha^3} - \frac{t}{\alpha^3} + \frac{1}{\alpha^4} - \frac{e^{-\alpha t}}{\alpha^4} + \frac{te^{-\alpha t}}{\alpha^3} - \frac{e^{-\alpha t}}{\alpha^4} + \frac{e^{-2\alpha t}}{\alpha^4} \right] dt \quad (8.33)$$

$$= \gamma \left[ \frac{t^3}{3\alpha^2} - \frac{t^2}{2\alpha^3} - \frac{te^{-\alpha t}}{\alpha^4} - \frac{e^{-\alpha t}}{\alpha^5} - \frac{t^2}{2\alpha^3} + \frac{t}{\alpha^4} + \frac{e^{-\alpha t}}{\alpha^5} - \frac{te^{-\alpha t}}{\alpha^4} - \frac{e^{-\alpha t}}{\alpha^5} \right]_0^T$$

$$+ \gamma \left[ \frac{e^{-\alpha t}}{\alpha^5} - \frac{e^{-2\alpha t}}{2\alpha^5} \right]_0^T \quad (8.34)$$

$$= \gamma \left[ \frac{T^3}{3\alpha^2} - \frac{T^2}{\alpha^3} + \frac{T}{\alpha^4} [1 - 2e^{-\alpha T}] + \frac{1}{\alpha^5} \left[ \frac{1}{2} - \frac{e^{-2\alpha T}}{2} \right] \right] \quad (8.35)$$

### 8.1.6 The complete sensitivity expressions

Having performed all of these calculations, and having shown that indeed the sensitivities obtained via the direct forward and the backward adjoint methods are the same, we can now provide the complete expression for the sensitivity of our simple example:

$$\frac{dG}{dR_2} =$$

$$\beta \left[ \frac{T^3}{\alpha^3} \left[ \frac{-2}{6} + \frac{e^{-\alpha T}}{6} \right] + \frac{T^2}{\alpha^4} \left[ \frac{3}{2} + e^{-\alpha T} \right] + \frac{T}{\alpha^5} \left[ -2 + 4e^{-\alpha T} + \frac{e^{-2\alpha T}}{2} \right] \right]$$

$$+ \beta \left[ \frac{1}{\alpha^6} \left[ \frac{-5}{4} + \frac{5e^{-2\alpha T}}{4} \right] \right] \quad (8.36)$$

$$+ \gamma \left[ \frac{T^3}{3\alpha^2} - \frac{T^2}{\alpha^3} + \frac{T}{\alpha^4} [1 - 2e^{-\alpha T}] + \frac{1}{\alpha^5} \left[ \frac{1}{2} - \frac{e^{-2\alpha T}}{2} \right] \right]$$

where

$$\alpha = \left[ \left( \frac{1}{C_1} \right) \left( \frac{1}{R_1} + \frac{1}{R_2} \right) \right],$$

$$\beta = \frac{-2A}{C_1^2 R_2 R_1},$$

$$\gamma = \frac{-A^2}{C_1^2 R_1^2 R_2^2},$$

Thus, even though this concerns a very simple example, the sensitivity is far from being a simple expression in terms of the parameters of the circuit.

## 8.2 A simple rectifier example

We now consider a more challenging, but nevertheless still rather simple, example, namely a rectifier circuit.

The diode bridge rectifier is an electronic circuit used to convert AC voltage to DC voltage, or as near as possible DC signal. Figure (8.3) shows a schematic of the bridge rectifier, a sinusoidal input signal varying from positive to negative is inputted on the left on nodes 1 and 2, on the right a signal of positive values is outputted. The circuit's ability to rectify a sin signal depends upon the properties of the four junction diodes arranged in the centre of the schematic.

Junction diodes are highly non-linear devices, their ideal operating characteristics are that once a forward bias is applied the device switches on and conducts current, a constant voltage drop is observed across the junction. In reverse bias mode, the diode switches off, no current is able to pass through.

Junction diodes can be modeled as non-linear resistors, figure (8.2) shows a simple equivalent model of the junction diode. Equations (8.37) and (8.38) model the diode current and conductance.

$$I_d = I_s (\exp((v_p - v_n)/V_{th}) - 1) + (v_p - v_n)/R_{par} \quad (8.37)$$

$$g_d = I_s \exp((v_p - v_n)/V_{th})/V_{th} + 1/R_{par} \quad (8.38)$$

Figure (8.4) demonstrates the operation of the circuit under a positive input voltage. Diodes 1 and 4 are switched off, diodes 2 and 3 are able to conduct the current. If you follow the polarity of the devices, you will see the output voltage is positive.

Figure (8.5) demonstrates the rectifier response when a negative input signal is applied at the input. Diodes 2 and 3 switch off, diodes 1 and 4 are switched on and are now able to conduct, again if you follow the polarity of the devices you will see that the output

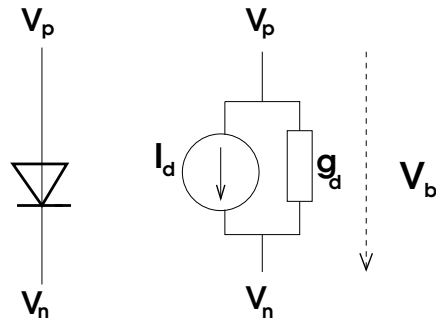


Figure 8.2: PN-Diode

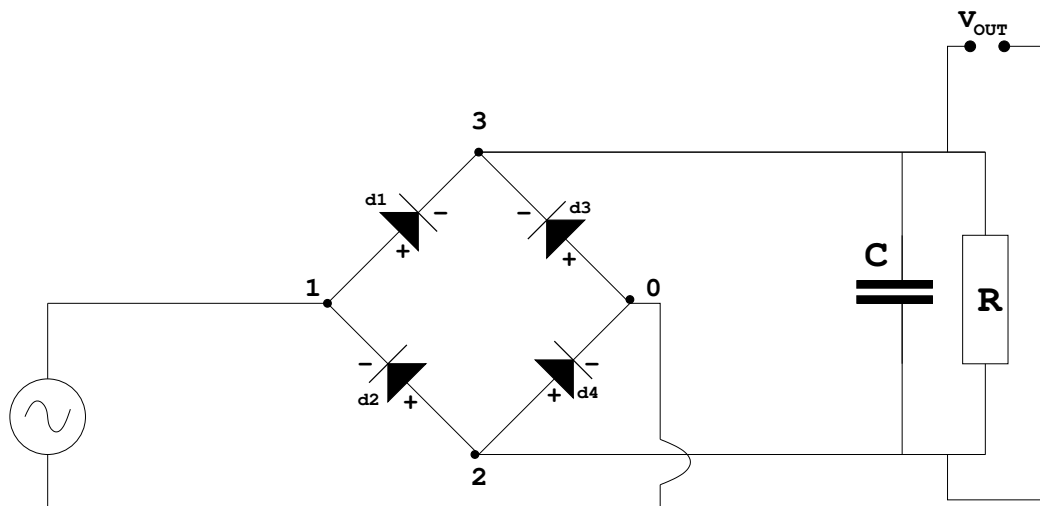


Figure 8.3: Bridge rectifier circuit.

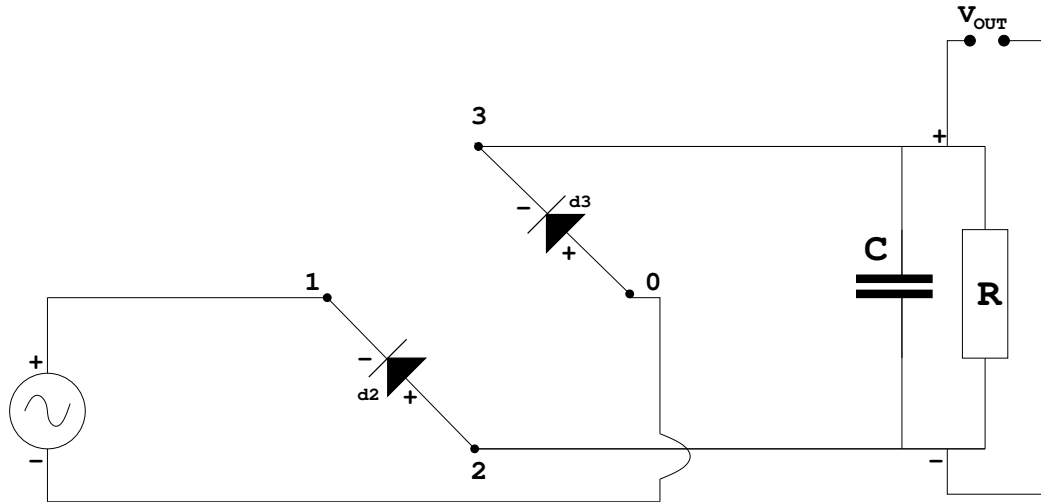


Figure 8.4: Operation under positive half cycle

signal is again positive.

The capacitor at the output is a smoothing capacitor, its smoothing effect will be explained with the aid of some simulation results.

An input sinusoidal signal and its rectified signal is shown in figure (8.6), the peak frequency voltage is doubled and the output signal is diminished by the amount of voltage falling across the diodes. This rectified voltage although non-negative, does not yet resemble a direct current.

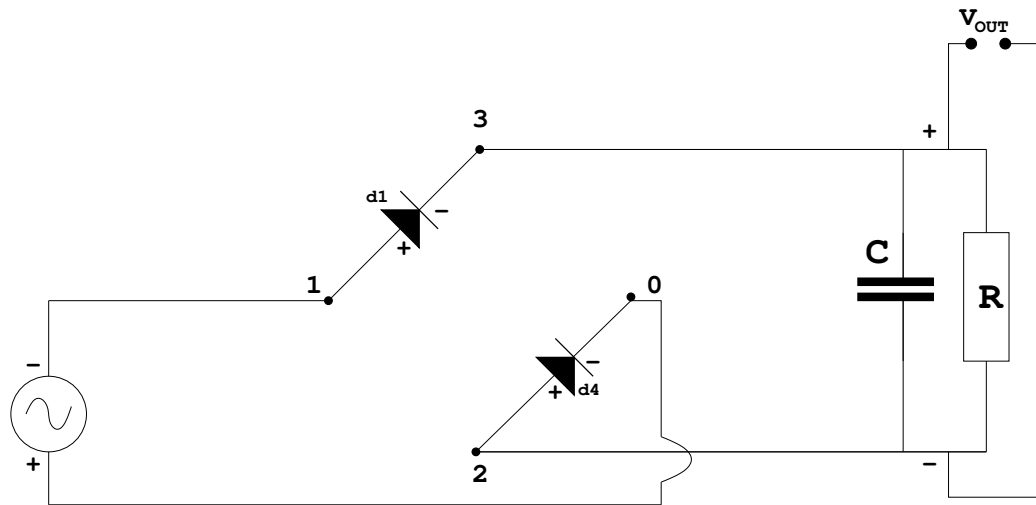


Figure 8.5: Operation under negative half cycle

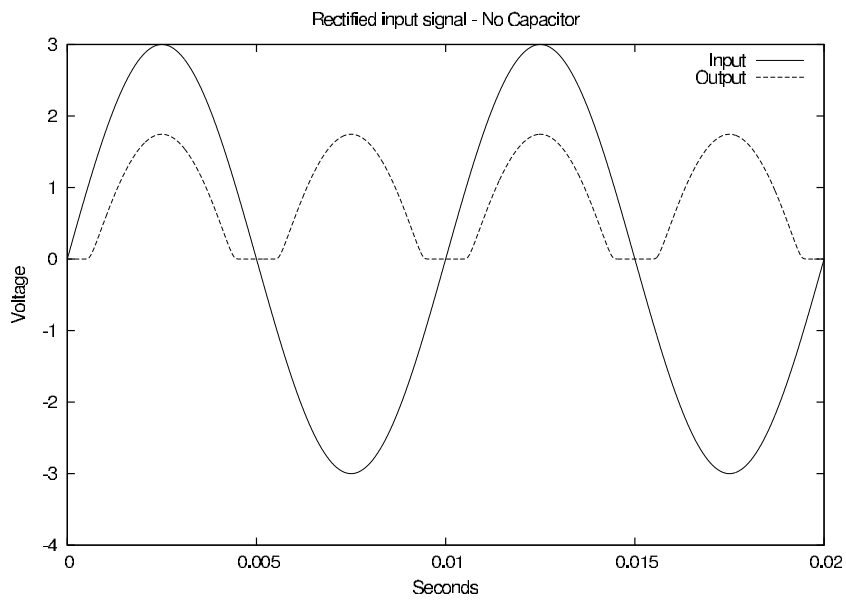


Figure 8.6: Bridge rectifier - no smoothing.

To smooth the signal a capacitor can be added and varied to suit, this is shown in (8.7). On the upward voltage output curve the capacitor charges, once the rectified output voltage curve drops the capacitor discharges at a slower rate which keeps the voltage at a higher level, the capacitor charge is refilled on the next upward cycle. The rectified output signal is now a ripple that looks more like a DC signal. The peaks of the ripples coincide with the peaks of the original rectified signal.

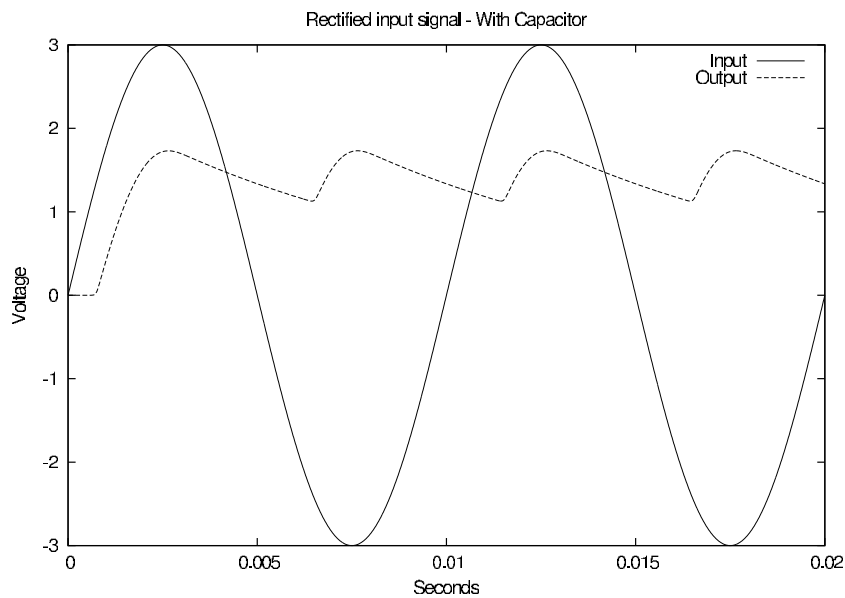


Figure 8.7: Bridge rectifier- with smoothing.

A singular value analysis of the circuit above, with and without a smoothing capacitor gives the distribution in (8.8). You can see that we could reduce the circuit by a POD projector constructed using 3 or even 2 singular values. In figures (8.9) and (8.10) we overlay the circuit output signal, for smoothing and non-smoothing configurations, when solving systems of varying model sizes.

In both cases, a reduction to 3 singular values preserves the circuit behaviour, a reduction to 2 singular values also gives a reasonable preservation of circuit behaviour which would depend upon design tolerances.



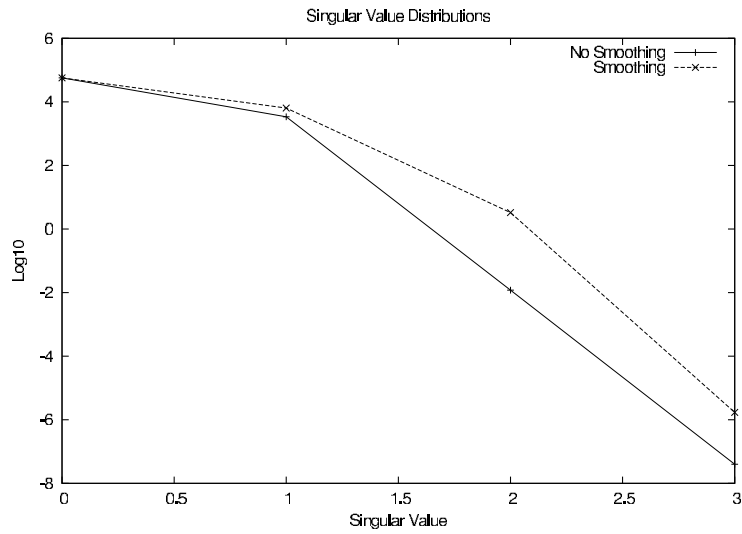


Figure 8.8: Bridge rectifier singular value distribution

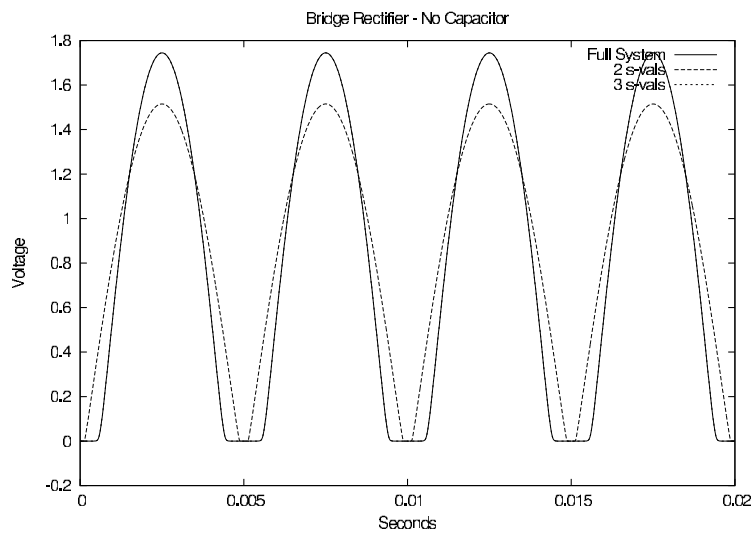


Figure 8.9: Bridge rectifier - no smoothing, reduced models.

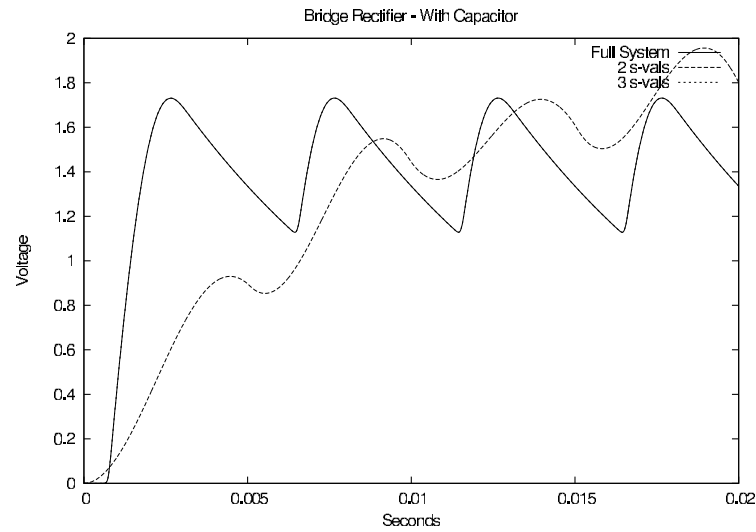


Figure 8.10: Bridge rectifier - with smoothing, reduced models.

### 8.3 Industrial examples

As the methods developed need to be used in an industrial environment with challenging and state-of-the-art problems, designers within NXP Semiconductors were asked to provide a number of examples for which we can put our methods to the test. As the focus within the company is mainly on high performance mixed signal designs rather than the large digital designs that were studied some years ago, we concentrated on the following 3 examples:

- Battery Charger
- Ring Oscillator
- Car Transceiver

These are not the largest of examples, but nevertheless do contain quite a lot of unknowns and many nonlinear devices. The examples can be considered to be representative also for larger examples.

As mentioned in Chapter 6, implementation of BRAM and BRAM II into the in-house circuit simulator Pstar was declined and advised against in view of the effort needed

in programming the methods. These conclusions were mainly a result of the financial crisis, urging managers to carefully consider any new developments and investments. We feel, however, that the results in this section are convincing, and will hopefully lead to a future implementation of the backward reduced adjoint methods. But, for now, we will need to be satisfied with investigations of the performance of POD on these three industrial examples, which will enable us to estimate the speed-up when using the backward reduced adjoint methods.

In the final section, we consider another way of using the sensitivity results, namely as the basis of a reduction technique for parasitic layout networks.

## 8.4 Battery charger

This circuit, that was already presented in Section 7.4, has been designed to charge a Li-Ion battery using a standard USB port, a process that is quite delicate and in need of constant detailed monitoring. Without attention and care the battery and the charger itself could be damaged beyond practical use. The most obvious monitoring requirement is the detection of the battery voltage, to detect when a battery needs charging or when it is full. Other monitoring requirements are the detection of the charge current and the ability to vary the current and various stages of the battery charge level; battery temperature and chip temperature, again used to limit the current trickled or even stop the circuit at limits placed by the designer.

The charge cycle requirements are implemented with a combination of ring oscillators, temperature diode threshold sensors, current detectors and resistive temperature sensors which turns the deceptively simple task of charging a battery into a complex circuit implementation consisting of at least 3495 unknowns demanding an involved time domain simulation.

Figure (8.11) shows the three well defined active states of the battery charger.

1. Trickle charging takes place up to a battery voltage of 2.8V
2. Constant current charge (Fast Charge) takes place between 2.8V and 4.2V
3. Under constant voltage charge the current is down from 100% to 10%

The charging cycle (8.12) begins with `charger_on` at this step a check takes place to see whether or not a supply source is detected and present. If successful, the next step is to check if the attached battery is charged or not. The threshold here is 4.05 volts, if the battery voltage is below this value the battery charging cycle can begin. The following order of checks are designed so as not to damage the battery. The first check taken is to find out if the battery is in a state where trickle charge is the most suited initial charge.

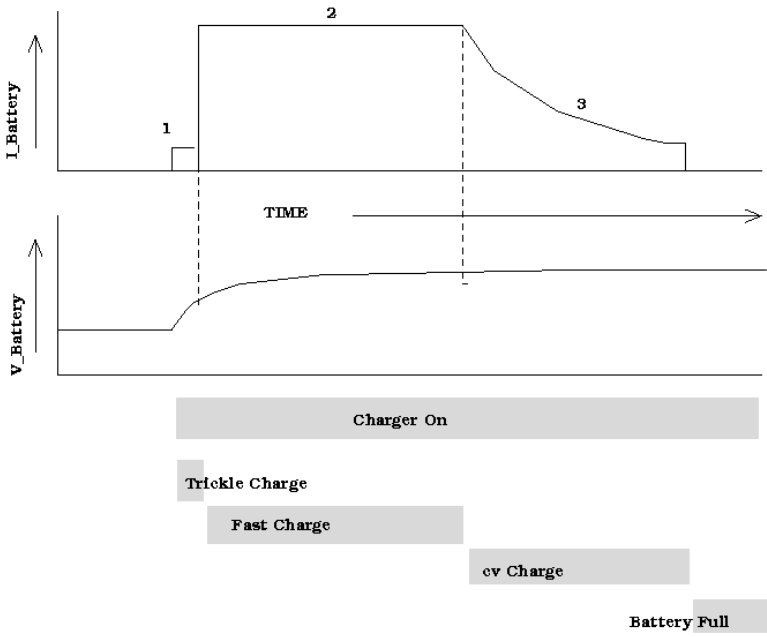


Figure 8.11: Active Charge States

This is done by testing the battery voltage and checking to see if it is below 2.8V. If trickle charge is needed, a constant current of 10% the full normal current charge phase is supplied to the battery. The next voltage level threshold is 2.8V, above this value it is safe to apply the maximum current at a constant value, it is safe to do so up until about 4.2 volts. Above this level the battery of the voltage is held at a constant voltage and the current is varied supply the charge in this final stage. As the battery becomes full, the applied current is reduced and once the battery current decreases below 10% of the nominal charge current the charge process is stopped, the battery is full. Unexpected variations in current supply can occur and this also must be monitored, figure (8.14) shows the flow chart for a subcircuit responsible for this monitoring.

Li-Ion batteries have a built in thermistor to help give an measure on the battery temperature. The set up is as follows, two resistive dividers one not dependent on the temperature and one with negative temperature coefficient thermistor generates a cross-over voltage that can be used to define a temp\_ok signal. (8.13) is a flow chart of the temperature detection circuit requirements.

The charging circuit temperature itself must also be taken care of, and current flows must be limited at higher chip temperatures. Depending upon the battery voltage, the USB voltage and the charge current, the output transistor can dissipate up to 600mW. The decision to operate in a temperature limit mode is taken by referencing the voltage over a bipolar diode (PNP as diode) compared with a temperature threshold level derived from the bandgap voltage. Figure (8.15) shows the flow chart for this subcircuit. As you can tell by these brief descriptions this circuit is quite involved. A schematic is shown in figure (8.16).

#### 8.4.1 Voltage states only: 0-200u

The following figure shows the transient response of the circuit between 0 and 200u, where we made a selection of the states. This gives an impression about the transient behavior of the circuit. The complete transient response can be captured in a snapshot matrix whose structure is a collection of column vectors representing the state of the circuit at each point on the time grid used in the simulation. Simulations were carried out using Pstar, which is NXP's in-house circuit simulator.

$$W = [x(t_1) \dots x(t_n)], W \in \mathbb{R}^{N \times n} \quad (8.39)$$

The dimensions of this snapshot matrix are  $N \times n$  where  $N$  is defined as the number of unknown circuit states and  $n$  is the number of time grid points. In this case,  $N = 3459$  and  $n$  was taken to be 502.

The singular value decomposition analysis of the system snapshot matrix  $W \in \mathbb{R}^{N \times n}$  reveals the complete basis  $U$  spanning the circuit response space, the singular values  $S$  and the coordinate data  $V$ . Figure 8.18 shows the log plot of the circuit singular values in descending order of importance.

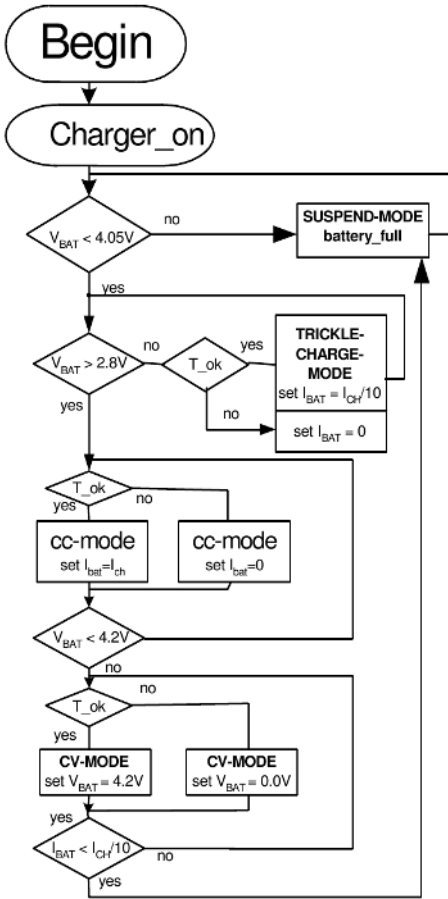


Figure 8.12: Charging Process Flow Chart

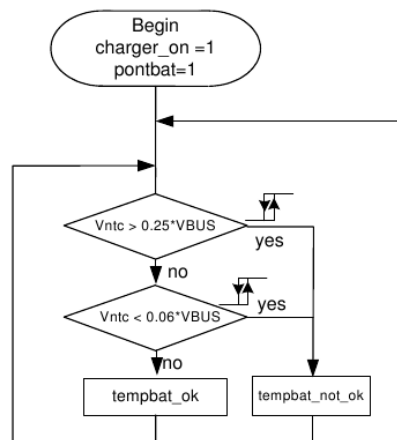


Figure 8.13: Battery Temperature Detection Flow Chart

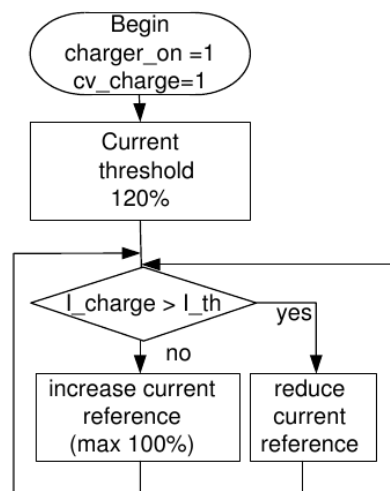


Figure 8.14: Current Monitor Flow Chart

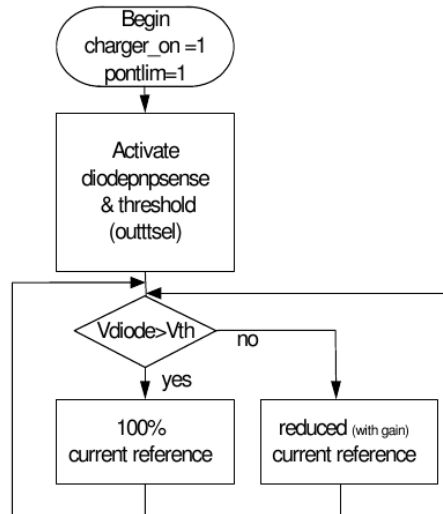


Figure 8.15: Chip Temperature Flow

It is clear from the decay of these singular values some are more dominant than others; each of these values has an associated dominant basis vector which suggests that the complete system can be approximated by a subset of the most dominant components. Therefore, the POD approximation of the system response is done by using the  $K$  most dominant basis vectors to reconstruct the snapshot matrix from the singular value decomposition  $W_k = U(:, 1 : K)S(:, 1 : K)V(:, 1 : K)$ . The reconstructed snapshot can be compared with the original by taking a vector induced matrix norm of the difference  $W - W_k$ , this represents the error of reconstruction from the first  $K$ th dominant singular values. This error, using the 2-norm, is plotted against the index value of the associated  $K$ th singular value. This error analysis shows clearly that an 80 percent data reduction is possible or in other words only the first 100 dominant POD modes are needed. A similar plot, using the infinity norm, on the next page is in agreement.

In fact, these results imply that the original system can be represented by approximately 100 POD basis vectors, and this indeed leads to a significant reduction of the full system of equations.



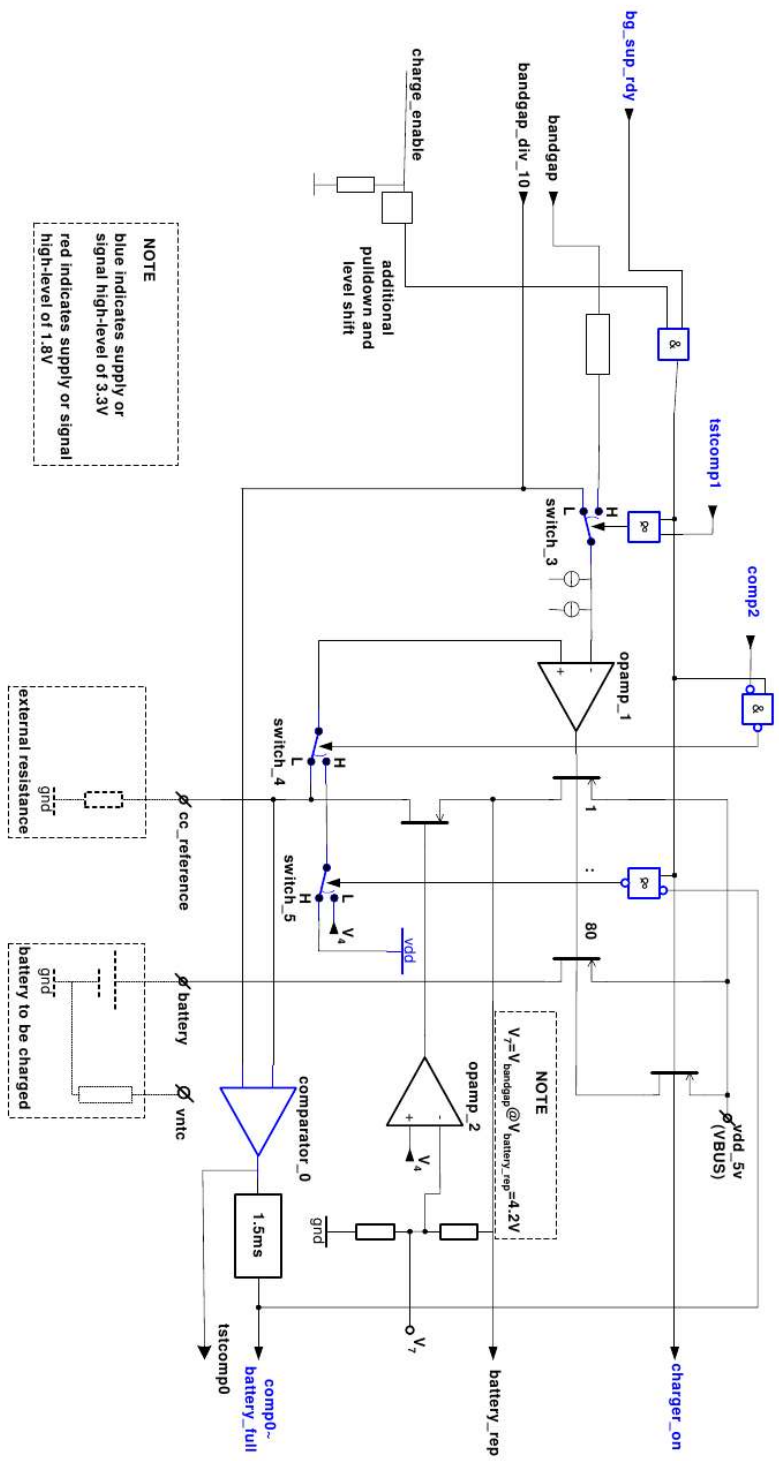


Figure 8.16: Battery Schematic

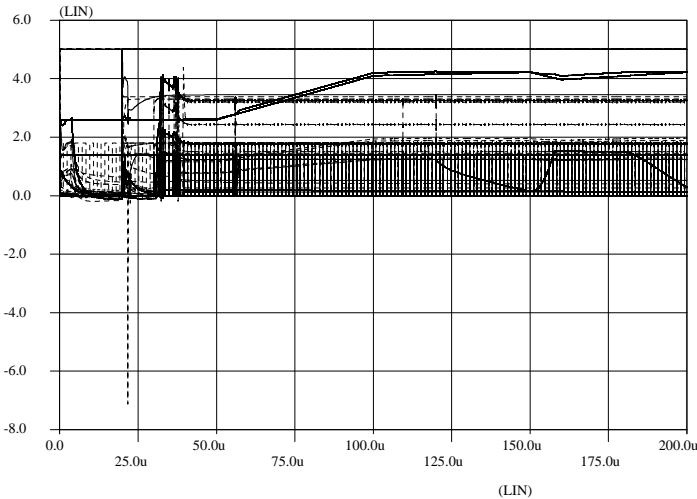


Figure 8.17: Transient response of battery charger circuit

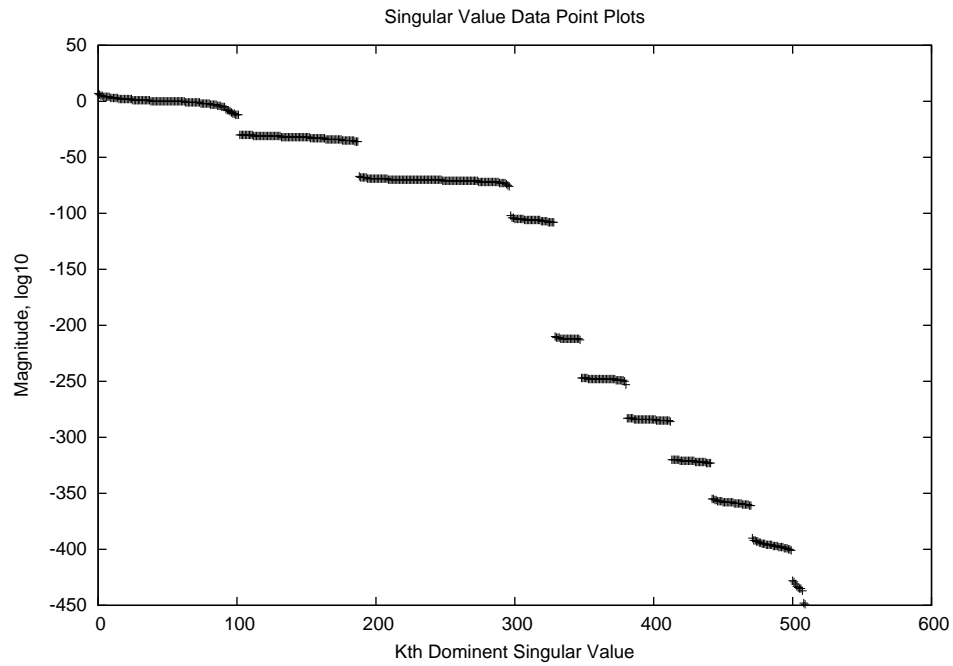


Figure 8.18: Decay of singular values for battery charger circuit

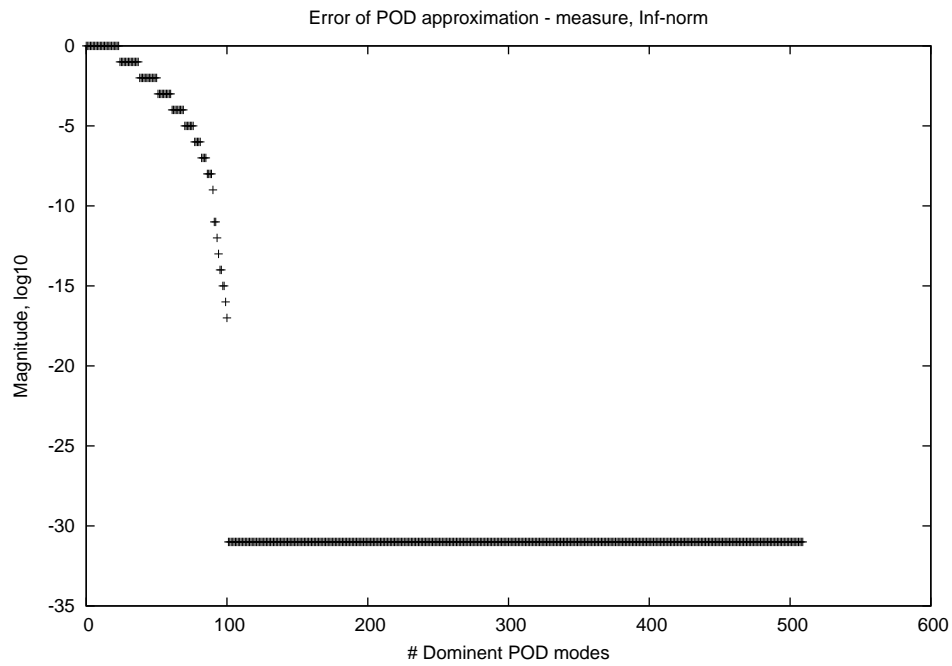


Figure 8.19: Error of POD approximation in 2-norm

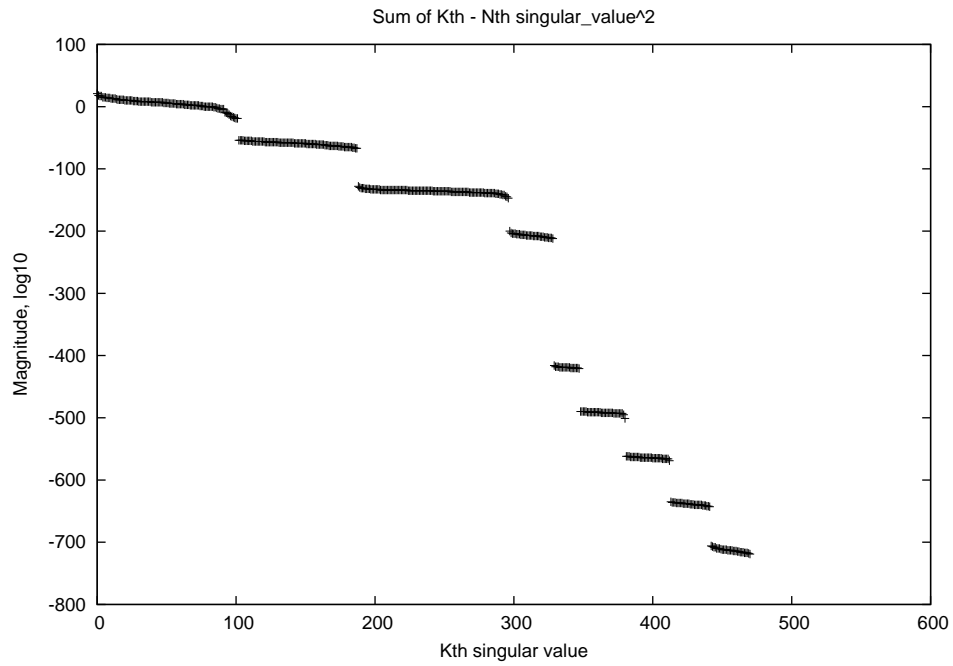


Figure 8.20: Error of POD approximation in  $\infty$ -norm

## 8.5 Ring Oscillator

The next example is a much smaller circuit, as the state vector contains 252 unknowns. But again, approximately 500 data samples are taken for SVD analysis. In the first figure, we show the transient response of the circuit as simulated by the NXP in-house simulator Pstar.

Next we look at the singular values that were calculated for the snapshot matrix, these are displayed in Figure 8.22. Figures 8.23 and 8.24 show the error of the POD approximation in the 2-norm and  $\infty$ -norm, respectively.

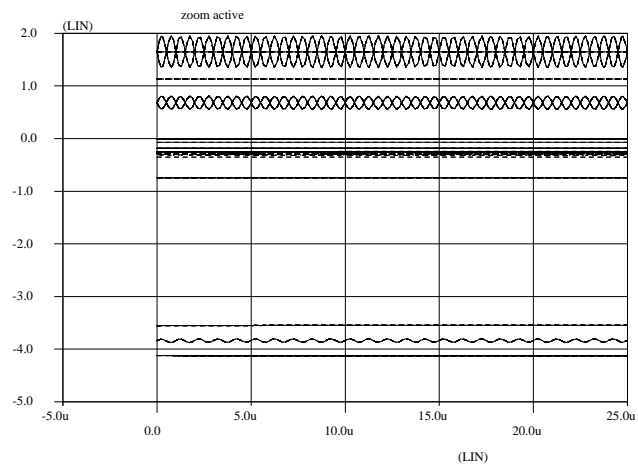


Figure 8.21: Transient response of ring oscillator

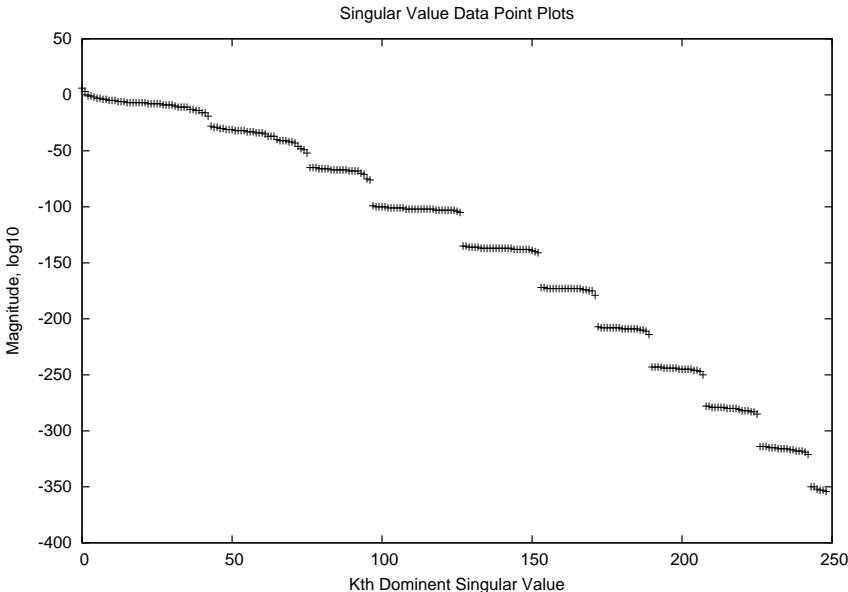


Figure 8.22: Decay of singular values for ring oscillator

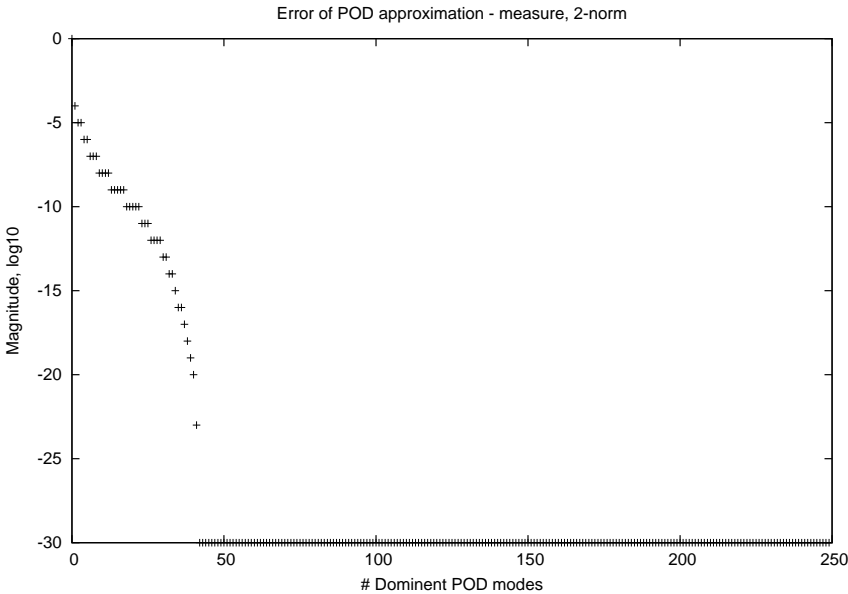


Figure 8.23: Error of POD approximation for ring oscillator (2-norm)

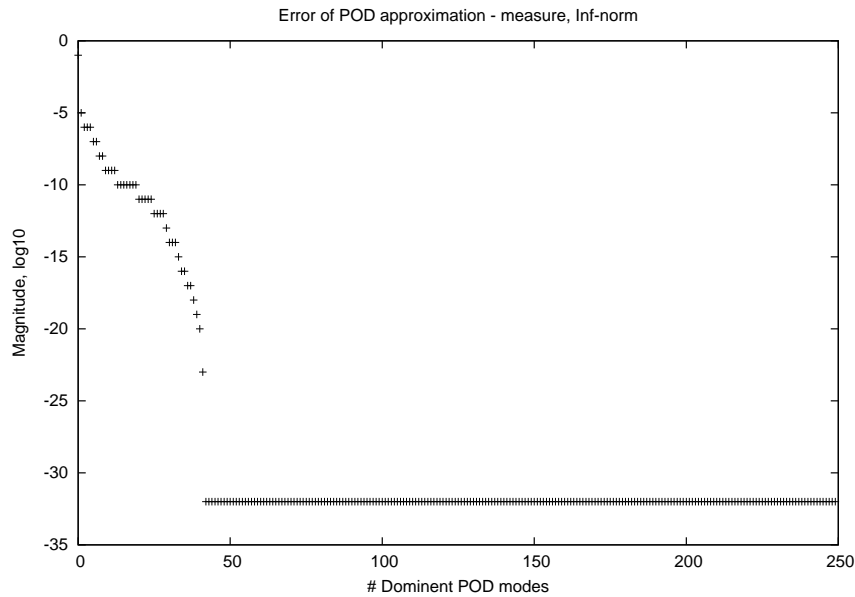


Figure 8.24: Error of POD approximation for ring oscillator ( $\infty$ -norm)

Also for this example we find a drastic reduction of the POD basis is possible, from the full basis to a reduced one. The original matrix contains more than 500 snap shots, but it turns out that 45 basis vectors is more than enough for an accurate modeling of the ring oscillator.

## 8.6 Car Transceiver

Our final industrial example is a car transceiver, again a bit smaller than the previous example with a state Vector containing 126 unknowns. Just as for the previous 2 industrial examples, the simulator Pstar determines approximately 500 time points within the simulation of the circuit. Thus, we have just over 500 snap shots for which we perform an SVD.

First we present the transient response; as can be seen from Figure 8.25, the circuit has most of its activity in the range up to 60u, and then becomes steady.

As for our previous two industrial examples, we now examine the singular values (Figure 8.26) and the error for the POD approximation, both in the 2-norm (Figure 8.27) and the  $\infty$ -norm (Figure 8.28).

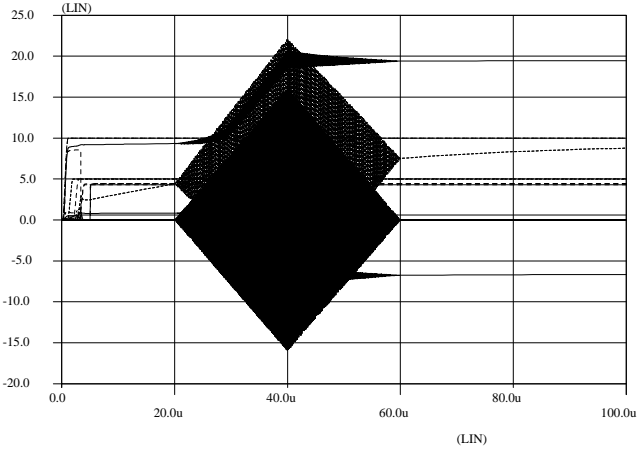


Figure 8.25: Transient behavior of car transceiver



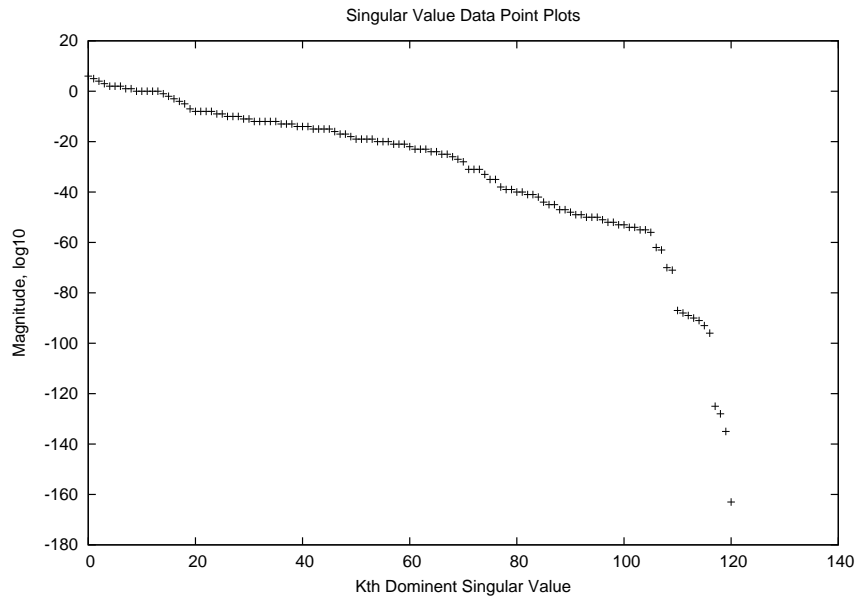


Figure 8.26: Singular values for car transceiver

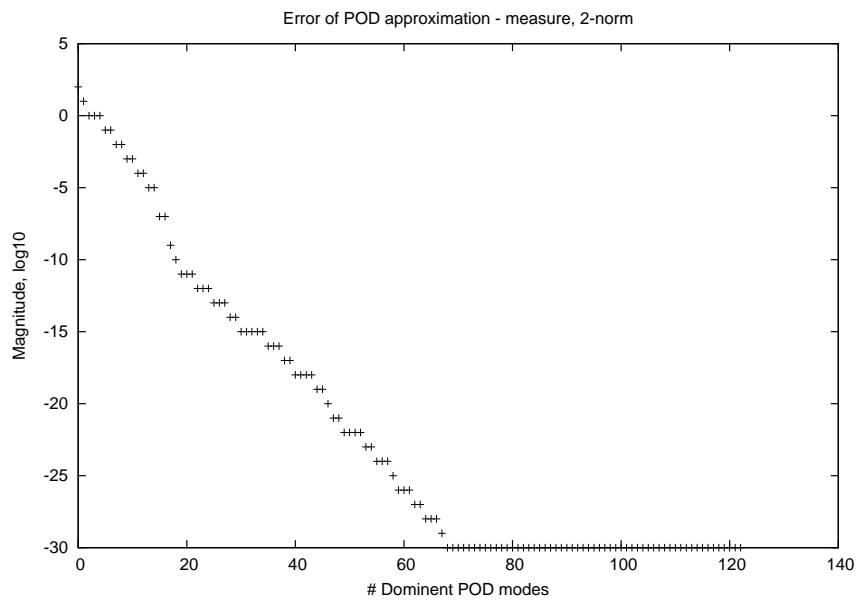


Figure 8.27: 2-norm of POD error for car transceiver

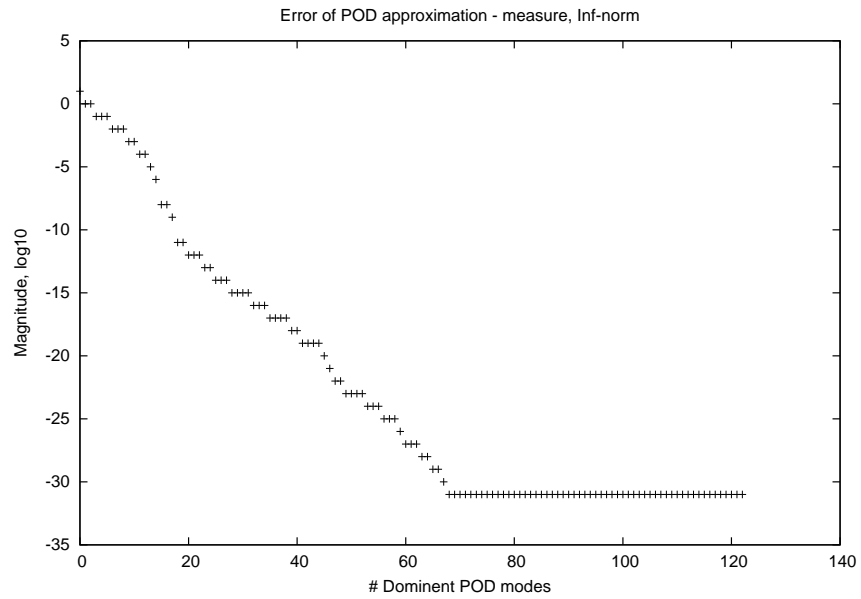


Figure 8.28:  $\infty$ -norm of POD error for car transceiver

The conclusion in this case is that the POD basis need only contain at most 60 elements, but we can probably do very well with only 20 in view of the rapid decay of the error. It is clear that this, again, provides a very significant saving, and will lead to reduced order models that are extremely efficient for this problem.

## 8.7 An alternative reduction technique using sensitivities

Within the COMSON project (see Section 1.2), also other reduction techniques have been investigated. One technique that has been quite effective makes use of the calculated sensitivities in order to decide which parameters are important, and which are less important. Thus, the emphasis is not on the efficient calculation of the sensitivities, but merely on the reduction of the total coupled circuit including parasitic effects.

The approach starts with the simulation of the original circuit, in conjunction with an extraction (via the commercially available tool Assura, see <http://www.cadence.com>) of the layout in order to account for electromagnetic effects due to the interconnect structure (see Chapter 1 for a brief explanation). Next, a sensitivity analysis is performed for the coupled circuit. After this has been done, a list of parasitic elements with the highest contribution to the circuit behavior is selected, based upon the results of the sensitivity analysis. This information is used to perform an aggressive reduction of the circuit to achieve a major reduction in complexity. Clearly, this reduction process will not affect the parameters which were found to have a significant impact on the circuit behavior.

The resulting reduced (original + parasitic) circuit can then be fed back into multiple simulations in which transistor parameters are varied, as well as inputs to the system. As the number of cases to be simulated is increasing with each technology generation, a significant speed-up can be achieved by this approach.

The approach was tested on an industrial benchmark, namely a block in a DRAM design whose task is to correct a duty cycle of a clock signal. The frequency of the input and output signals is 400 MHz, and the time needed to stabilize the output is 300 nanoseconds. The goal of the optimized design is to achieve a duty cycle of 50 percent. The latter is measured by performing two measurements during a simulation: the output clock period  $T$ , and the time  $T_1$  between an rising and a falling edge. The duty cycle is then simply given by  $T_1/T$ .

The original circuit was coupled with a circuit describing the extracted layout, which contained many parasitic resistances and coupled capacitors. Then the coupled circuit was simulated, performing the measurements  $T$  and  $T_1$  and also calculating the sensitivities with respect to all resistors and capacitors in the extracted layout. This led to the computation of in total over 60000 sensitivities.

Then the filtering took place: any resistor or capacitor whose sensitivity was more than 10 percent of the highest sensitivity found was considered to be important, and kept. The rest was discarded, and in this way the extracted layout was reduced considerably. The resulting new coupled circuit, i.e. the original one coupled to the reduced extracted one, was then simulated many times in order to optimize the value of the duty cycle. The results of the reduction are shown in Table 8.1.

From the table we see that, even with the drastic reduction, a very good result is obtained. In addition, the reduced circuit runs much faster than the original one. Of

	full circuit	reduced circuit
# resistors	15199	251
# capacitors	144067	2089
$\bar{T}$ (ns)	2500	2492
$T_1$ (ns)	1238	1232
duty cycle	49.52 %	49.44 %

Table 8.1: Result of DRAM reduction

	full circuit	full circuit + sensitivity	reduced circuit
run time	167 min	424 min	35 min
memory	77 Mb	1.3 Gb	27 Mb

Table 8.2: Timing and memory consumption

course, this comes with the price of the sensitivity analysis, but once this has been done and the parasitic netlist has been reduced, the optimization can perform many simulations at a very low cost. This is demonstrated in Table 8.2, where the timings and memory consumption are provided.

## 8.8 Conclusions

The most important conclusion from this chapter is that POD is an excellent method to reduce challenging industrial circuits, and obtain reduced order models (via the Galerkin projection) that lead to very efficient simulations. Having projected the forward problem, the BRAM II method can be used to obtain sensitivities, and it is clear that these are obtained much more efficiently than with the normal backward adjoint method as presented in literature.



## Chapter 9

# Conclusions

In this thesis, we described our efforts to perform more efficient sensitivity analyses for electronic circuits. To this end, two methods have been suggested, both of them based upon the well known backward adjoint method. The new feature introduced is the use of proper orthogonal decomposition (POD), which is a method that is able to produce reduced order models of nonlinear circuits. The first method proposed is the backward reduced adjoint method (BRAM), which consists of using the POD basis determined for the forward simulation problem also for the backward adjoint problem. The second method, BRAM II, is a modification in which the forward problem is reduced by employing a Galerkin projection with the POD basis. In this case, the backward reduced problem will yield the exact sensitivities, whence the error is fully due to the difference of the original and the projected forward problem.

### 9.1 Main results described in the thesis

We arrived at the main results as follows:

- In Chapter 2, we presented the method of Modified Nodal Analysis, leading to a differential algebraic system for nonlinear circuits. One of the most important observations here is that the circuit incidence matrix remains constant, this is vital for the analysis in view of Tellegen's theorem
- Chapter 3 briefly discusses model order reduction as a field of research, and then concentrates on the proper orthogonal decomposition (POD) method that is a vital ingredient for the methods developed in this thesis
- In Chapter 4, we began by showing how adjoints are being formed for differential algebraic systems, and then continued to show how sensitivities can be calculated

using either direct forward methods or backward adjoint methods. The latter class of methods is most attractive for us, as it is very efficient for situations where many parameters are involved. A special derivation was given for the case of the circuit equations, being a DAE of special type

- Chapter 5 then contains a discussion of the first method suggested for reducing the computation time of the backward adjoint method. In this case, we use the POD basis obtained for the forward problem also for the backward adjoint problem. We made plausible that this can be done, i.e. that the POD bases are very similar. This is partly done by showing that Tellegen's theorem applies, and also by showing that the observation function is often a linear combination of the original state vector
- As the error for the BRAM method contains several different contributions, it is hard to analyse this method. For this reason, a second method has been suggested in Chapter 6, BRAM II, that avoids these complications entirely. It comes with a cost, namely an additional forward simulation, but in practice the Galerkin projected forward system may be used anyway as a reduced order model to simulate the forward problem in an optimization loop. The sensitivities obtained from the backward adjoint method in this case are exact, and hence the error of the method can be influenced fully by choosing the set of POD basis vectors appropriately
- As the main error is caused by the POD basis, we analyzed in Chapter 7 how this basis is influenced by the parameters in the problem. Indeed, it is essential for the BRAM II method to work properly, that the POD basis used for the nominal parameter setting coincides (approximately) with the POD basis for other parameter settings, say, up to 30 percent difference. We checked this for an industrial example, and also developed a method of comparing two POD bases
- Chapter 8 then contains a number of challenging industrial examples, all with quite a large proportion of nonlinear components. The experimental results clearly demonstrate the effectiveness of the POD method

## 9.2 Suggestions for future work

This thesis provides a sound basis for a much more effective sensitivity analysis for nonlinear electronic circuits. It will be especially effective in case of design optimization, as the forward problem can be reduced considerably, and sensitivities can be calculated exactly.

What is missing, unfortunately, is a sound implementation of the methods. As has been explained, this was not feasible within the framework of this project due to restrictions in the company. However, enough evidence has been provided so as to be certain that the methods will perform well in an industrial or commercial simulator. This is also the most important message regarding future work:

- implement the possibility to calculate a POD basis
- implement the Galerkin projection using the POD basis
- use the reduced order model in subsequent design optimizations
- use the reduced forward model as the basis for the backward adjoint method to calculate sensitivities

A remaining research topic is to use an adaptive way for determining the POD basis. In this thesis, we stored all time snap shots and calculated the SVD for the corresponding full matrix. Clearly, this may lead to very large systems and may render the SVD impossible. However, a smart way of coping with the snap shots may be very useful, as our examples have clearly shown that the POD basis can be reduced considerably. For example, we may gradually build up the POD basis and check whether new time snap shots are already within the span of the previously generated basis vectors. As this was not our prime research target, this is left for future investigations.





# Bibliography

- [1] T. Ahmed, E. Gad and M. Yagoub, *An adjoint-based approach to computing the time-domain sensitivity of systems described by reduced-order models*, presented at IEEE MTT-S Int. Microwave Symp., Long Beach, 2005.
- [2] T. Ahmed, E. Gad and M.C.E. Yagoub, *An adjoint-based approach to computing time-domain sensitivity of multiport systems described by reduced-order models*, IEEE Trans. Microw. Theory Techn., vol. 53, pp. 3538-3547, 2005.
- [3] A.C. Antoulas: *Approximation of large-scale dynamical systems*, SIAM series on Advances in Design and Control, 2005.
- [4] H.-U. Armbruster, U. Feldmann and M. Frerichs: *Analysis based reduction using sensitivity analysis*, Proc. Signal Processing on Interconnects SPI 2006, IEEE Workshop, Berlin, 2006.
- [5] W.E. Arnoldi: *The principal minimized iteration in the solution of the matrix eigenproblem*, Quart. Appl. Math., vol. 9, pp. 17-29, 1951.
- [6] P. Astrid: *Reduction of process simulation models: a proper orthogonal decomposition approach*, PhD-Thesis, Department of Electrical Engineering, Eindhoven University of Technology, 2004.
- [7] P. Astrid, A. Verhoeven: *Application of least squares MPE technique in the reduced order modeling of electrical circuits*, TU Eindhoven, Center for Analysis, Scientific Computing and Applications, CASA Report 11, 2006.
- [8] J. Baglama and L. Reichel: *Augmented implicitly restarted Lanczos bidiagonalization methods*, SIAM J. Scient. Comput., 2006
- [9] Z. Bai, R. Li and Y. Su: *A unified Krylov projection framework for structure-preserving model reduction*, in: W.H.A. Schilders, H.A. van der Vorst and J. Rommes (eds.), Model order reduction: theory, research aspects and applications, Mathematics in Industry, vol. 13, Springer-Verlag, Berlin, pp. 75-93, 2008.
- [10] K. Balla and R. März: *A unified approach to linear differential algebraic equations and their adjoint equations*, Institute of Mathematics Technical report, Humboldt University, Berlin, 2000.
- [11] K. Balla and R. März: *Linear differential algebraic equations of index 1 and their adjoint equations*, Result in Mathematics, vol. 37, pp. 13-35, 2000.

- [12] O. Balima, Y. Favennec, M. Girault, D. Petit: *Comparison between the modal identification method and the POD-Galerkin method for model reduction in nonlinear diffusion systems*, Int. J. Numer. Meth. Engng., Vol. 67, pp. 895–915, 2006.
- [13] P. Benner, V. Mehrmann and D.C. Sorensen: *Dimension reduction of large-scale systems*, Lecture Notes in Computational Science and Engineering, vol. 45, Springer-Verlag, 2005.
- [14] R.K. Brayton and S.W. Director: *Computation of delay time sensitivities for use in time domain optimization*, IEEE Trans. Circ. Syst., vol. 22, pp. 910-920, 1975.
- [15] B. Burdick: *Generation of optimal test stimuli for nonlinear analog circuits using nonlinear programming and time-domain sensitivities*, Proc. DATE 2001, Mnchen, pp. 603-608.
- [16] S.L. Campbell, N.K. Nichols and W.J. Terrell: *Duality, observability and controllability for linear time-varying descriptor systems*, Circ. Syst. Sign. Proc., vol. 10, pp. 455-470, 1991.
- [17] Y. Cao, S. Li, L. Petzold, R. Serban: *Adjoint sensitivity analysis for differential-algebraic equations: the adjoint DAE system and its numerical solution*, SIAM J. Sci. Comput., Vol. 24-3, pp. 1076–1089, 2002.
- [18] W.-K. Chen: *Mathematics for circuits and filters*, CRC Press, 2000.
- [19] L.O. Chua and P.-M. Lin: *Computer-aided analysis of electronic circuits: algorithms and computational techniques*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [20] E.A. Coddington and N. Levinson: *Theory of ordinary differential equations*, McGraw-Hill, New York, 1955.
- [21] A.R. Conn, P.K. Coulman, R.A. Haring, G.L. Morrill, C. Visweswariah, C.W. Wu: *JiffyTune: circuit optimization using time-domain sensitivities*, IEEE Trans. on CAD of ICs and Systems, Vol. 17-12, pp. 1292–1309, 1998.
- [22] L. Daldoss, P. Gubian, M. Quarantelli: *Multiparameter time-domain sensitivity computation*, IEEE Trans. on Circuits and Systems - I: Fund. Theory and Applics, Vol. 48-11, pp. 1296–1307, 2001.
- [23] G. Denk: *Circuit simulation for nanoelectronics*, in: M. Anile, G. Ali and G. Mascali (Eds.), Scientific Computing in Electrical Engineering, Mathematics in Industry, vol. 9, Springer, Berlin, pp. 13-20, 2006.
- [24] S.W. Director and R.A. Rohrer: *The generalized adjoint network and network sensitivities*, IEEE Trans. Circ. Theory, vol. 16, pp. 318-323, 1969.
- [25] D. Estevez Schwarz and C. Tischendorf: *Structural analysis of electric circuits and consequences for MNA*, Int. J. Circuit Theory Appl., vol. 28, pp. 131–162, 2000.
- [26] Y. Favennec, M. Girault, D. Petit: *The adjoint method coupled with the modal identification method for nonlinear model reduction*, Inverse Probl. in Science and Engng., Vol. 14, No. 3, 153–170, 2006.
- [27] W.F. Feehery, J.E. Tolsma, P.I. Barton: *Efficient sensitivity analysis of large-scale differential-algebraic systems*, Appl. Numer. Maths., Vol. 25, pp. 41–54, 1997.
- [28] P. Feldmann and R. Freund: *Efficient linear circuit analysis by Padé approximation via the Lanczos process*, IEEE Trans. Computer-Aided Design, vol. 14, p. 137–158, 1993.

- [29] R.W. Freund and P. Feldmann, *Small-signal circuit analysis and sensitivity computations with the PVL algorithm*, IEEE Trans. Circuits Syst., vol. 43, pp. 577-585, 1996.
- [30] R.W. Freund: *Structure-preserving model order reduction of RCL circuit equations*, in: W.H.A. Schilders, H.A. van der Vorst and J. Rommes (eds.), *Model order reduction: theory, research aspects and applications*, Mathematics in Industry, vol. 13, Springer-Verlag, Berlin, pp. 49-73, 2008.
- [31] K. Fujimoto and J.M.A. Scherpen: *Singular value analysis and balanced realizations for nonlinear systems*, in: W.H.A. Schilders, H.A. van der Vorst and J. Rommes (eds.), *Model order reduction: theory, research aspects and applications*, Mathematics in Industry, vol. 13, Springer-Verlag, Berlin, pp. 253-272, 2008.
- [32] M. Gerdt, C. Büskens: *Consistent initialization of sensitivity matrices for a class of parametric DAE systems*, BIT, Vol. 42-4, pp. 796-813, 2002.
- [33] K. Glover: *Optimal Hankel-norm approximations of linear multivariable systems and their  $l^\infty$ -error bounds*, Int. J. Control, vol. 39, pp. 115-193, 1984.
- [34] M. Günther, U. Feldmann and J. ter Maten: *Modelling and discretization of circuit problems*, in: W.H.A. Schilders and E.J.W. ter Maten, *Special volume on Numerical Methods in Electromagnetics*, Handbook of Numerical Analysis, vol. XIII, Elsevier North Holland, pp. 523-659, 2005.
- [35] M. Günther (Ed.): *The COMSON Handbook*, to appear with Springer Verlag, Berlin, 2010.
- [36] R. Gurram and B. Subramanyam: *Sensitivity analysis of radial distribution network - adjoint network method*, Electr. Power Energy Syst., pp. 323-326, 1999.
- [37] P.J. Heres: *Robust and efficient krylov subspace methods for model order reduction*, PhD-Thesis, TU Eindhoven Center for Analysis, Scientific Computing and Applications, TU Eindhoven, 2005.
- [38] I. Higuera, R. März and C. Tischendorf: *Numerically well-formulated index-1 DAEs*, Preprint 2001-05, Humboldt-Universität Berlin, Institut für Mathematik, Berlin, 2001.
- [39] M. Hinze and S. Volkwein: *Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition*, Technical Report IMA02-05, KFU Graz, 2005.
- [40] D.E. Hocevar, P. Yang, T.N. Trick, B.D. Epler: *Transient sensitivity computation for MOSFET circuits*, IEEE Trans. on CAD of Integr. Circuits and Systems, Vol. CAD-4, Nr. 4, pp. 609-620, 1985.
- [41] M.E. Hochstenbach, *A Jacobi-Davidson type SVD method*, SIAM J. on Sci. Comp., vol. 23, pp. 606-628, 2001.
- [42] M.E. Hochstenbach, *Harmonic and refined extraction methods for the singular value problem, with applications in least squares problems*, BIT, vol. 44, pp. 721-754, 2004.
- [43] D. Hömberg and S. Volkwein: *Control of laser surface hardening by a reduced-order approach using proper orthogonal decomposition*, Math. Comp. Mod., vol. 38, pp. 1003-1028, 2003.
- [44] C. Homescu, L.R. Petzold and R. Serban: *Error estimation for reduced-order models of dynamical systems*, SIAM J. Numer. Anal., vol. 43, pp. 1693-1714, 2005.

- [45] S.H.M.J. Houben: *Circuits in motion: the numerical simulation of electrical oscillators*, PhD Thesis, Eindhoven University of Technology, 2003.
- [46] Z. Ilievski, H. Xu, A. Verhoeven, E.J.W. ter Maten, W.H.A. Schilders and R.M.M. Mattheij, *Adjoint Transient Sensitivity Analysis in Circuit Simulation*, in: D. Ioan and G. Ciuprina (Eds.), Proc. SCEE 2006 Conference, Sinaia, Romania, Springer-Verlag, Berlin, 2007.
- [47] R. Khazaka, P. Gunupudi and M.S. Nakhla: *Efficient sensitivity analysis of transmission-line networks using model reduction techniques*, IEEE Trans. Microw. Theory Tech., vol. 48, pp. 2345-2351, 2000.
- [48] E. Kleihorst, *Frequency domain analysis for nonlinear electronic circuits*, Ph.D. Thesis, Delft University of Technology, 1994.
- [49] L. Knockaert and D. De Zutter: *Laguerre-SVD reduced-order modeling*, IEEE Trans. Microwave Theory and Techn., vol. 48, pp. 1469-1475, 2000.
- [50] K.S. Kundert, *The Designers Guide to SPICE and SPECTRE*, Kluwer Academic Publishers, Boston, 1998.
- [51] K. Kunisch and S. Volkwein: *Crank-Nicholson Galerkin proper orthogonal decomposition approximations for a general equation in fluid dynamics*, Proc. 18th GAMM seminar, Leipzig, pp. 97-114, 2002.
- [52] C. Lanczos: *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards, vol. 45, pp. 225-280, 1950.
- [53] C. Lanczos: *Solution of systems of linear equations by minimized iteration*, J. Res. Nat. Bur. Standards, vol. 49, pp. 33-53, 1952.
- [54] H.-J. Lee, C.-C. Chu and W.-S. Feng: *Interconnect modeling and sensitivity analysis using adjoint networks reduction technique*, in: Proc. Int. Circuits Systems Symp., vol. IV, Bangkok, pp. 648-651, 2003.
- [55] W. Magnus and W. Schoenmaker: *Introduction to electromagnetism*, in: W.H.A. Schilders and E.J.W. ter Maten, Special volume on Numerical Methods in Electromagnetics, Handbook of Numerical Analysis, vol. XIII, Elsevier North Holland, pp. 3-103, 2005.
- [56] R. März: *Canonical projectors for linear differential algebraic equations*, Computers Math. Appl., vol. 31, pp. 121-135, 1995.
- [57] R. März: *Differential algebraic systems anew*, Institute of Mathematics Technical report, Humboldt University, Berlin, 2000.
- [58] R.M.M. Mattheij, P.M.E.J. Wijckmans: *Sensitivity of solutions of linear DAE to perturbations of the system matrices*, Numer. Alg., Nr. 19, pp. 159-171, 1998.
- [59] B.C. Moore: *Principal component analysis in linear systems: controllability, observability, and model reduction*, IEEE Trans. Automatic Control, vol. 26, pp. 17-31, 1981.
- [60] L.W. Nagel and D.O. Pederson: *SPICE: Simulation Program with Integrated Circuit Emphasis*, Memorandum No. ERL-M382, University of California, Berkeley, 1973.
- [61] T.V. Nguyen, A. Devgan and O.J. Nastov: *Adjoint transient sensitivity computation in piecewise linear simulation*, Proc. DAC 98, June 15-19, San Francisco, pp. 477-482, 1998.

- [62] G. Obinata and B.D.O. Anderson: *Model reduction for control system design*, Springer-Verlag, London, 2001.
- [63] A. Odabasioglu, M. Celik and L.T. Pileggi: *PRIMA: passive reduced-order interconnect macromodeling algorithm*, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 17, pp. 645-654, 1998.
- [64] P. Penfield, Jr., R. Spence and S. Duinker: *Tellegen's theorem and electrical networks*, M.I.T Press, Cambridge, Mass., 1970.
- [65] P. Penfield, Jr., R. Spence and S. Duinker: *A generalized form of Tellegen's theorem*, IEEE Trans. Circuit Theory, vol. 17, pp. 302-305, 1970.
- [66] L.T. Pillage and R.A. Rohrer: *Asymptotic waveform evaluation for timing analysis*, IEEE Trans. Comp.-Aided Design Int. Circ. Syst., vol. 9, pp. 352-366, 1990.
- [67] L.T. Pillage, R.A. Rohrer, C. Visweswariah: *Electronic circuit and system simulation methods*, McGraw-Hill, Inc, New York, USA, ISBN 0070501696, 1994.
- [68] M. Rathinam, L.R. Petzold: *A new look at proper orthogonal decomposition*, SIAM J. Numer. Analysis, Vol. 41-5, pp. 1893-1925, 2003.
- [69] M.J. Rewienski: *A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems*, PhD-Thesis, Massachusetts Institute of Technology, 2003.
- [70] J. Rommes and W.H.A. Schilders: *Efficient methods for large resistor networks*, IEEE Trans. Comput.-Aided Design Int. Circ. Syst., vol. 29, pp. 28-39, 2010.
- [71] J.M.A. Scherpen: *Balancing for nonlinear systems*, PhD thesis, Univ. of Twente, 1994.
- [72] A.J. Scholten, G.D.J. Smit, B.A. De Vries, L.F. Tiemeijer, J.A. Croon, D.B.M. Klaassen, R. van Langevelde, X. Li, W. Wu and G. Gildenblat: *The new CMC standard compact MOS model PSP: advantages for RF applications*, IEEE J. Solid-State Circ., vol. 44, pp. 1415-1424, 2009.
- [73] W.H.A. Schilders, H.A. van der Vorst and J. Rommes: *Model order reduction: theory, research aspects and applications*, Mathematics in Industry, vol. 13, Springer-Verlag, Berlin, 2008.
- [74] R. Serban, C. Homescu and L.R. Petzold: *The effect of problem perturbations on nonlinear dynamical systems and their reduced-order models*, SIAM J. Sci. Comput., vol. 29, pp. 2621-2643, 2007.
- [75] G. Shi, B. Hu, C.-J. R. Shi: *On symbolic model order reduction*, IEEE Trans. on CAD of Integr. Circuits and Systems, Vol. 25-7, pp. 1257-1272, 2006.
- [76] L. Sirovich: *Turbulence and the dynamics of coherent structures. Part I: Coherent structures*, Quart. Appl. Math., vol. 45, pp. 561-571, 1987.
- [77] M.N.S. Swamy and K. Thulasiraman: *Graphs, networks and algorithms*, New York, Wiley Interscience, 1981.
- [78] B.D.H. Tellegen: *A general network theorem, with applications*, Philips Res. Rep., vol. 7, pp. 259-269, 1952.
- [79] C. Tischendorf, *Solution of index-2-DAEs and its application in circuit simulation*, Ph.D. thesis, Humboldt-Univ. zu Berlin, 1996.
- [80] C. Tischendorf *Topological index calculation of DAEs in circuit simulation*, Surv. Math. Ind., vol. 8, pp. 187-199, 1999.

- 
- [81] A. Verhoeven, E.J.W. ter Maten, M. Striebel, R.M.M. Mattheij: *Model order reduction for nonlinear IC models*, TU Eindhoven, Center for Analysis, Scientific Computing and Applications, CASA Report 12, 2007.
- [82] A. Verhoeven, M. Striebel, J. Rommes, E.J.W. ter Maten and T. Bechtold: *Proper orthogonal decomposition model order reduction of nonlinear IC models*, Proc. SCEE 2006 Conference, Sinaia, Romania, Springer-Verlag, Berlin, 2007.
- [83] A. Verhoeven: *Redundancy reduction of IC models by multirate time-integration and model order reduction*, PhD Thesis, TU Eindhoven, 2008.
- [84] E.I. Verriest: *Time variant balancing and nonlinear balanced realizations*, in: W.H.A. Schilders, H.A. van der Vorst and J. Rommes (eds.), *Model order reduction: theory, research aspects and applications*, Mathematics in Industry, vol. 13, Springer-Verlag, Berlin, pp. 213–250, 2008.
- [85] S. Volkwein: *Error estimates for POD Galerkin schemes*, PhD program in Mathematics for Technology, Catania, 2007.
- [86] T. Voss: *Model reduction for nonlinear differential algebraic equations*, MSc. Thesis University of Wuppertal, 2005; Unclassified Report PR-TN-2005/00919, Philips Research Laboratories, 2005.
- [87] K. Willcox and J. Peraire: *Balanced model reduction via the proper orthogonal decomposition*, AIAA, vol. 40, pp. 2323-2330, 2002.
- [88] H. Xu: *Transient Sensitivity Analysis in Circuit Simulation*, MSc-Thesis, Department of Mathematics and Computing Science, Eindhoven University of Technology, 2004. [Online via author search at Library at <http://w3.win.tue.nl/en/>]
- [89] X. Ye, P. Li and F.Y. Liu: *Exact time-domain second-order adjoint-sensitivity computation for linear circuit analysis and optimization*, IEEE Trans. Circuits Syst., vol. 57, pp. 236-249, 2010.

# Summary

The electronics industry provides the core technology for numerous industrial innovations. Progress in the area of microelectronics is highlighted by several milestones in chip technology, for example microprocessors and memory chips. The ongoing increase in performance and memory density would not have been possible without the extensive use of computer simulation techniques, especially electronic circuit simulation. The basis of the latter is formed by a sound framework of methods from the area of numerical methods.

In recent years, the demands on the capabilities of circuit simulation have become even more stringent. Circuit simulators have become the core of all simulations within the electronics industry. Crosstalk effects in interconnect structures are modeled by large extracted RLC networks. Also, substrate effects that start playing a crucial role in determining the performance are modeled by extracting, again, large resistive or RC networks. New algorithms are needed to cope with such situations that are extremely crucial for designers.

The complexity caused by these parasitic extractions must be reduced to facilitate the simulation of the circuit while preserving accuracy. Fortunately, highly accurate parasitic extraction is not necessary for all parts of the design. Each layout contains critical blocks or paths whose timing and performance is crucial for the overall functionality of the chip. High precision interconnect modeling must be used for these circuit parts to verify the functionality of the design. On the other hand, there is interconnect outside of critical paths which adds to the complexity but whose exact model is not necessary and can be simplified. For the critical paths a so-called sensitivity analysis can bring a major achievement in speed-up, by automatically determining the critical parasitic elements that provide the most dominant influence.

Another important aspect is the fact that there is an increasing deviation between design and manufacturing. Due to the ever decreasing feature sizes in modern chips, deviations from the intended dimensions are becoming more probable. Designers need to cope with this, and design the circuits in such a way that a deviation from intended dimensions does not alter the functionality of the circuit. In order to investigate this properly, one needs to assume that all components can possibly be slightly different after manufacturing. The effects this has on the performance of the circuit can be studied



by introducing many thousands or even millions of parameters, describing the deviations, and performing a sensitivity analysis of the circuit w.r.t. parameter changes.

The aforementioned problems form the inspiration for the study in this thesis. Sensitivity analysis is crucial for the correctness of virtual design environments based on electronic circuit simulators, and gives designers insight in how to alter the designs in order to guarantee more robustness with respect to variability in the design. The problem is that a thorough sensitivity analysis requires derivatives of the solution with respect to a large amount of parameters. This is not feasible using classical methods, being far too time-consuming for modern circuits. Recently proposed methods using the adjoint problem to calculate sensitivities are far more efficient, and these form the basis for our methodology. Our work has concentrated on making such methods even more efficient, by mixing them with concepts from the area of model order reduction. This leads to very efficient, robust and accurate methods for sensitivity analysis, even if the underlying circuit is large and the number of parameters is excessive.

# Samenvatting

De elektronische industrie levert de kerntechnologie voor veel industriële innovaties. Er zijn verschillende hoogtepunten aan te wijzen van de sterke vooruitgang op het gebied van de microelectronica, zoals bijvoorbeeld de ontwikkeling van microprocessoren en geheugenchips. De almaar voortdurende verbeteringen in rekenkracht en geheugendichtheid zouden onmogelijk zijn geweest zonder uitgebreid gebruik te maken van computersimulaties, en dan met name de simulatie van elektronische schakelingen. De basis voor dit laatste wordt gevormd door een krachtig bouwwerk van numerieke methoden.

In de afgelopen jaren zijn de eisen aan de mogelijkheden van software voor de simulatie van elektronische circuits enkel nog maar stringenter geworden. Circuitsimulatoren, zoals we ze kortweg noemen, vormen de kern van alle simulaties die tegenwoordig worden uitgevoerd in de elektronische industrie. Zo worden overspraakeffecten in interconnectstructuren gemodelleerd door grote RLC netwerken. Daarnaast worden substraateffecten ook steeds belangrijker, en daardoor is extractie van modellen voor dit gedrag ook essentieel. Vaak gebeurt dit door grote weerstandsnetwerken of RC modellen mee te simuleren. Nieuwe numerieke algoritmen zijn nodig teneinde met dit soort grote netwerken om te kunnen gaan. Dit alles is van groot belang voor de ontwerpers van de elektronische schakelingen.

De additionele complexiteit die wordt veroorzaakt door deze parasitaire extracties dient gereduceerd te worden teneinde de volledige simulatie van een schakeling te kunnen uitvoeren, maar met behoud van nauwkeurigheid. Gelukkig is het niet nodig om overal dezelfde mate van nauwkeurigheid te hebben. Elk ontwerp bevat een aantal kritische blokken of paden die cruciaal zijn voor het overall gedrag, en het is duidelijk dat deze nauwkeurig gesimuleerd dienen te worden. Maar de delen van de interconnect structuur die buiten de kritische paden vallen, kunnen drastisch vereenvoudigd meegenomen worden. Voor de kritische paden kan een zogenaamde gevoeligheidsanalyse gebruikt worden om de simulaties dramatisch te versnellen, juist door de meest dominante parasitaire elementen te identificeren.

Een ander belangrijk aspect is het feit dat er een toenemende discrepantie is tussen ontwerp en gereed product. Vanwege de alsmaar kleiner wordende details in moderne schakelingen, worden afwijkingen van de bedoelde afmetingen steeds waarschijnlijker.

Ontwerpers dienen hiermee om te leren gaan, en de schakelingen op een zodanige wijze te ontwerpen dat afwijkende afmetingen de functionaliteit van de schakeling niet beïnvloeden. Om dit te kunnen onderzoeken, dient men aan te nemen dat alle componenten enigszins af kunnen wijken na productie. Het effect dat dit heeft op het gedrag van de schakeling kan dan bestudeerd worden door vele duizenden of zelfs miljoenen parameters in te voeren welke de afwijkingen beschrijven. Men dient dus een gevoeligheidsanalyse uit te voeren van het gedrag van het circuit ten opzichte van de parameterveranderingen.

De hiervoor genoemde problemen vormen de inspiratie voor de studie en het onderzoek uitgevoerd en beschreven in dit proefschrift. Gevoeligheidsanalyses zijn cruciaal om de correctheid van een virtueel ontwerp te kunnen testen met circuitsimulators, en geven ontwerpers inzicht in hoe de ontwerpen aan te passen om meer robuustheid ten opzichte van parameterveranderingen te verkrijgen. Het probleem nu is dat een gedegen gevoeligheidsanalyse afgeleiden vereist van de oplossing naar een grote hoeveelheid parameters. Dit is onmogelijk met behulp van klassieke methoden te doen, omdat het veel te tijdrovend zou zijn. Recent voorgestelde methoden welke gebruikmaken van het geadjungeerde probleem om gevoeligheden te berekenen zijn echter veel efficiënter, en vormen derhalve de basis voor onze methodieken. Het werk heeft zich geconcentreerd op het nog efficiënter maken van deze klasse van methoden, door gebruik te maken van model orde reductie. Dit leidt tot zeer tijdsefficiënte, robuuste en nauwkeurige methoden voor het doen van gevoeligheidsanalyses, zelfs als de onderhavige elektronische schakeling erg groot is en het aantal parameters excessief.

# Acknowledgements

First and foremost I would like to thank my supervisor Wil Schilders for giving me the opportunity to work in the CASA group, as a member of COMSON and for my time at NXP, for his guidance and the unlimited support I have received throughout my PhD. I would also like to thank him for giving me the possibility to travel, to meet so many people in the field of my study, for the experiences and enjoyment, and the friends I have made across Europe.

I am heartily thankful to Jos Maubach, who as my copromoter, has provided me with endless encouragement, advice and the benefit of his knowledge.

I would also like to give my sincere gratitude to the other members of my promotion committee, Tom Dhaene, Michael Guenther, Bob Mattheij, Jacquélien Scherpen and Michael Striebel.

I have been fortunate to have spent many days working at NXP Semiconductors (founded by Philips) at the Eindhoven High Tech Campus, I would like to give special thanks to Jan ter Maten for investing his time and advice, and also for the many discussions we have had at NXP. Over the years at NXP, I had the pleasure to share an office with many people who made it a pleasant, enjoyable and welcoming working environment, thank you to Theo Beelen, Davit Harutyunyan, Joost Rommes, Maryam Saasvandi, Bratislav Tasic, Luciano De Tommasi and also to Aad Duinmaijer.

I thank all of my current and former CASA colleagues, for the many social events we have shared together, our regular poker matches, lunches, sunny evening nights out, meals and music, and the many CASA events, all of which contributed to a great feeling of being at home; thank you Laura Astola, Evgeniya Balmashnova, Sinatra Canggih, Miguel Patricio Dias, Remco Duits, Ali Etaati, Tasnim Fatima, Yves van Gennip, Christina Giannopapa, Shruti Gumaste, Andriy Hlod, Bart Janssen, Jan Willem Knopper, Mark van Kraaij, Jan Kroot, Kundan Kumar, Sanda Lefteriu, Bas van der Linden, Kamyar Malakpoor, Temesgen Markos, Oleg Matveichuk, Jos and Peter in't Panhuis, Maxim Pisarenco, Patricio Rosen, Maria Rudnaya, Valeriu Savcenco, Neda Sepasian, Berkan Şeşen, Volha Shchetnikava, Antonino Simone, Sudhir Srivastava, Jurgen Tas, Arie Verhoeven, Venkat Viswanathan, Erwin Vondenhoff, Fan Yabin and many more. Special thanks to the people with whom I have shared a large office in the past: Mirela

Darau, Hans Groot, Qingzhi (Darcy) Hou, Roxana Ionutiu, Godwin Kakuba, Evelyne Knapp, Agnieszka Lutowska, Remo Minero, Evgueni Shcherbakov and Maria Ugryumova.

I am appreciative of the help, assistance and support given to me by Enna van Dijk, Marèse Wolfs-van de Hurk and Irene Andringa Portela for the many times I have asked for help on a wide variety of issues, thank you for looking after me.

I would also like to mention my COMSON colleagues with whom I have shared many meetings and events accross Europe, thank you for making COMSON the experience it was for me, thank you Giuseppe Ali, Gabriela Ciuprina, Massimiliano Culpo, Georg Denk, Carlo de Falco, Franco Fiorante, Daniel Ioan, Sebastian Kula, Kasra Mohaghegh, Roland Pulch, Alexander Rusakov, José R. Sepúlveda Sanchis, Carmelo Scordia, Alexander Vasenev.

On a more personal note, I would like to thank my family who to me are the most precious people on this earth, especially my mother, Ivana Ilievska, who has fought so strongly and bravely to be here with us today, I love you Mum. Thank you to my father, Slavé Ilievski and my sister, Angelina Ilievska for the support they have given me, especially during recent times, when we needed it most.

Lastly, but certainly by no means least, I would like to thank my special one, Regina Lally, for being an unlimited source of love, support and patience when I needed it most.

Zoran Ilievski

Eindhoven, July 2010.

# Curriculum vitae

Zoran Ilievski was born on 30th May 1979 in Carshalton, United Kingdom. Between 1995 and 1997 Zoran studied for his A-levels in Physics, Mathematics and Further Mathematics at Rutlish Sixth Form, Merton, London, United Kingdom. In 2001, he finished his Masters of Engineering (MEng) at King's College Department of Electronic and Electrical Engineering, University of London in London, United Kingdom. Zoran then went on to study for his Masters of Science (MSc) in Advanced Computing at King's College Department of Computer Science, University of London, graduating in 2002. Between 2002 and 2005 Zoran worked as a computer applications analyst for CapGemini and UK government departments. In 2005, Zoran began a PhD project at the Eindhoven University of Technology, Eindhoven, The Netherlands, the results of which are presented in this dissertation. Since the beginning of 2010, Zoran has been working at the European Space Research and Technology Centre, Noordwijk, The Netherlands, and is an employee of the European Space Agency.