Evaluation of Uncertainty in Dynamic, Reduced-Order Power System Models

by

James R. Hockenberry

B.S., Swarthmore College (1995)
S.M., Massachusetts Institute of Technology (1997)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2000

Author ______________________________
Department of Electrical Engineering and Computer Science
July 12, 2000

Certified by ______________________________
Bernard C. Lesieutre
Associate Professor of Electrical Engineering
Thesis Supervisor

Accepted by ______________________________
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Evaluation of Uncertainty in Dynamic,
Reduced-Order Power System Models

by
James R. Hockenberry

Submitted to the Department of Electrical Engineering and Computer Science
on July 12, 2000, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

With the advent of high-speed computation and the desire to analyze increasingly complex behavior in power systems, simulation techniques are gaining importance and prevalence. However, while simulations of large, interconnected power systems are feasible, they remain time-consuming. Additionally, the models and parameters used in simulations are uncertain, due to measurement uncertainty, the need to approximate complex behavior with low-order models and the inherent changing nature of the power system. This thesis explores the use of model reduction techniques to enable the study of uncertainty in large-scale power system models.

The main goal of this thesis is to demonstrate that uncertainty analyses of transient simulations of large, interconnected power systems are possible. To achieve this, we demonstrate that a basic three stage approach to the problem yields useful results without significantly increasing the computational burden. The first stage is to reduce the order of the original power system model, which reduces simulation times and allows the system to be simulated multiple times in a reasonable time-frame. Second, the mechanics of the model reduction are closely studied; how uncertainties affect the reduction process and the parameters in the reduced-order model as well as how the process of reduction increases uncertainty are of particular interest. Third, the reduced-order model and its accompanying uncertainty description are used to study the uncertainty of the original model.

Our demonstration uses a particular model reduction technique, synchronic modal equivalencing (SME), and a particular uncertainty analysis method, the probabilistic collocation method (PCM). Though our ideas are applicable more generally, a concrete demonstration of the principle is instructive and necessary. Further, while these particular techniques are not relevant to every system, they do apply to a broad class of systems and illustrate the salient features of our methodology. As mentioned above, a detailed analysis of the model reduction technique, in this case SME, is necessary. As an ancillary benefit of the thesis work, interesting theoretical results relevant to the SME algorithm, which is still under development, are derived.

Thesis Supervisor: Bernard C. Lesieutre
Title: Associate Professor of Electrical Engineering

Dedication

Za Anamariju, moju jednu i jedinu ljubav

(For Anamarija, my one and only love)

Acknowledgements

While only one name appears on the author line of a thesis, every thesis is the product of a collaborative effort, and this thesis is certainly no exception. This section is designed to acknowledge individuals who played important roles in the development of the thesis work or in the development of the author (in most cases both). If I forgot anyone, please do not take offense. It doesn't mean that I don't appreciate you; it just means that I'm an idiot and, in the final rush to finish this thesis, my brain froze. I promise to repay both the forgotten and the remembered with cool (but not cold) beers, or other appropriate barter.

I'd like to thank (a lot) Bernie Lesieutre for being my research advisor here at MIT during the past five years. While it is true that Bernie knows his stuff and has lead me in interesting research directions, that's not primarily why he gets the marquee slot in the acknowledgements sections. While it is also true that Bernie has always encouraged me without standing and looking over my shoulder (although maybe that wouldn't have been such a bad idea sometimes), I don't think that quite captures it either. And though it's not an elegant or profound statement, I'd mostly like to thank Bernie just for being a nice guy and all-around good role model, which is why I picked him to be my advisor and I've never regretted that decision.

I'd like to thank the members of my committee, George Verghese, Steve Leeb and Pete Sauer, for reading and critiquing this document. I'd particularly like to thank George for various suggestions and ideas over the years. George is the driving force behind the current research on SME, and he's been a good resource as I've worked on issues related to SME.

Much of the work in this thesis was influenced by weekly meetings and other discussions with those working on SME development under the auspices of EDF. In particular, George Verghese, Bernie Lesieutre, Marek Teste, Anuradha Saxena and Ernst Scholtz have all been very helpful to me and my work. My sympathies go out to Ernst, who will be left (at least for the time being) as the sole graduate student working actively in this area. Hey Ernst, I'll at least buy you a beer to help lighten the weight of the world on your shoulders.

I've spent my days (and, for better or for worse, occasional nights) here at MIT in the Laboratory for Electromagnetic and Electronic Systems, better known as LEES. While the setting is not exactly idyllic, the quality of the people here more than compensate for the lack of faux marble. I have interacted both professionally and less than professionally with many folks over the years. Many thanks go out to all and particularly Vivian Mizuno, Jama "Total Package" Mohamed, Vahe "Dick Callahan" Caliskan, Babak Ayazifar, Jamie Byrum, Tim Neugebauer, Ernst Scholtz, Tim Denison, John Rodriguez, Ike Trefz, Jesko Hagee, Kathy Millis and Joshua Phinney. I give a special mention to Jama, who was always willing to share his computer (i.e., LaTeX) knowledge and was gracious enough to lend me his computer. Special thanks go out to Vahe; he's always willing to lend a hand and is one of the most considerate people I know, as long as you consider telling bad jokes and making wrestling references "considerate".

The bulk of the work in this thesis (if not all of it) was completed while I was supported with funds from a generous gift from the Grainger Foundation. I am very appreciative of their support, which has allowed me to work on my thesis without having to worry about my funding situation. It has truly been my pleasure to meet and discuss my work with both David Grainger and Admiral Bill Hayden.

The ideas presented in this thesis were tested on real system data graciously supplied to us by Carson Taylor of BPA; we are, needless to say, grateful.

My academic advisor, Professor Jeff Shapiro, deserves particular attention for helping keep me on track during the last five years. Since the PhD process is not a short one, it can be easy to procrastinate and prolong the process. Jeff always helped me stay on track and tried to find a reasonable academic goal for me to complete each term.

During my time in Boston, I've been fortunate to have a few friends here who I knew before I came here and a few others that I've made along the way. To one and all, my heartfelt thanks. In particular, I'd like to mention Erik Thoen, who has managed to survive the rigors of Swarthmore and MIT with most of his sanity intact. He recently agreed to be my "Godfather" for baptism, although his name doesn't sound like something out of a Mario Puzo novel. I'd also like to send out my deep appreciation to Joe Teja, who has always been there when I needed him, ever since we "hit it off" while TAing 6.002 way back in Fall 1995. Joe was smart enough to get out while the getting was good. I decided to take the scenic route. Last (for this section; as Bernie says, "You can't say your *name* in less than two minutes"), though I've only known most of them for an (unfortunately) relatively brief amount of time, I'd like to thank all of the members of the Tech Catholic Community, who were supportive of me when I sought full membership in the church. In particular, I give heartfelt thanks to Sister Kathleen Crowley and Brother Hugh Gillespie, who're the best catechists in the world, period.

Last, but not least, I'd like to thank my family, both old and new. My parents have always supported me and haven't "second-guessed" me very often, even though I haven't always done the obvious (or even right) thing. At least, it seems to have all worked out pretty well so far, and I owe them a lot for helping make it all possible. My sister has been through the wars and doesn't seem to hold any of it against me, which is more than anybody has any right to ask for. And very last but very most important, I'd like to thank Anamarija Baričević, who has had to survive a lot of uncertainty (no pun intended) and several months of separation (I had to survive the separation too, but it was mostly my fault, so...) as I tried to finish this thesis. I couldn't ask for a more patient (or beautiful) wife.

So, Hvala Lijepa and Thanks!

Contents

1 **Introduction** **17**

1.1 Scope and Contributions of the Thesis 19

1.2 Organization of the Thesis 22

2 **The Probabilistic Collocation Method** **23**

2.1 Underlying Theory 24

2.1.1 Orthogonal polynomials 24

2.1.2 Gaussian Quadrature Integration 26

2.2 Monte Carlo Methods and Single Parameter PCM 28

2.2.1 Brute Force Monte Carlo 28

2.2.2 Single Parameter PCM 29

2.2.3 Variance Reduction Techniques - Importance Sampling 31

2.2.4 Simple Circuit Example 33

2.2.5 Single Machine, Infinite Bus Example 43

2.2.6 Multiple Machine Power System Example 47

2.3 Multiple Uncertain Parameters 51

2.3.1 Dependent Uncertain Parameters - General Case 52

2.3.2 Independent Uncertain Parameters 53

2.3.3 Necessary Computation 55

2.4 Summary 56

3 **Synchronic Modal Equivalencing - a primer** **57**

3.1 Basic Concepts and Components 57

3.2 Linearization and Extraction of Simplified Model 59

3.2.1 Ideal Simplified Model for Generator Partitioning 60

3.2.2 Simplified Model Extraction and Load Partitioning 63

3.3 Chord and Basis Generator Selection 68

3.3.1 Chord Selection 68

3.3.2 Basis Generator Selection 71

3.4 Partitioning the System 73

3.5 Study Area Selection and System Reduction 75

3.5.1 Generator Equivalencing 77

3.5.2 Network Reduction 79

3.6 Summary 83

4 Uncertain Parameters - Modeling and Prioritizing 85

4.1 Analysis of the External Reduction 86

4.1.1 Topological Approach 87

4.1.2 Direct Computation of Sensitivities 93

4.1.3 New Element Uncertainties 96

4.2 Identification of Key Parameter Uncertainties 97

4.2.1 Our Method - Introduction 98

4.2.2 Our Method - Practical Issues 99

4.3 Summary 100

5 Examples - Both Great and Small 103

5.1 “Small” Example 104

5.1.1 Synchronic Modal Equivalencing 105

5.1.2 Identification of “key” uncertain parameters 106

5.1.3 The Probabilistic Collocation Method 107

5.1.4 Simulations, Results and Comparisons 108

5.2 “Large” Example 128

5.2.1 Synchronic Modal Equivalencing 129

5.2.2 Key Uncertainties 133

5.2.3 PCM, Simulations and Results 134

5.3 Summary 142

6 Conclusions and Future Work **145**

6.1 Summary 145

6.2 Suggestions for Future Work 147

6.2.1 Further Analysis of SME 147

6.2.2 Extension and Refinement of PCM 147

6.2.3 Modeling of Uncertain Parameters 148

6.2.4 Further Refinement and Automation of Implementation 148

6.2.5 Further Testing 148

6.2.6 Simulation-specific "Key" Parameters 149

6.2.7 Studying Uncertainty 149

A Orthogonal Polynomial Generation **151**

B SME - Error and Robustness **153**

B.1 Error Analysis 153

B.1.1 Full-Mode Correction 154

B.1.1.1 Change of Variables Approach 155

B.1.1.2 Direct Substitution Approach 158

B.1.1.3 Special Case: Reduction to a Scalar System 161

B.1.1.4 Summary 162

B.1.2 Zero-Mode Correction 162

B.1.2.1 Full $\mathbf{K}$ 163

B.1.2.2 Linear, Swing-Equation Example 168

B.1.2.3 Basis Generator Correction $\mathbf{K}_\mathbf{b}$ 170

B.1.2.4 Summary 172

B.1.3 Error Analysis Conclusions 173

B.2 "Robustness" of the SME partition 174

B.2.1 Eigenvector Sensitivity 175

B.2.2 Synchrony 176
B.2.3 Non-Asymptotic Results 179
B.2.4 "Empirical" Studies and Conclusions 181
B.3 Summary 184

C Key Uncertain Parameter Identification Program **187**

List of Figures

2.1 Simple RC-Circuit Example 33

2.2 PDF for Resistance R 34

2.3 Comparison of Linear Approximations 37

2.4 PDFs from Linear Approximations 39

2.5 Comparison of Quadratic and Linear PCM 41

2.6 Quadratic and Linear Model PDFs 42

2.7 Single Machine, Infinite Bus 43

2.8 Linear Approximations for SMIB Example 46

2.9 Multiple Machine System 47

2.10 PCM Time-Varying Example 50

3.1 Schematic of Reduction 76

4.1 External Network Before Reduction 88

4.2 External Network After Reduction 88

5.1 “Small” Power System Example 104

5.2 “Small” Power System Example - SME Reduction 105

5.3 Generator A Angle - Benchmark Uncertainty Analysis 109

5.4 Voltage at Bus Lunt - Benchmark Uncertainty Analysis 110

5.5 Generator A Speed - Benchmark Uncertainty Analysis 111

5.6 Generator A Angle - “Key” Uncertainties 113

5.7 Voltage at Bus Lunt - “Key” Uncertainties 114

5.8 Generator A Speed - “Key” Uncertainties 115

5.9 Generator A Angle - Probabilistic Collocation Method 116

5.10 Voltage at Bus Lunt - Probabilistic Collocation Method 117

5.11 Generator A Speed - Probabilistic Collocation Method 118

5.12 Generator A Angle - PCM & SME 119
5.13 Voltage at Bus Lunt - PCM & SME 120
5.14 Generator A Speed - PCM & SME 121
5.15 Generator A Angle - PCM & SME (Benchmark) 122
5.16 Voltage at Bus Lunt - PCM & SME (Benchmark) 123
5.17 Generator A Speed - PCM & SME (Benchmark) 124
5.18 Nominal Generator A Angle - Full Model and SME Model 125
5.19 Nominal Voltage at Bus Lunt - Full Model and SME Model 126
5.20 Nominal Generator A Speed - Full Model and SME Model 127
5.21 Nominal Generator 26233_1 Angle - Full Model and SME Model 130
5.22 Nominal Voltage at Bus 26234NQI - Full Model and SME Model 131
5.23 Nominal Generator 26233_1 Speed - Full Model and SME Model 132
5.24 Generator 26233_1 Angle - Uncertainty 136
5.25 Voltage at Bus 26234NQI - Uncertainty 137
5.26 Generator 26233_1 Speed - Uncertainty 138
5.27 Generator 26233_1 Angle - Comparison 139
5.28 Voltage at Bus 26234NQI - Comparison 140
5.29 Generator 26233_1 Speed - Comparison 141

B.1 Steady-State Matching for Damped Swing-Equation Model 170
B.2 France-Spain and Partitioning 182

List of Tables

2.1 Comparison of Monte Carlo and PCM 48

2.2 Comparison of Polynomial Orders 51

4.1 Tier Example 90

5.1 Network Reduction Sensitivities for Murphy - Deering 107

5.2 Third-Order Polynomial Roots 108

5.3 Large Network SME Reduction 129

5.4 Time-Savings 142

B.1 Partitioning Changes Induced by Load Parameter Variations 183

B.2 Degrees of Synchrony of Affected Nodes 183

Chapter 1

Introduction

Large, nonlinear, dynamic systems, such as power systems, are challenging to model and analyze. Numerous techniques have been developed to alleviate the difficulties stemming from the size and complexity of a typical realistic power system model; however, this research area is still very active, and many problems remain open.

One other important adjective which can be applied to a typical power system model is uncertain. As with any physical system, a model for a power system is only a model. The form and the parameters of the model are designed to approximate the physical behavior of the real system, but they will always stubbornly remain approximations. Additionally, a power system model is uncertain as a result of more particular issues not shared by all large dynamic systems. Even if we could claim validity for the form of a particular power system model, our knowledge of the parameters is still plagued by uncertainties stemming from difficulties in reliable data acquisition and the changing nature of the real power system. As an example, let us consider the situation of those power system elements commonly referred to as loads.

Loads are those portions of the power system which consume electricity. When the power system is viewed on the large scale typically used for transmission studies, many individual consumers of electricity are aggregated to create one large load. For example, when a toaster is turned on, the load, as modeled by a power system engineer, is affected. This action illustrates the modeling and identification problems presented by loads. First, the actual load model used for this aggregate load comprises many individual (possibly disparate) power consuming devices. Any reasonably simple aggregate model will only be a very rough approximation to the actual composite behavior of the individual devices. Second, the aggregate model is uncertain and changes with time. The exact composition of the aggregate at any given time is not known and changes as soon as a toaster is turned on. (A few papers from the vast literature related to load modeling are [1], [2], [3], [4] and [5]; the first three papers are surveys/bibliographies and may be particularly helpful.)

One approach to overcoming the problems outlined above is to improve the models, perhaps using models that more closely approximate the physical behavior of the devices being modeled. (See [6] for one example.) However, physically-based load modeling requires extensive load monitoring capabilities, which are not yet available. (See [7] for one presently feasible load monitoring scheme.) Another approach is the one taken *de facto* presently, which is essentially to ignore the problem and declare present load modeling techniques "good enough". But, the recent WSCC outage in the western United States graphically illustrates the problems with this approach [8]. Although the system was subjected to a drastic and detrimental set of events precipitated by heavy loading conditions and the loss of a transmission line, the initial simulations of these conditions showed that the system was "secure", i.e., the real set of events were not predicted by the simulations. One key conclusion of research on this sequence of events is that some of the original load models needed to be modified to produce the actual system behavior. (A series of events have occurred on various systems over the past couple of decades, which have pointed to the need for better load modeling. For another example, see [9].)

We would like to advocate yet another approach to the load modeling accuracy problem; since we cannot overcome the uncertainty in our power system models, we should attempt to model these uncertainties using appropriate methods.

Of course, the already difficult problem of accurate power system modeling and analysis is only further complicated by the extra effort required to accurately model the uncertainty. However, using sophisticated model reduction and uncertainty analysis techniques, the problem is not intractable, and if the security of the system is increased, the effort is well spent. Therefore, the goal of this thesis is to motivate the study of uncertainty in large, nonlinear, dynamic systems, particularly power systems. The aim is to enable power system engineers to incorporate the study of uncertainty into their system studies.

1.1 Scope and Contributions of the Thesis

The overall goal of this thesis is to motivate and enable the incorporation of uncertainty into the analysis of large power system networks. We believe that this is a broad and ambitious goal, and we do not presume to "solve" all of the problems inherent in studying uncertainty in very large models. However, we have carved out a sizable project within this area and demonstrate a framework and set of techniques which are applicable to a broad range of problems of interest.

Specifically, we only consider simulation studies and parameter uncertainties within those studies. While analytical techniques are useful and appropriate for many problems, simulation studies are ubiquitous in the power system field, due to their wide applicability and the ability to apply them to problems where an analytical treatment is intractable. Therefore, only studying simulations is a fairly minor restriction and the results are broadly applicable to a variety of power system problems of interest. Only treating parameter uncertainties is a more limiting restriction. Specifically, we assume that the components of the power system model are "correctly" modeled, but that the parameters used to describe a particular component within the context of that model are subject to uncertainty. Restricting the scope in this way is necessary to apply the techniques used, and the results are, nevertheless, useful. For example, in the case of the WSCC event described above, many of the changes to the system model necessary to recreate the actual course of events were parameter changes, as opposed to changes to the actual models used.

Within the context of the restrictions outlined above, the framework presented here is more general than the individual tools used to realize our goal practically. Our general approach consists of three stages. First, the power system model is reduced using a model reduction technique. Model reduction of some variety is necessary to obtain a model which is more amenable to analysis and simulation. If the reduced model can be simulated ten times as fast as the original model, obviously the reduced model can be simulated ten times in the amount of time currently used to simulate the full model, which opens up the possibility for uncertainty studies. Next, the relationship and interactions between the original uncertainty and the model reduction are studied to obtain an uncertainty description of the reduced model. This step is key and is the main focus of the work presented in this thesis. Last, an uncertainty analysis of the reduced model is performed. Under the heading of uncertainty analysis, we include Monte Carlo simulations and related techniques.

While we are advocating a general way to approach uncertainty analyses of power system models, our work centers on a specific realization of this general schema. We have attempted to select the best possible available methods of realizing each of the stages in our approach. It is important to realize that substitutions of other specific procedures for model reduction or the final stage of uncertainty analysis are possible and encouraged, especially if new and superior methods eventually present themselves. Of course, the crucial second step of studying how the uncertainty from the original model relates to the final model, depends strongly on both the model reduction technique and the ultimate uncertainty analysis procedure. The first dependency is obvious, and the second is clear because different uncertainty analysis techniques require different descriptions of the uncertainty (e.g., bounds on the parameters may be necessary as opposed to probability descriptions).

Synchronic modal equivalencing (SME) was chosen as the model reduction technique. (See [10], [11], [12], [13] and [14].) SME is well-suited to our particular needs, but we postpone explaining its relative strengths and weaknesses until a later chapter, where we also carefully explain the details of the method. We do note here that the analysis is largely based on a linearization of the original, nonlinear system and SME delivers a reduced-order model with physically meaningful variables and parameters. SME is still a relatively new technique and is currently the focus of ongoing research; as a result, some of the work in this thesis has been, of necessity, involved in developing and describing the mechanics of its reduction so that its effects on the uncertainty and the uncertainty's effect on it could be studied.

The probabilistic collocation method (PCM), which is a polynomial modeling technique for Monte Carlo simulations, was chosen to do the uncertainty analysis. (See [15], [16] and [17].) Again, we postpone a detailed description of the method until later in this thesis. However, the basic principle of the technique is to model the relationship between uncertain parameters and system variables of interest using a polynomial. If the assumptions underlying PCM are satisfied, remarkably accurate results are obtained with startlingly few simulations of the actual model. We have adopted the standpoint that we are primarily interested in studying power systems using simulation techniques because we wish to study the transient behavior of many different system variables; this meshes well with our use of PCM, since it is a general technique for simulation studies, in the sense that the same set of simulations can be used to study all system "output" variables. One other important point to make is that this method requires uncertain parameters to be described using probability density functions, which guides the analysis of the model reduction method.

While the contribution of this thesis is primarily the definition and development of the framework for studying uncertainty described above which enables uncertainties in power system simulations to be studied, this thesis, as with most theses, also contributes in other peripheral ways. We have attempted to list below what we see as the contributions:

- A good understanding of a model reduction technique is essential for studying the effects of uncertainty on it. We have chosen to use SME as our model reduction technique, and we have developed a deeper understanding of it in the course of our investigations. Specifically, the error introduced by the reduction (specifically a possible steady-state error) was studied and an expression for this error was derived for an ideal case; the structure of the matrices used to select the "chord" in SME was investigated and explained; and, the robustness of ideal synchrony to perturbations of the system matrices was studied. In all cases, these developments are necessary for justifying the use of SME as an accurate model reduction technique; however, the deeper understanding gained is also more widely applicable to current research on the SME algorithm.

- As we have already stressed, a general methodology for studying uncertainties in power systems is outlined in this thesis. As shown by the previously cited bibliography and survey papers, the desire to incorporate uncertainty and probability into power system studies is not new. Indeed, the arguments in favor of incorporating probabilistic techniques into power system studies are strong. (See [18] for some cogent arguments.) However, much of the work previously done focuses on reliability indices, which are scalar quantities and do not necessarily require extensive time-based simulations to evaluate. (For an introduction to the vast literature related to probability in the context of power systems, see the following bibliography papers: [19], [20], [21], [22], [23], [24] and [25].) Another interesting approach has been to use trajectory sensitivity analysis to study the effects of parameter changes. (See [26], [27] and [28].) However, this approach only allows one to determine the small signal sensitivity of a given simulation to a particular parameter; the necessary extensions to allow one to obtain statistical information, such as expected values and standard deviations, based on this technique have not yet been developed. In addition, the small signal analysis may not be valid for parameters with large fluctuations.

- To demonstrate our methodology, we have implemented our three-stage approach using SME and PCM. The process is largely automated using a combination of MATLAB scripts [29] and a Maple notebook [30]. This code can be used to study any general power system. However, the results are only valid for models that are well-suited to SME reduction and for outputs that can be well-modeled using PCM.

This thesis may be of interest to anyone studying reduced-order modeling or uncertainty analyses of power systems. Of course, the techniques are applicable to a broader class of problems, but we are primarily interested in power systems and we use almost exclusively power system examples. However, the level of specific knowledge about power systems assumed is not high. Most of the necessary details from power systems are explained within the thesis, as are the basics of synchronic modal equivalencing and the probabilistic collocation method. The assumed prerequisites are basic linear algebra, state-space modeling techniques and basic probability.

1.2 Organization of the Thesis

The remainder of this thesis comprises five chapters and three appendices. In Chapter 2 we present a primer on the probabilistic collocation method. In a similar manner, Chapter 3 discusses synchronic modal equivalencing. We attempt to present enough of the theory to allow the reader to follow our work without becoming overwhelmed. Chapter 4 summarizes the way in which we have combined the probabilistic collocation method and synchronic modal equivalencing, which includes determining critical uncertainties in the reduced model and an analysis of the sensitivity of the topology-reduction portion of the SME algorithm. The results of tests of our method on both a small and a large power system model are presented in Chapter 5. The method is designed to work well on large models; however, it is infeasible to verify our results on such a model, due to the time and computational power required using traditional methods. Therefore, we present results on a large model to demonstrate the enabling aspect of our work and results on a small model, which we can compare to results obtained using traditional methods. Chapter 6 concludes and summarizes the thesis as well as presenting possible future research directions. In addition, Appendix B may be of interest to readers familiar with SME; it presents an analysis of the errors introduced by SME and a discussion of the robustness of the groupings created by SME.

Chapter 2

The Probabilistic Collocation Method

Although model reduction is necessary to create a reduced-order model which can be simulated reasonably quickly, model reduction alone is not sufficient to enable uncertainty analyses of power systems. Any uncertainty analysis performed on the reduced-order model also needs to be quick, in the sense that it should not require too many simulations. We desire something akin to a Monte Carlo method (see [31] for an introduction to Monte Carlo methods in the context of power systems), but a simplistic Monte Carlo analysis of a large power system model with many uncertainties would require literally thousands of simulations, if not more. Present model reduction techniques simply cannot speed up simulations of large power systems sufficiently to make thousands of simulations practical. A more sophisticated Monte Carlo analysis using standard variance reduction techniques (e.g., importance sampling) can significantly reduce the number of required simulations. However, in most cases, too many simulations are still necessary, and, more importantly, a new set of simulations would generally be required for each output of interest, even for the same event. For example, to examine a time-varying response such as a voltage, one can define each of the voltage values at a number of sample times to be an output; each of these sample times would require a new set of simulations.

Our solution is a relatively new technique, based on polynomial modeling of the relationship between uncertain parameters and the output(s) of interest. The probabilistic collocation method (PCM) generates fairly accurate results using one small set of simulations for all possible outputs of interest. PCM was originally introduced in the context of global climate change studies. The first mention of it occurs in [15], and the ideas are further explored in [16]. The method has also been applied to a power system example in [17]. In this chapter, we present the relevant background on PCM, since our uncertainty analyses would not be feasible without it or a similar technique.

We start by discussing orthogonal polynomials and Gaussian quadrature integration. (Many numerical integration texts treat these subjects. For example, see [32].) The probabilistic collocation method is largely heuristic, but the theory of orthogonal polynomials and Gaussian quadrature integration provides strong motivation for the method. We then explore the method for a single parameter uncertainty using a number of different basic examples. Multiple uncertain parameters are also discussed and we conclude with a brief discussion of the limitations of the method.

2.1 Underlying Theory

Two major ideas are necessary in order to understand how the probabilistic collocation method works. First, orthogonal polynomials are not only necessary background for understanding Gaussian quadrature, but they also figure prominently in the actual computations involved in the probabilistic collocation method. Second, Gaussian quadrature integration is a numerical integration technique based on the theory of orthogonal polynomials. Gaussian quadrature integration supplies a justification for the largely heuristic probabilistic collocation method as well as proving that PCM is optimal for a particular class of problems under certain assumptions.

2.1.1 Orthogonal polynomials

As the name implies, orthogonal polynomials are polynomials which are orthogonal with respect to a legitimate inner product. The general conditions under which a function is a legitimate inner product for a vector space can be found in any algebra text and most analysis texts; see [33] for example. We denote inner products using an angled bracket notation. Over the space of all polynomials, a scalar function of a pair of polynomials is an inner product if the following conditions are met for all polynomials g, h and k and all scalars c:

$$\begin{aligned} \langle g, h \rangle &= \langle h, g \rangle & (2.1) \\ \langle g + h, k \rangle &= \langle g, k \rangle + \langle h, k \rangle & (2.2) \\ \langle cg, h \rangle &= c\langle g, h \rangle = \langle g, ch \rangle & (2.3) \\ \langle g, g \rangle &> 0 \text{ if } g \neq 0 & (2.4) \end{aligned}$$

Let the polynomials be functions of x. Then, the following function is an inner product on the space of polynomials:

$$\langle g(x), h(x) \rangle = \int_{\mathcal{A}} f(x)g(x)h(x)dx \qquad (2.5)$$

where $f(x)$ is *any* non-negative weighting function defined everywhere in a connected $\mathcal{A}$. This particular inner product is the one used for Gaussian quadrature integration and the probabilistic collocation method.

Given this inner product, a pair of polynomials is said to be orthogonal if their inner product is zero. Further, a set of polynomials $\mathcal{H}$ is said to be orthonormal if and only if the following relationship holds for all $h_i(x)$ in $\mathcal{H}$:

$$\langle h_i, h_j \rangle = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \qquad (2.6)$$

Given a weighting function $f(x)$, we are interested in a particular such orthonormal set $\mathcal{H}$, where the h_i are in increasing order. In other words, h_i is a polynomial of order i. These polynomials are unique and form a basis for the space of all polynomials. Efficient methods exist for obtaining these polynomials; one such method is outlined in the appendices.

Such a set of orthonormal polynomials under the inner product (2.5) has a number of nice properties. Of particular interest are the roots of these polynomials. We omit the proofs, but it can be shown that each h_i has exactly i roots and all of the roots are contained in $\mathcal{A}$. These roots play a pivotal role in the probabilistic collocation method.

2.1.2 Gaussian Quadrature Integration

Gaussian quadrature integration is a numerical integration technique for integrals of the form:

$$\int_{\mathcal{A}} f(x)g(x)dx \tag{2.7}$$

where $g(x)$ is a polynomial and $f(x)$ is a non-negative weighting function. As the reader may note, this integral is strikingly similar to our earlier inner product definition. The main result of Gaussian quadrature integration is the following exact formula for calculating this integral:

$$\int_{\mathcal{A}} f(x)g(x)dx = \sum_{i=1}^{n} f_i g(x_i) \tag{2.8}$$

where **the f_i are constants which only depend on the weighting function $f(x)$** and the x_i are constants in the region of integration. The formula is **exact for all polynomials $g(x)$ of order less than or equal to $2n - 1$**. This result is somewhat surprising. The polynomial $g(x)$ itself could be determined using $2n$ samples, but significantly fewer samples of $g(x)$ are needed to compute the integral.

To compute the x_i, we use our earlier orthonormal polynomial set $\mathcal{H}$, where $f(x)$ is used as the weighting function. The constants x_i are the roots of h_n, which exist and are contained in $\mathcal{A}$. Furthermore, the x_i only depend on $f(x)$, since $\mathcal{H}$ only depends on $f(x)$. We show the independence of the f_i and x_i from $g(x)$ constructively.

The polynomials in $\mathcal{H}$ up to and including order i constitute an orthonormal basis for the space of all polynomials of degree less than or equal to i. Therefore, a polynomial of order $2n - 1$ can be expressed in terms of these orthonormal polynomials using constant coefficients a_i and b_i:

$$g(x) = h_n(x)\left(a_{n-1}h_{n-1}(x) + \cdots + a_0 h_0(x)\right) + b_{n-1}h_{n-1}(x) + \cdots + b_0 h_0(x) \tag{2.9}$$

We note that, if we expand this expression by multiplying through by $h_n(x)$, the result is a sum of 2n linearly independent polynomials, which proves that such an expansion is always feasible. Since $h_0(x)$ is a constant, the integral is easily determined by orthogonality:

$$\int_{\mathcal{A}} f(x)g(x)dx = b_0 \int_{\mathcal{A}} f(x)h_0(x)dx \tag{2.10}$$

Finally, we build a linear set of equations by evaluating $g(x)$ in (2.9) at the roots of $h_n(x)$:

$$\begin{bmatrix} g(x_1) \\ \vdots \\ g(x_n) \end{bmatrix} = \begin{bmatrix} h_{n-1}(x_1) & \cdots & h_0(x_1) \\ \vdots & \ddots & \vdots \\ h_{n-1}(x_n) & \cdots & h_0(x_n) \end{bmatrix} \begin{bmatrix} b_{n-1} \\ \vdots \\ b_0 \end{bmatrix} \tag{2.11}$$

By inverting this matrix, we derive an expression for b_0:

$$\begin{bmatrix} b_{n-1} \\ \vdots \\ b_0 \end{bmatrix} = \begin{bmatrix} h_{n-1}(x_1) & \cdots & h_0(x_1) \\ \vdots & \ddots & \vdots \\ h_{n-1}(x_n) & \cdots & h_0(x_n) \end{bmatrix}^{-1} \begin{bmatrix} g(x_1) \\ \vdots \\ g(x_n) \end{bmatrix} \tag{2.12}$$

Since $\mathcal{H}$ is determined solely by $f(x)$, both x_i and f_i are independent of $g(x)$. If we define $h_0(x)$ to be 1 (which is customary), then b_0 is the desired result:

$$b_0 = \int_{\mathcal{A}} f(x)g(x)dx = \sum_{i=1}^{n} f_i g(x_i) \tag{2.13}$$

The weights f_i are given by the last row of the matrix in (2.12).

This last point is important as we segue from the underlying theory to Monte Carlo methods and the probabilistic collocation method itself. The independence of the x_i from the particular polynomial for which we are calculating the integral is analogous to the desired independence between the set of parameter values for which we perform simulations and the particular output variable of interest. One set of x_i suffices for all polynomials of order less than or equal to $(2n-1)$ and one set of simulations is sufficient for all outputs of interest.

2.2 Monte Carlo Methods and Single Parameter PCM

As we switch gears from our theoretical discussion to a description of PCM for a single parameter uncertainty, a brief digression to discuss the basic "brute force" Monte Carlo method is fruitful. We start with a single uncertain parameter because the connections to the underlying theory are more easily seen in this case. In a later section, we discuss the extension to multiple uncertain parameters, which is our real interest.

2.2.1 Brute Force Monte Carlo

In its simplest form, the Monte Carlo method is a method of repeated trials. As an example, let $f(x)$ be a probability density function (PDF) describing some uncertain parameter x:

$$\begin{aligned} f(x) &\geq 0, \text{ for all x} && (2.14) \\ \int_{\mathcal{A}} f(x)dx &= 1 && (2.15) \end{aligned}$$

In other words, a PDF must be nonnegative and also must integrate to unity over its domain $\mathcal{A}$. For some desired output which is a function of that parameter, $g(x)$, the expected value of the output is computed using the following integral:

$$\int_{\mathcal{A}} f(x)g(x)dx = \mathrm{E}[g(x)] \qquad (2.16)$$

Before we continue, we would like to point out that this integral is exactly the form we discussed in the context of Gaussian quadrature integration if $g(x)$ is a polynomial. We return to this similarity shortly.

A simplistic Monte Carlo method approximates the integral by generating random numbers using $f(x)$ and then performing a simple average of the resulting answers:

$$\int_{\mathcal{A}} f(x)g(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} g(\hat{x}_i) \qquad (2.17)$$

where the $\hat{x}_i$ are generated according to $f(x)$ using a pseudo-random number generator. Given sufficiently large N, this method is guaranteed to converge to the actual expected value as a direct result of the law of large numbers [34]. Unfortunately, the required N may be quite large. Since all of the moments of $g(x)$ (e.g., the variance) are expected values, the law of large numbers applies to these statistics as well under the Monte Carlo method.

2.2.2 Single Parameter PCM

The probabilistic collocation method is essentially a polynomial modeling technique. The desired output is described as a polynomial in the uncertain parameter of the system. After this model is identified, a standard, simplistic Monte Carlo method can be applied to the polynomial model. The problem with (2.17) is that each $g(x_i)$ is computationally expensive, since it represents a separate simulation of the power system. If this function $g(x)$ can be modeled reasonably accurately by a polynomial $\hat{g}(x)$, an essentially unlimited number of samples $\hat{g}(x_i)$ can be computed because no simulations are involved once the polynomial has been identified. The only simulations necessary are those to identify the polynomial. Therein lies the power of the probabilistic collocation method.

Concretely, let x be the uncertain parameter, $f(x)$ be the PDF describing this parameter and $g(x)$ be the output of interest. We desire an equation of the form:

$$\hat{g}(x) = g_0' + g_1'x + \cdots + g_n'x^n \tag{2.18}$$

where the g_i' are constants. The model parameters could be identified by performing many simulations using various x_i and using the results in a least squares algorithm. However, we are trying to perform an absolute minimal number of simulations, since each simulation is expensive in terms of time.

If we choose to model the relationship between the output and the uncertain parameter as an order $(2n-1)$ polynomial, we need minimally $2n$ simulations to find all of the coefficients. However, we can do "better" if we are primarily interested in just matching the expected value of the output. If we compare (2.8) and (2.16), we see that the integrals are identical! In other words, if $g(x)$ is really a polynomial of order less than or equal to $(2n-1)$, we have the following result:

$$\mathrm{E}[g(x)] = \mathrm{E}[\hat{g}(x)] = \sum_{i=1}^{n} f_i g(x_i) \tag{2.19}$$

We can use $f(x)$ to compute orthonormal polynomials $\mathcal{H}$ and calculate the roots of $h_n(x)$. We can then perform simulations with parameter values at these roots and use those results to find the coefficients of our polynomial approximation. We reiterate a key point: the values for which the simulations are performed are completely independent of the particular *output* of interest. The same set of simulations can be used to fit the polynomial models for *all* of the outputs of interest.

Continuing, let us represent $\hat{g}(x)$ using the polynomials in $\mathcal{H}$ without loss of generality, where $\hat{g}(x)$ is of order $(n-1)$:

$$\hat{g}(x) = g_0 h_0(x) + g_1 h_1(x) + \cdots + g_{n-1} h_{n-1}(x) \tag{2.20}$$

The following linear system of equations can be solved to determine the g_i:

$$\begin{bmatrix} g(x_1) \\ \vdots \\ g(x_n) \end{bmatrix} = \begin{bmatrix} h_{n-1}(x_1) & \cdots & h_0(x_1) \\ \vdots & \ddots & \vdots \\ h_{n-1}(x_n) & \cdots & h_0(x_n) \end{bmatrix} \begin{bmatrix} g_{n-1} \\ \vdots \\ g_0 \end{bmatrix} \tag{2.21}$$

These equations are identical to (2.11), which implies that, if $g(x)$ is really a polynomial of order less than or equal to $(2n-1)$, the expected value computed using $\hat{g}(x)$ is exact and is in fact $g_0 \int_{\mathcal{A}} f(x) h_0(x) dx$. So, if $h_0(x) = 1$, the expected value is simply g_0 and we need do no further calculations. Similar relationships exist for higher moments of $g(x)$. For example, a direct application of orthogonality to the definition of variance yields an expression for the variance:

$$\sigma^2_{g(x)} = \sum_{i=1}^{n-1} g_i^2 \tag{2.22}$$

In general, this approach is heuristic. We cannot usually guarantee that the actual relationship between the uncertain parameter and the output of interest is exactly a polynomial. In addition, we may be interested in further statistical information about the output, not just its moments. As previously noted, higher order moments are also expected values, so the arguments for the expected value hold just as well for the variance, although a higher-order polynomial model may be required, but further statistical information, such as the exact probability that the output lives in some interval, may not be accurate. PCM is designed to calculate the coefficients of the polynomial model with the bare minimal number of simulations while trying also to reproduce the moments of the output with high fidelity by modeling the polynomial particularly well in the regions that are more probable. Given that, one could also expect that the PCM model would perform well when used to compute the probability of events in the high probability region of the output.

2.2.3 Variance Reduction Techniques - Importance Sampling

One might ask whether the probabilistic collocation method is necessary or, said differently, whether other better techniques exist. Most work on Monte Carlo simulation speed-up centers on a variety of methods collectively referred to as variance reduction techniques. Many such techniques exist (see [31]), but we present only one representative method, importance sampling.

In general, for a sufficiently large number of simulations N the Monte Carlo method will converge because of the law of large numbers. However, the number of simulations required may be unreasonably large. The number of simulations necessary to guarantee that the computed expected value is within a neighborhood of the actual expected value with a given probability depends on the variance of the output variable. A larger variance will generally require more simulations to guarantee an accurate result. As the name implies, variance reduction techniques seek to modify the problem to obtain an output with lower variance.

In all cases, variance reduction techniques require knowledge about the relationship between the uncertain parameter and the output of interest. Obtaining such information may require a few initial simulations, which is one drawback of variance reduction techniques. A further limitation is that variance reduction techniques fundamentally alter the problem at hand in such a way that the same set of simulations cannot be used to calculate statistics for all possible outputs of interest. Examining importance sampling more closely makes this last point clear.

Consider again the initial expected value problem:

$$\mathrm{E}[g(x)] = \int_{\mathcal{A}} f(x)g(x)dx \tag{2.23}$$

We can express this alternately in the following way:

$$\mathrm{E}[g(x)] = \int_{\mathcal{A}} \frac{f(x)g(x)}{k(x)}k(x)dx \tag{2.24}$$

where $k(x)$ is any legitimate probability density function. We refer to $k(x)$ as the importance sampling distribution. Now, we have two alternate Monte Carlo representations for the expected value:

$$\begin{aligned}
\mathrm{E}[g(x)] &\approx \frac{1}{N}\sum_{i=1}^{N} g(\hat{x}_i) && (2.25)\\
\mathrm{E}[g(x)] &\approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(\hat{x}_i)g(\hat{x}_i)}{k(\hat{x}_i)} && (2.26)
\end{aligned}$$

where the $\hat{x}_i$ are generated using $f(x)$ in (2.25) and using $k(x)$ in (2.26). The law of large numbers guarantees that for a sufficiently large N both estimates approach the correct answer. However, for an intelligently chosen $k(x)$, the variance of $\frac{f(x)g(x)}{k(x)}$ is lower than that of $g(x)$, and the importance sampling estimate converges more quickly.

The optimal $k(x)$ is given by:

$$k(x) = \frac{|g(x)|f(x)}{\int_{\mathcal{A}} |g(x)|f(x)dx} \tag{2.27}$$

It is straightforward to verify that this $k(x)$ is a legitimate PDF and that this choice of $k(x)$ forces the variance of $\frac{f(x)g(x)}{k(x)}$ to zero! However, two important observations cloud this seemingly sunny result. First, the optimal $k(x)$ is a function of a quantity very similar to the one we are trying to compute. If we know the expected value of the absolute value of $g(x)$ already, we have almost solved our problem. Picking a good importance sampling distribution requires a good deal of knowledge about the system and is by no means trivial. Second, even if a good importance sampling distribution could be found for a given output variable, the result is extremely limited. While the expected value can be computed exactly using this distribution, the higher-order statistics of the output are modified, which should

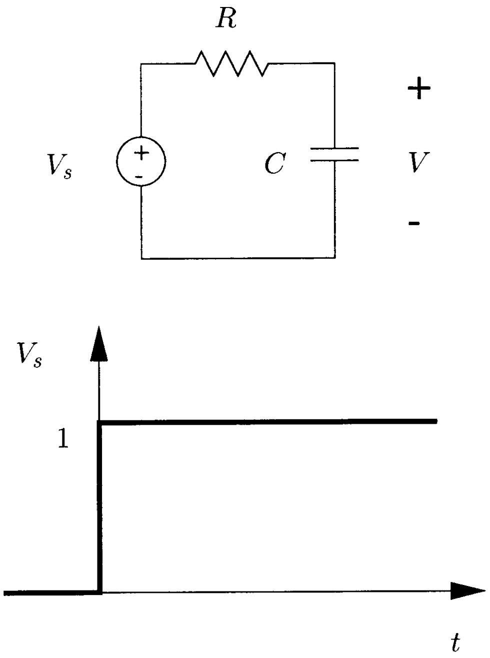


Figure 2.1: **Simple RC-Circuit Example.** The simple first-order circuit shown above has zero initial conditions at time zero. The example discussed is a step input at the voltage source with the output being the voltage across the capacitor at a particular instant of time.

be obvious because our goal was to reduce the variance. A different importance sampling distribution is needed for every statistic of a given output. To compound this problem, the importance sampling distribution is optimized for the expected value of a particular output of interest; if we desire multiple output variables, the variance may be significantly increased for some expected values while others are decreased, with no net gain.

While variance reduction techniques are powerful tools, they have significant practical limitations if multiple outputs and multiple statistics are needed.

2.2.4 Simple Circuit Example

To illustrate the probabilistic collocation method, a simple first-order circuit, such as the one shown in Figure 2.1, may be helpful because of both its great familiarity and its tractability to analytical techniques. The response voltage is described by a first-order differential

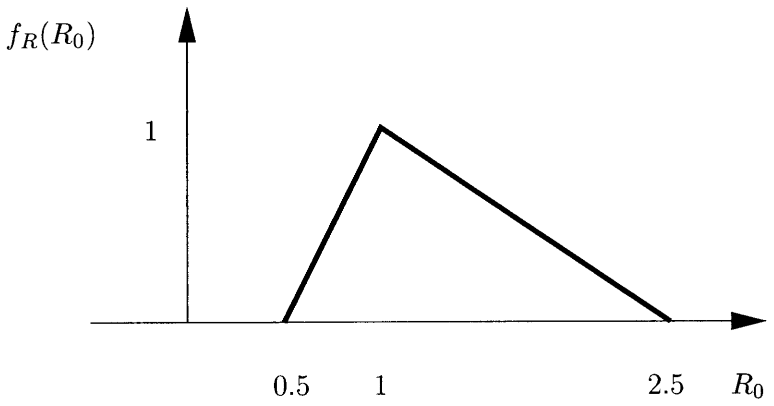


Figure 2.2: **PDF for Resistance *R*.** The PDF shown above is used as the uncertainty description for the first-order circuit example.

equation:

$$\dot{V} = -\frac{V}{RC} + \frac{V_s}{RC} \tag{2.28}$$

Given zero initial conditions and the step input shown, the output is a familiar exponential response:

$$V(t) = 1 - e^{-t/RC} \tag{2.29}$$

Since only single parameter techniques have been discussed so far, the resistance is treated as uncertain and the capacitance is treated as if it were known exactly. The probabilistic collocation method is only presently able to handle scalar outputs, though it is able to accommodate an unlimited number of scalar outputs with the same small set of system simulations. We choose the voltage at time $C/10$ as our output of interest g for this example:

$$g(R) = V(C/10) = 1 - e^{-1/10R} \tag{2.30}$$

Before any uncertainty analysis can be performed, a description of the parameter uncertainty must be available. We use a somewhat unusual probability density function to describe the uncertainty in R to demonstrate that the probabilistic collocation method can accommodate any desired PDF:

$$f_R(R_0) = \begin{cases} 2R_0 - 1 & \frac{1}{2} \le R_0 < 1 \\ -\frac{2}{3}R_0 + \frac{5}{3} & 1 \le R_0 \le \frac{5}{2} \end{cases} \tag{2.31}$$

This PDF is shown graphically in Figure 2.2. For completeness, we compute the expected value and variance of R:

$$\mathrm{E}[R] = \frac{4}{3} \tag{2.32}$$

$$\sigma_R^2 = \frac{13}{72} = 0.1806 \tag{2.33}$$

Since an algebraic expression for g as a function of R is available, the PDF for g can be derived directly, along with its associated moments:

$$f_g(g_0) = \begin{cases} (\frac{2}{30\ln(1-g_0)} + \frac{5}{3})(\frac{1}{10(1-g_0)(\ln(1-g_0))^2}) & 1 - e^{-1/25} \le g_0 < 1 - e^{-1/10} \\ (\frac{-2}{10\ln(1-g_0)} - 1)(\frac{1}{10(1-g_0)(\ln(1-g_0))^2}) & 1 - e^{-1/10} \le g_0 < 1 - e^{-1/5} \end{cases} \tag{2.34}$$

where the expected value, variance and standard deviation of g are as follows:

$$\mathrm{E}[g] = 0.0797 \tag{2.35}$$

$$\sigma_g^2 = 0.000660 \tag{2.36}$$

$$\sigma_g = 0.0257 \tag{2.37}$$

At this point, we would like to demonstrate how to apply PCM to this problem. In normal practice, one would not use PCM in such a situation, since the relationship between the uncertain parameter and the output of interest is known analytically, but having the exact answer for comparison purposes is helpful pedagogically. The first step is to obtain the first few orthogonal polynomials based on $f_R(R_0)$:

$$h_0(R) = 1 \tag{2.38}$$

$$h_1(R) = \frac{12}{\sqrt{26}}\left(R - \frac{4}{3}\right) \tag{2.39}$$

$$h_2(R) = \frac{1560}{\sqrt{96915}}\left(R^2 - \frac{37}{13}R + \frac{191}{104}\right) \tag{2.40}$$

$$h_3(R) = \frac{39760}{\sqrt{15174901}}\left(R^3 - \frac{2157}{497}R^2 + \frac{29163}{4970}R - \frac{6087}{2485}\right) \tag{2.41}$$

The roots of $h_2(R)$ yield two values of R, which we can use to develop a linear approximation of the relationship between g and R. The roots of $h_2(R)$ and the resulting values of g are as follows:

$$R = 1.857 \quad \rightarrow \quad g(R) = 0.0524 \tag{2.42}$$

$$R = 0.989 \quad \rightarrow \quad g(R) = 0.0962 \tag{2.43}$$

Since two points define a line and $h_0(R)$ and $h_1(R)$ form a basis for the space of all linear functions of R, the PCM approximation may be written as follows:

$$\tilde{g}(R) = -0.0215\, h_1(R) + 0.0787\, h_0(R) = (-0.0505)R + 0.146 \tag{2.44}$$

The expected value, variance and standard deviation are directly available from the coefficients, as noted earlier:

$$\mathrm{E}[\tilde{g}] \;=\; 0.0787 \tag{2.45}$$

$$\sigma_{\tilde{g}}^2 \;=\; 0.000462 \tag{2.46}$$

$$\sigma_{\tilde{g}} \;=\; 0.0215 \tag{2.47}$$

The accuracy of these results is startling considering that they are based on a linear approximation and that approximation was created using two points selected based on the PDF for R and **not** based on the relationship between g and R.

As a basis of comparison, we also examine another standard way to obtain a linear approximation of $g(R)$, namely, a Taylor series approach. Instead of creating a line using two widely separated values of R, we select a nominal value for R and another value of R close to the nominal value, which results in a line tangential to the actual relationship between g and R at that point. One might select $R = 1$ for the nominal value, since the PDF for R is maximal there. Another good choice is $R = 4/3$, since that is the expected value of R. We refer to the resulting linear models as $\hat{g}_1$ and $\hat{g}_{4/3}$. The PCM linear model as well as these two linear approximations are shown in Figure 2.3.

The equation describing $\hat{g}_1$ as well as its expected value, variance and standard deviation

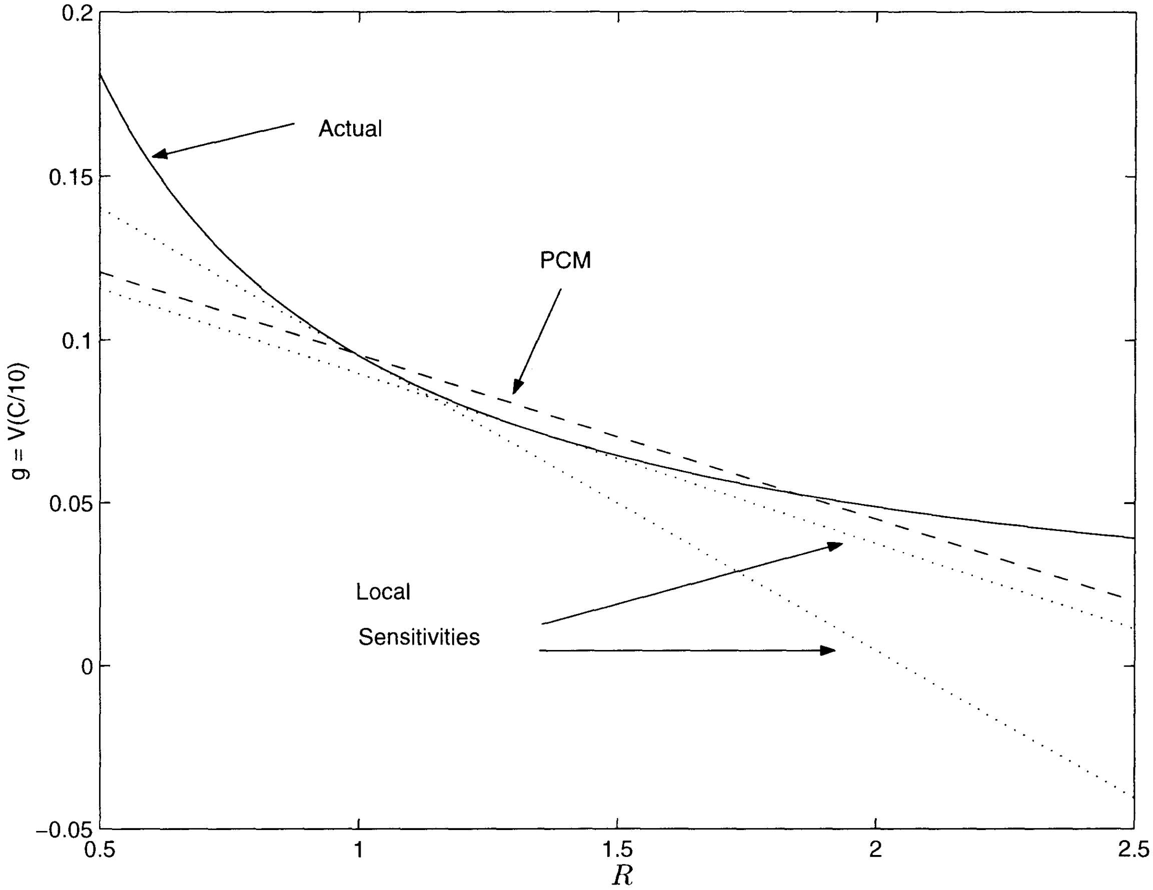


Figure 2.3: **Comparison of Linear Approximations.** Three different linear approximations to the relationship between the response voltage at a particular time and the uncertain system parameter R are presented here. The full line is the actual relationship. The dashed line represents a linear approximation calculated using PCM. The two dotted lines represent two different approximations based on a Taylor series expansion.

are as follows:

$$\begin{aligned}\hat{g}_1(R) &= (-0.0905)R + 0.186 && (2.48)\\ \mathrm{E}[\hat{g}_1] &= 0.0650 && (2.49)\\ \sigma^2_{\hat{g}_1} &= 0.00148 && (2.50)\\ \sigma_{\hat{g}_1} &= 0.0385 && (2.51)\end{aligned}$$

The error in the prediction of the expected value, variance and standard deviation is substantially worse than that with the probabilistic collocation method. Figure 2.3 shows that the matching for higher values of R with $\hat{g}_1$ is abysmal, which explains the poor predictions for the moments.

One might expect that $\hat{g}_{4/3}$ would perform better since it is an approximation around a more meaningful value of R. The equation of this line, its expected value, variance and standard deviation are as follows:

$$\begin{aligned}\hat{g}_{4/3}(R) &= (-0.0522)R + 0.142 && (2.52)\\ \mathrm{E}[\hat{g}_{4/3}] &= 0.0723 && (2.53)\\ \sigma^2_{\hat{g}_{4/3}} &= 0.000492 && (2.54)\\ \sigma_{\hat{g}_{4/3}} &= 0.0222 && (2.55)\end{aligned}$$

This line is very similar to that obtained with the probabilistic collocation method, and, as a result, the moments are also fairly comparable. However, though the variance and standard deviation are nearly identical to that of the PCM model, the expected value as computed using the Taylor series approximation model has a noticeably greater error than the expected value as computed using the PCM model. This result is not surprising since $\hat{g}_{4/3}$ is always less than g over the range of interest. The PCM model is able to perform better since it reproduces the desirable slope of $\hat{g}_{4/3}$ but is not limited to being tangential to g.

As a final comparison of the linear approximations of g, Figure 2.4 presents the resulting PDFs for g, $\tilde{g}$, $\hat{g}_1$ and $\hat{g}_{4/3}$. Since the approximations to g are all linear, their PDFs must be scaled and translated versions of the PDF of R. Obviously, they cannot hope to accurately represent the true PDF of g, since it is not composed of two linear segments. However, this

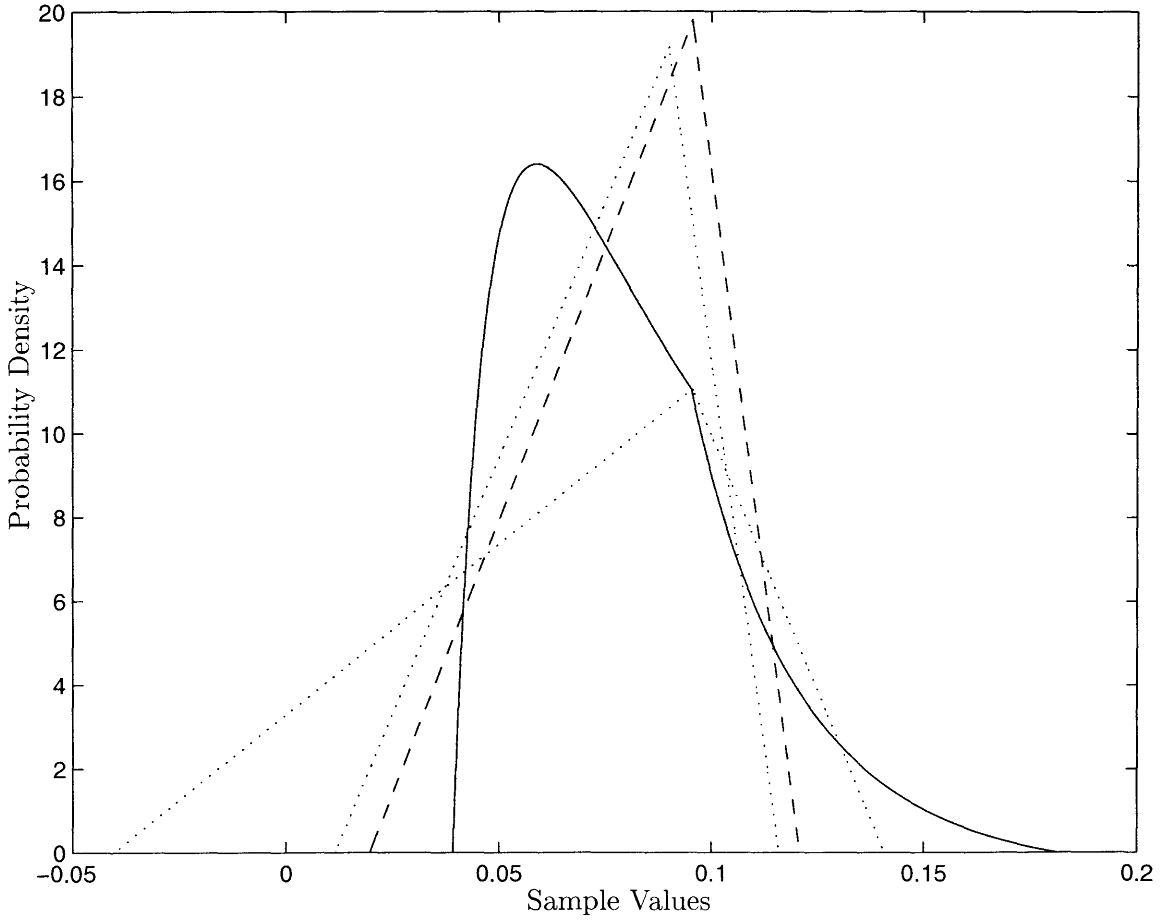


Figure 2.4: **PDFs from Linear Approximations.** Three different linear approximations to the relationship between the response voltage at a particular time and the uncertain system parameter R are used to calculate the probability density function of g. The full line is the actual PDF of g. The dashed line is the PDF of $\tilde{g}$. The two dotted lines are the PDFs of $\hat{g}_1$ and $\hat{g}_{4/3}$.

figure clearly shows that $\hat{g}_1$ is a substantially worse approximation to g than either $\tilde{g}$ or $\hat{g}_{4/3}$. The key to its poor performance is that $\hat{g}_1$ has a PDF which approximates the true PDF well for large sample values but at the cost of poorly reproducing the main lobe of probability. On the other hand, $\tilde{g}$ and $\hat{g}_{4/3}$ have PDFs which approximate the outline of the main lobe of probability without reproducing the particular structure of the PDF of g at any point. The PCM model $\tilde{g}$ does this by design, whereas $\hat{g}_{4/3}$ happens to be a particularly good Taylor series approximation for this problem. Since the analytical relationship between the uncertain parameter and the output of interest is not available in a realistic problem, it is not possible in general to predict which nominal values will lead to good Taylor series approximations and which will not, which severely limits the utility of the technique in this context.

We can also create higher-order polynomial approximations using PCM. To create a quadratic model, the roots of $h_3(R)$ are used to find the coefficients. The roots of h_3 and the corresponding values of $g(R)$ are as follows:

$$R = 2.106 \quad \rightarrow \quad g(R) = 0.0464 \tag{2.56}$$

$$R = 1.407 \quad \rightarrow \quad g(R) = 0.0686 \tag{2.57}$$

$$R = 0.826 \quad \rightarrow \quad g(R) = 0.1140 \tag{2.58}$$

The quadratic PCM model is as follows:

$$\tilde{g}(R) = 0.0072\, h_2(R) - 0.0238\, h_1(R) + 0.0795\, h_0(R) = (0.0362)R^2 - (0.159)R + 0.221 \tag{2.59}$$

The moments can be computed directly from the coefficients, just as for the linear model:

$$\mathrm{E}[\tilde{g}] = 0.0795 \tag{2.60}$$

$$\sigma^2_{\tilde{g}} = 0.000618 \tag{2.61}$$

$$\sigma_{\tilde{g}} = 0.0249 \tag{2.62}$$

The PCM quadratic model almost exactly reproduces the true expected value, variance and standard deviation using only three samples of g. As Figure 2.5 shows, the quadratic model more closely matches g than the linear model does. For completeness, we also present the PDF of the quadratic model in Figure 2.6; again, the PDF of the quadratic model is not

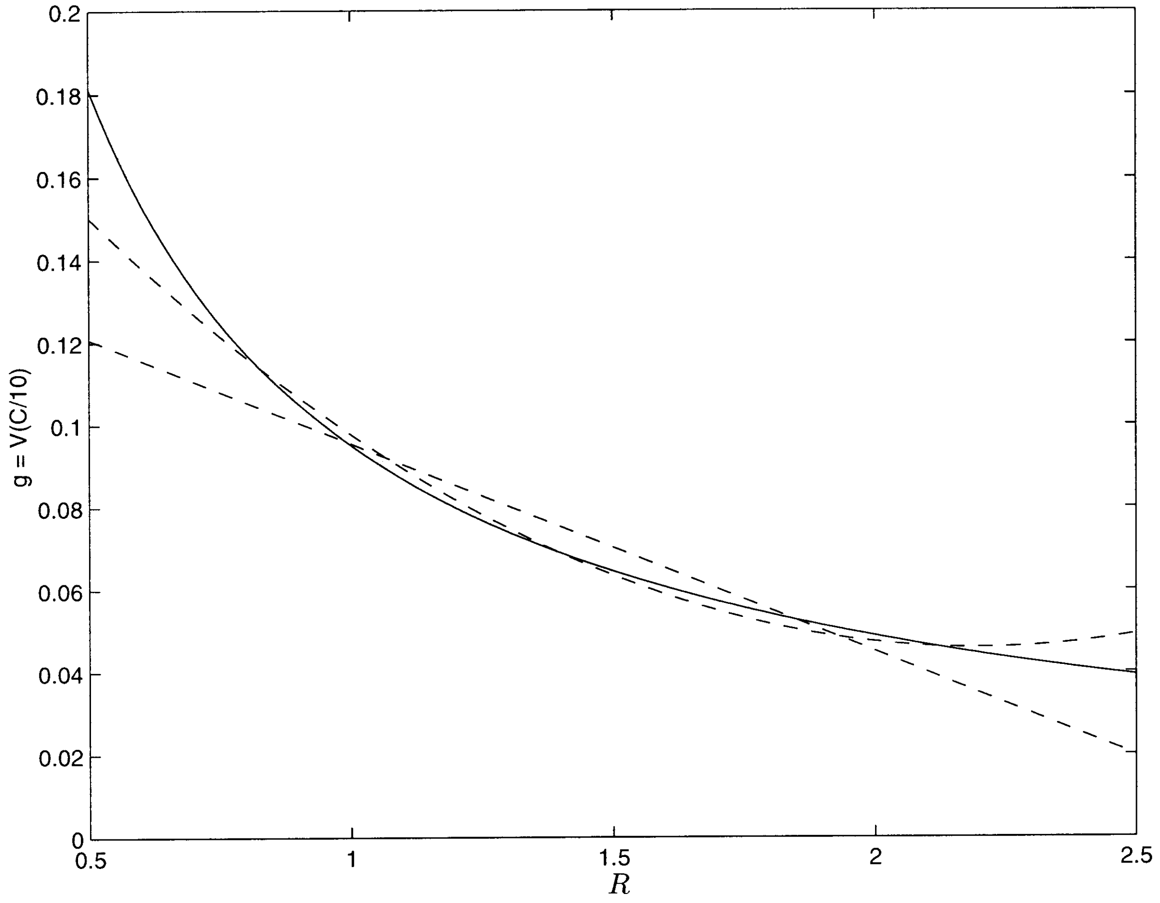


Figure 2.5: **Comparison of Quadratic and Linear PCM.** The linear and quadratic PCM approximations to the relationship between the response voltage at a particular time and the uncertain system parameter R are compared in this figure. The full line is the actual relationship. The dashed lines are the linear and quadratic PCM models.

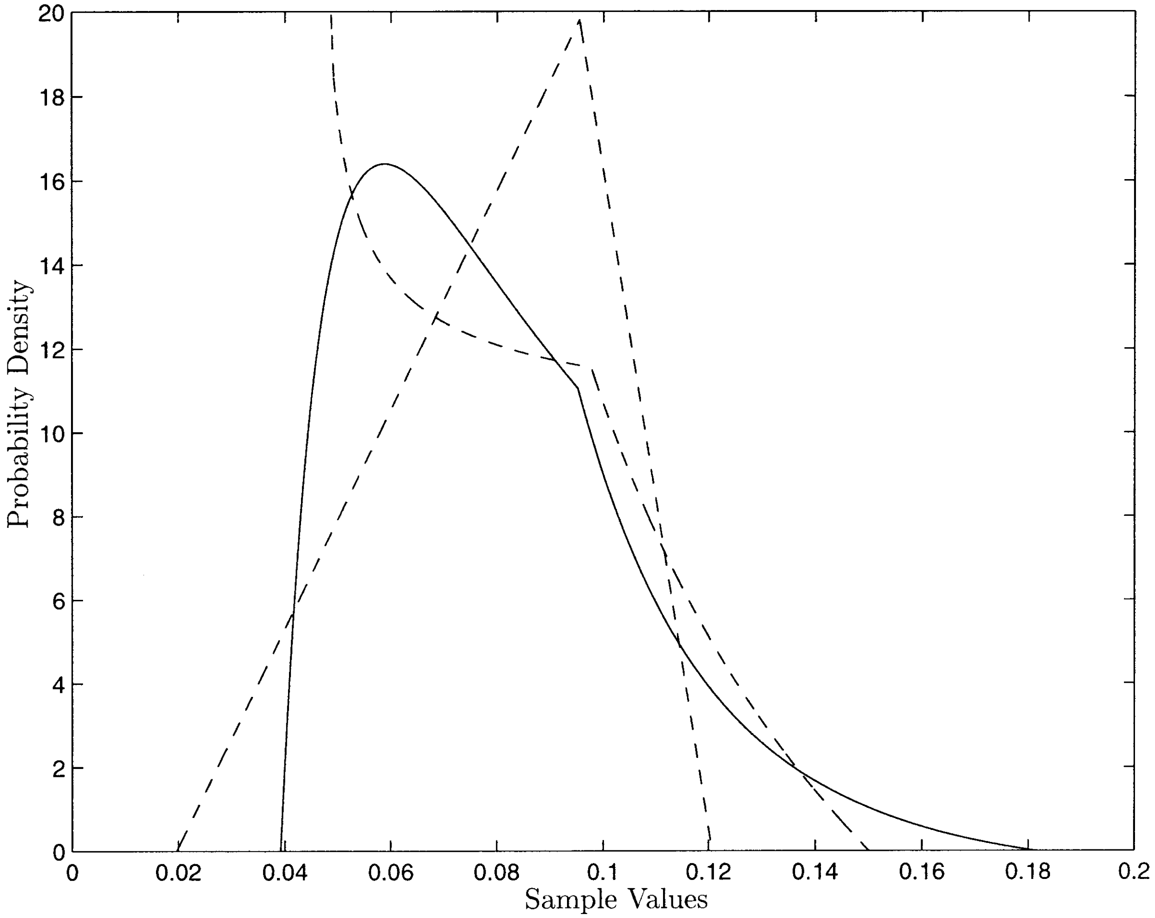


Figure 2.6: **Quadratic and Linear Model PDFs.** The linear and quadratic PCM approximations to the relationship between the response voltage at a particular time and the uncertain system parameter R are used to calculate the probability density function of g. The full line is the actual PDF of g. The dashed lines are the PDFs calculated using the linear and quadratic PCM models.

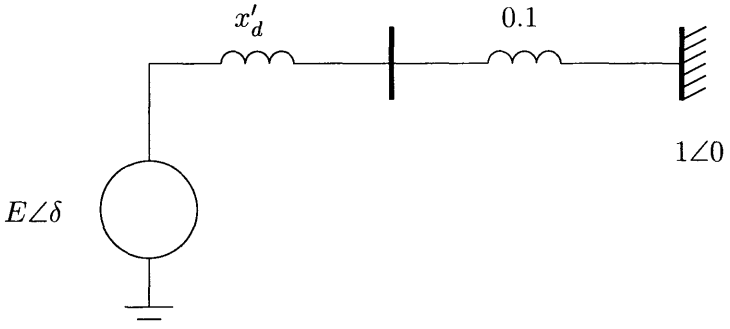


Figure 2.7: **Single Machine, Infinite Bus.** This schematic represents a classical single machine, infinite bus system. This system has no shunts and damping is neglected.

able to match the actual PDF of g exactly, but it does capture the approximate size and shape of the main probability lobe well enough for the expected value and variance to be nearly exact.

2.2.5 Single Machine, Infinite Bus Example

Another standard, familiar example is the single machine, infinite bus system shown in Figure 2.7. The equation governing this system is as follows:

$$\frac{2H}{\omega_R}\frac{d^2\delta}{dt^2} = P_m - P \tag{2.63}$$

where H is the inertia constant of the generator, ω_R is the angular speed reference of the system, δ is the internal angle of the generator, P_m is the mechanical input power to the generator and P is the electrical output power of the generator. For this example, we use $H = 5\text{MJ/MVA}$, $\omega_R = 2\pi 60\text{rad/sec}$, and the following initial conditions:

$$E = 1.108 \text{ p.u.} \tag{2.64}$$

$$P_0 = P_m = 1 \text{ p.u.} \tag{2.65}$$

$$\delta_0 = 0.1815 \text{ radians} \tag{2.66}$$

$$\left(\frac{d\delta}{dt}\right)_0 = 0 \tag{2.67}$$

which give us a single nonlinear, second-order differential equation to describe the system:

$$\frac{d^2\delta}{dt^2} = \frac{377}{10}(P_m - 5.54\sin\delta) \tag{2.68}$$

In the circuit example, we explored a single system parameter uncertainty, where we refer to parameters intrinsic to the system and not directly related to any particular simulation as system parameters. It is also possible to study parameter uncertainties which are related to a particular simulation. To demonstrate this, we increase the mechanical input power to the generator at time zero by one per unit and designate the length of time until the mechanical input power is returned to its initial value of one per unit to be uncertain. In other words, the mechanical input power is a pulse and its duration is uncertain. The duration τ in seconds is described by the following PDF:

$$f_\tau(\tau_0) = (0.0783)e^{30\tau_0} \text{ for } 0.1 \leq \tau_0 \leq 0.2 \tag{2.69}$$

where the PDF is zero everywhere outside this range. The maximum value of the generator internal angle during the simulation is chosen to be the output of interest g.

The relationship between g and τ is well approximated over the region of interest by a quadratic polynomial. Therefore, we concentrate on linear approximations in this example because a good quadratic model could be obtained by simulating the system at almost any three reasonable points. The power of the probabilistic collocation method lies in its ability to select appropriate simulation points to create a polynomial model which has the same moments as g even when g cannot be modeled by a polynomial of that order. The first three orthogonal polynomials based on f_τ are as follows:

$$h_0(\tau) = 1 \tag{2.70}$$

$$h_1(\tau) = 42.3(\tau - 0.1719) \tag{2.71}$$

$$h_2(\tau) = 1609(\tau^2 - (0.320)\tau + 0.0250) \tag{2.72}$$

The roots of $h_2(\tau)$ and the resulting values of g are as follows:

$$\tau = 0.187 \rightarrow g(\tau) = 0.548 \tag{2.73}$$

$$\tau = 0.134 \rightarrow g(\tau) = 0.487 \tag{2.74}$$

The PCM linear model is then easily computed:

$$\tilde{g}(\tau) = 0.0275\, h_1(\tau) + 0.530\, h_0(\tau) = (1.165)\tau + 0.3308 \tag{2.75}$$

as are the moments:

$$\begin{aligned} \mathrm{E}[\tilde{g}] &= 0.531 && (2.76)\\ \sigma_{\tilde{g}}^2 &= 0.000756 && (2.77)\\ \sigma_{\tilde{g}} &= 0.0275 && (2.78) \end{aligned}$$

The actual moments are very well approximated by the PCM model:

$$\begin{aligned} \mathrm{E}[g] &= 0.531 && (2.79)\\ \sigma_g^2 &= 0.000686 && (2.80)\\ \sigma_g &= 0.0262 && (2.81) \end{aligned}$$

Again, one could also choose other simulation points and create a different linear model. As a basis of comparison, we create a linear Taylor series approximation $\hat{g}$ around the nominal value $\tau = 0.172$, which is the expected value of τ. The resulting linear model is:

$$\hat{g}(\tau) = (0.9851)\tau + 0.3661 \tag{2.82}$$

and has the following moments:

$$\begin{aligned} \mathrm{E}[\hat{g}] &= 0.535 && (2.83)\\ \sigma_{\hat{g}}^2 &= 0.000543 && (2.84)\\ \sigma_{\hat{g}} &= 0.0233 && (2.85) \end{aligned}$$

While the expected value is not as accurate as the one computed with PCM, it is still very close to the true value. However, the Taylor series approximation variance and standard deviation errors are substantially greater than those with PCM. The two models are shown along with the true g in Figure 2.8. As with the previous example, the Taylor series approximation is tangential to g and represents a good local approximation at the nominal value, but it does not perform as well as the PCM model does globally.

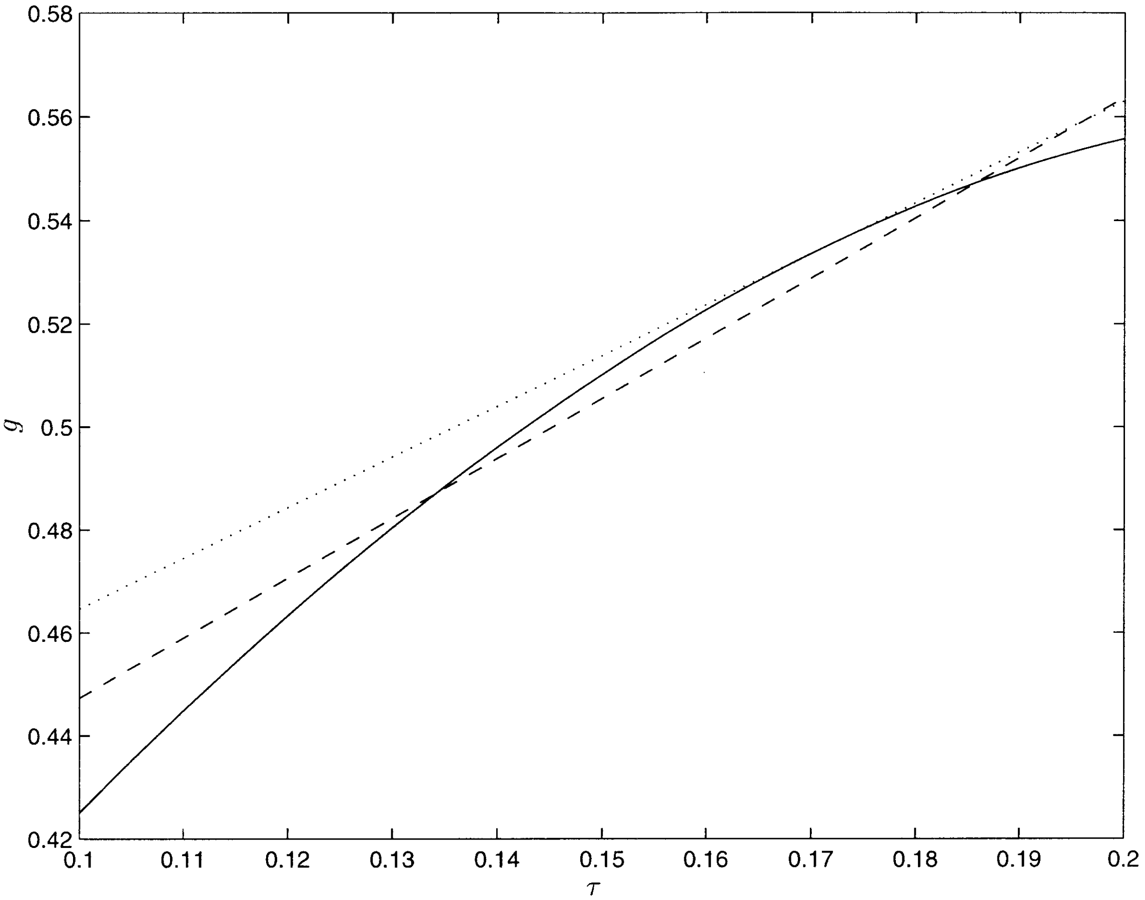


Figure 2.8: **Linear Approximations for SMIB Example.** Two different linear approximations to the relationship between the maximum generator internal angle and the uncertain simulation parameter τ are presented here. The full line is the actual relationship. The dashed line represents a linear approximation calculated using PCM. The dotted line represents an approximation based on a Taylor series expansion.

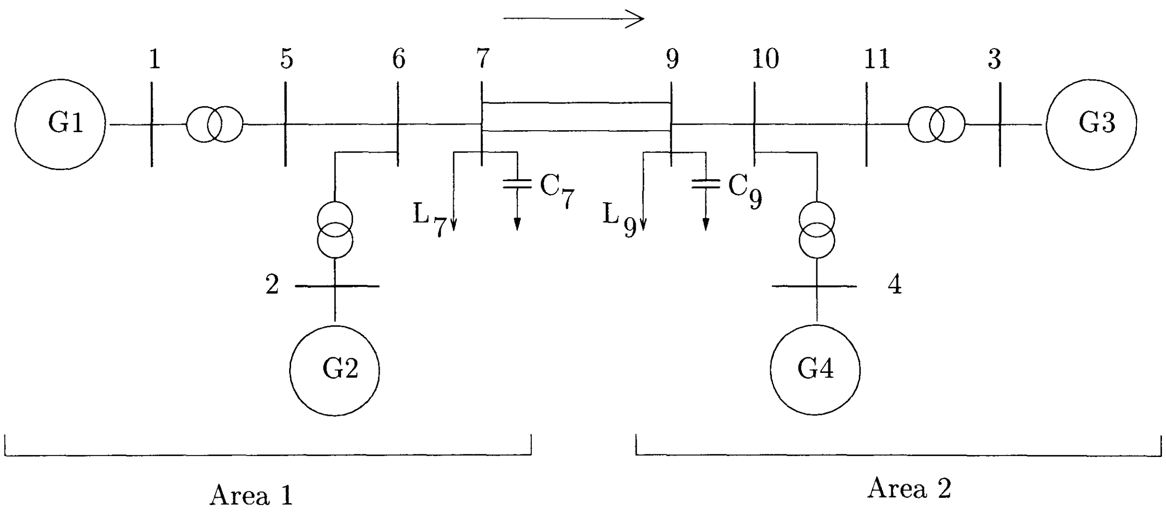


Figure 2.9: **Multiple Machine System.** This small power system model is adapted from [35]. The same parameters and models as in the reference are used, excepting the power transfer between the two areas and the load parameter α. Changes to the power transfer are implemented by proportionally changing the generation at each of the four generators.

2.2.6 Multiple Machine Power System Example

As a final illustration, we present the example explored in [17]. The power system model comprises a two-area system adapted from [35] and shown in Figure 2.9. With the nominal model, we explore the amount of power which can be transferred between the two areas. A power transfer is deemed unacceptable if voltage or frequency limits are exceeded during the transient following a fault and reset on one of the two lines connecting buses 7 and 9. (The trivial bus 8 from [35] was removed). If Area 1 and Area 2 each represent a utility, we are attempting to determine the maximum amount of power which can safely be transferred from one utility to the other.

Through simulation studies on the nominal system and increasing the power transfer in 50MW increments, we see that at a power transfer of 450MW the minimum voltage limit is violated during the post-fault transient, while a 400MW transfer is acceptable. The question we seek to answer is whether the 400MW transfer scenario is acceptable for all reasonable parameter variations of a single parameter within the model, and, if it is not, what is the likelihood that a violation occurs. Such information is crucial if the system is to be operated safely while the need and/or desire to transfer power from one area to the other is also considered.

	Monte Carlo 7000 pts.	PCM, second-order 3 pts.
$\mathrm{E}[V_{min}]$	0.913	0.913
$\mathrm{SD}[V_{min}]$	0.00830	0.00833
Prob. of Violation	7.82%	8.92%

Table 2.1: **Comparison of Monte Carlo and PCM.** We present a comparison of the results obtained using a standard brute force Monte Carlo method and using the probabilistic collocation method.

For this study, we model one of the load parameters as uncertain. The standard real power load model discussed earlier is used in this example, namely:

$$P = P_0|V|^{\alpha} \tag{2.86}$$

In the reference, the parameter α is nominally 1.0. Here we assume that it is uniformly distributed between 0.75 and 2.0. We vary the load parameter simultaneously and identically at each of the two load locations.

In all cases when the system transient is unacceptable, the minimum voltage constraint is violated in this system. Therefore, we concentrate on three quantities in our study: the probability of violation, the expected value of the minimum system voltage during the transient, and the standard deviation of the minimum system voltage during the transient. A comparison of the results of our studies using both Monte Carlo methods and the probabilistic collocation method is presented in Table 2.1.

For the assumed uniform distribution, the orthonormal polynomials up to third order are as follows:

$$\begin{aligned}
h_0(\alpha) &= 1 && (2.87)\\
h_1(\alpha) &= \frac{\sqrt{3}}{5}(8\alpha - 11) && (2.88)\\
h_2(\alpha) &= \frac{1}{5\sqrt{5}}(96\alpha^2 - 264\alpha + 169) && (2.89)\\
h_3(\alpha) &= \frac{\sqrt{7}}{25}(256\alpha^3 - 1056\alpha^2 + 1392\alpha - 583) && (2.90)
\end{aligned}$$

The roots of the third-order polynomial and the corresponding system minimum voltages given those parameter values for α are as follows:

$$\alpha = \frac{1}{8}(11 - \sqrt{15}) \;\rightarrow\; V_{min} = 0.901 \tag{2.91}$$

$$\alpha = \frac{11}{8} \;\rightarrow\; V_{min} = 0.915 \tag{2.92}$$

$$\alpha = \frac{1}{8}(11 + \sqrt{15}) \;\rightarrow\; V_{min} = 0.923 \tag{2.93}$$

From only these three simulations, a second-order representation of the relationship between α and V_{min} is easily calculated:

$$\hat{V}_{min} = 0.91333 + 0.0081989\, h_1(\alpha) - 0.0014907\, h_2(\alpha) \tag{2.94}$$

Literally, $\hat{V}_{min}$ can be evaluated at thousands of randomly generated α in less than a second. The three simulations to fit the polynomial are the only simulations we perform on the power system model itself.

As the table shows, our results using PCM are very good considering that we require only three simulations. As expected, the moments can be calculated to a high degree of accuracy. Even the probability of violation, which is a low probability event, is reasonably approximated. To verify our results, 7000 Monte Carlo simulations are used as a basis of comparison. Realistically, only about 1500 Monte Carlo simulations are actually necessary to obtain results with comparable accuracy to those obtained using the probabilistic collocation method.

To complete our discussion of the probability of violation, Table 2.2 presents results obtained using progressively higher-order polynomial representations. Determining the appropriate order of the polynomial model for a given problem remains an unresolved issue.

Another goal is to perform such studies on time-varying outputs of our system. To illustrate, we take this same example and study the voltage at Bus 7 after the fault is cleared. In this case, we seek something akin to error bars for our simulation. To accomplish this, we plot the expected value of the voltage at each time point along with two additional curves at plus and minus one standard deviation from the mean. Figure 2.10 summarizes our results using both brute force Monte Carlo and PCM. The correspondence between the two methods is striking, as is the considerable computational savings of PCM.

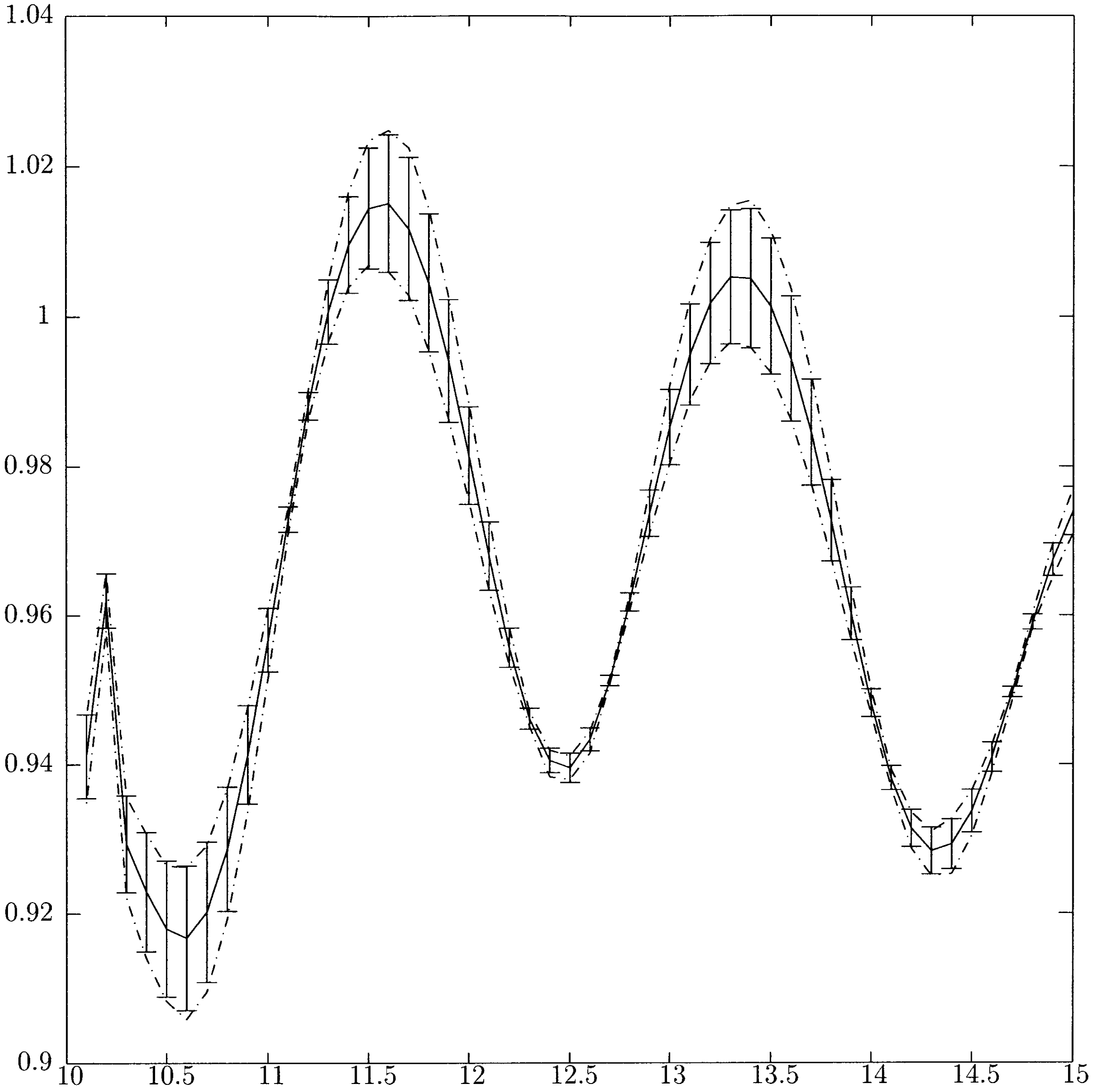


Figure 2.10: **PCM Time-Varying Example.** We present a plot of the response of the voltage at Bus 7 for the five seconds following the event. The solid line represents the expected value of the voltage as calculated by either a Monte Carlo method or PCM (they yield essentially identical results here). The error bars represent ±1 standard deviation from the mean as calculated using a brute force Monte Carlo method. The "dot-dash" lines represent ±1 standard deviation from the mean as calculated using PCM. The PCM results require 3 simulations of the system, while the Monte Carlo results require 1500.

Polynomial Order	E[V_{min}]	SD[V_{min}]	Predicted Prob. of Violation
Linear	0.91300	0.00800	3.00%
Quadratic	0.91333	0.00833	8.92%
3rd Order	0.91346	0.00834	8.83%
4th Order	0.91333	0.00831	8.90%
5th Order	0.91336	0.00838	8.08%
6th Order	0.91321	0.00832	8.06%
7th Order	0.91328	0.00825	7.86%

Table 2.2: **Comparison of Polynomial Orders.** The results of the collocation study using increasingly higher-order polynomials are summarized above.

On a final note, we observe that such time-varying simulation studies are impossible to produce using variance reduction techniques. We individually calculate the expected value and standard deviation at each of 100 time sample points in order to produce such a plot. One single importance sampling distribution cannot be found which yields good results for all of these different "output variables". On the other hand, the exact same three system simulations are used to create a different polynomial for each of the time points with the probabilistic collocation method. Additional savings are obtained if we are only interested in the mean and standard deviation, since they are directly available from the coefficients of the PCM model.

2.3 Multiple Uncertain Parameters

So far, we have only focused on a single uncertain parameter in our model. However, our ultimate goal is to allow essentially all of our system parameters to be uncertain. In practice, we limit ourselves to uncertain parameters which are independent. In many cases, this assumption is justifiable. In other cases, where the uncertain system parameters are dependent, a related set of independent parameters may be substituted.

Although we are primarily interested in independent uncertain parameters and the extant explanations of the method seem to imply that we are limited to this case, we motivate the discussion from the more general standpoint of dependent uncertain parameters. A natural generalization from a single uncertain parameter to multiple uncertain parameters is outlined first and then we discuss the particular case of independent uncertain parameters.

2.3.1 Dependent Uncertain Parameters - General Case

We discuss the generalization to two uncertain parameters, since the transition from one uncertain parameter to two is the real issue. The extension to three or more uncertain parameters is obvious once the two parameter case is understood.

Our justification for single parameter PCM relies on orthogonal polynomials. When we discussed inner products on the space of polynomials, we tacitly assumed that polynomials form a vector space $\mathbb{P}$. Before we proceed, we need an appropriate vector space to describe polynomial models which are functions of two (or more) variables. A natural choice for us is a vector space which we will refer to as $\mathcal{L}^2(\mathbb{P})$, of which any member can be written in the following form:

$$\psi(\alpha, \beta) = \sum_i p_{j_i}(\alpha) p_{k_i}(\beta) \tag{2.95}$$

where the p_i are polynomials. We do not prove it here, but the space of all such functions $\mathcal{L}^2(\mathbb{P})$ forms a vector space. (The proof is nearly identical to the proof that the set of all polynomials forms a vector space). Such a definition is necessary to enable our new models to have "cross-terms", which simple linear combinations of polynomials do not possess. (In a more mathematical parlance, our space is actually the space of all 2-tensors on $\mathbb{P}$ [33], hence the notation).

As for polynomials, we need an appropriate inner product before we can discuss orthogonality. Similarly, we want to allow for a weighting function as part of our definition. This weighting function $f(\alpha, \beta)$ is the joint probability density describing the uncertainty in the dependent uncertain parameters:

$$\begin{aligned} f(\alpha, \beta) &\geq 0, \text{ for all } (\alpha, \beta) \text{ in } \mathcal{A} && (2.96) \\ \int_{\mathcal{A}} f(x, y) dx\, dy &= 1 && (2.97) \end{aligned}$$

With this weighting function, we define the inner product as follows:

$$\langle g(\alpha, \beta), h(\alpha, \beta) \rangle = \int_{\mathcal{A}} f(x, y) g(x, y) h(x, y) dx\, dy \tag{2.98}$$

We are now in a position to form a set of orthonormal basis functions. However, one problem is that a single, clear ordering of these polynomials does not exist. For the space of polynomials, the basis functions can be forced to have increasing order, which is what we do. With this new framework, each basis function has two orders associated with it, one for α and one for β. We use $\mathcal{H}'$ to denote the set of orthonormal basis functions, but we now use a system of two subscripts to denote each basis function, such that $h_{i,j}(\alpha, \beta)$ is order i in α when β is a constant and order j in β when α is a constant. By construction, each basis function $h_{i,j}$ has i roots of α and j roots of β.

However, it is not clear if such a set of orthonormal basis functions can be constructed in general. At least one set of basis functions $h_{i,j}$ does exist, but any such set of basis functions may not be orthonormal. Because of the difficulty in ordering the basis functions, we have not yet been able to generate an existence proof for the desired set $\mathcal{H}'$. Nevertheless, this framework is useful since we can answer these questions for independent uncertain parameters (i.e., $f(\alpha, \beta) = f_\alpha(\alpha) f_\beta(\beta)$), which are our primary interest anyway. With further work, the dependent parameter case might also be resolved, and we feel that our approach merits attention.

Of course, an affine (or even higher-order) representation based on dependent uncertain parameters can always be created but the issue remains at which combinations of parameter values should we perform simulations in order to obtain the coefficients. Dependent parameters are no more difficult than independent parameters once the polynomial model is created, but no obvious method to pick the simulation points used to obtain the model coefficients exists for the dependent parameter case. As the earlier discussion of Taylor series approximations for the single parameter case demonstrates, a local approximation is easy to create, but it is not clear how to select the appropriate nominal value around which to create such an approximation.

2.3.2 Independent Uncertain Parameters

Fortunately, an orthonormal basis of the desired form exists if $f(\alpha, \beta) = f_\alpha(\alpha) f_\beta(\beta)$ for some functions f_α and f_β and $\mathcal{A}$ can be expressed as $\mathcal{A}_\alpha \times \mathcal{A}_\beta$. These conditions are satisfied if and only if the two uncertain parameters, α and β, are independent. In fact, the following equation relates the orthonormal basis functions of $\mathcal{H}'$ to those of $\mathcal{H}_\alpha$ and $\mathcal{H}_\beta$:

$$h_{i,j}(\alpha,\beta) = h_i(\alpha)h_j(\beta) \tag{2.99}$$

where h_i (h_j) is the appropriate orthonormal basis function corresponding to a weighting function f_α (f_β). Some simple manipulations demonstrate this result:

$$\begin{aligned}
\langle h_{i,j}, h_{k,l}\rangle &= \int_{\mathcal{A}} f(x,y)h_{i,j}(x,y)h_{k,l}(x,y)dx\,dy &(2.100)\\
&= \int_{\mathcal{A}} f_\alpha(x)f_\beta(y)h_i(x)h_j(y)h_k(x)h_l(y)dx\,dy &(2.101)\\
&= \int_{\mathcal{A}_\alpha} f_\alpha(x)h_i(x)h_k(x)dx \int_{\mathcal{A}_\beta} f_\beta(y)h_j(y)h_l(y)dy &(2.102)\\
&= \begin{cases} 1 & \text{if } i=k \text{ and } j=l \\ 0 & \text{otherwise} \end{cases} &(2.103)
\end{aligned}$$

In this case, we again have desirable existence and location proofs for the roots of each of our basis functions.

Proceeding in a fashion directly analogous to our development for single parameter uncertainties, we build a simple model of the relationship between the uncertain input parameters and the output of interest. Denoting the actual relationship by $g(\alpha,\beta)$ and our model by $\hat{g}(\alpha,\beta)$, we have the following partial list of possible models in order of increasing complexity:

$$\begin{aligned}
\hat{g}(\alpha,\beta) &= g_0h_0(\alpha)h_0(\beta) + g_1h_1(\alpha)h_0(\beta) + g_2h_0(\alpha)h_1(\beta) &(2.104)\\
\hat{g}(\alpha,\beta) &= g_0h_0(\alpha)h_0(\beta) + g_1h_1(\alpha)h_0(\beta) + g_2h_0(\alpha)h_1(\beta) + g_3h_1(\alpha)h_1(\beta) &(2.105)\\
\hat{g}(\alpha,\beta) &= g_0h_0(\alpha)h_0(\beta) + g_1h_1(\alpha)h_0(\beta) + g_2h_0(\alpha)h_1(\beta) + g_3h_1(\alpha)h_1(\beta) + \\
&\quad g_4h_0(\alpha)h_2(\beta) + g_5h_2(\alpha)h_0(\beta) &(2.106)\\
&\vdots
\end{aligned}$$

Although we can find the roots of some basis function which is not part of the model at hand to fit the model, the choice of which basis function to use is not nearly as clear as it was for the polynomial case. In addition, the roots are not generated as pairs (α_i,β_i). The method suggested by [16] is to take all combinations of the roots of $h_i(\alpha)$ and $h_i(\beta)$ such that i is greater than the largest subscript used in the model. (Note that $h_i(\alpha)$ and $h_i(\beta)$

are not identical in general if $f_\alpha \neq f_\beta$).

However, this method leaves us with "extra" points for (2.104) and (2.106). Our approach is to simply choose some subset of the parameter pairs and use those as our collocation points. We follow the selection algorithm suggested in [16], which is to pick pairs where the density function is relatively large, though admittedly this is *ad hoc.*

2.3.3 Necessary Computation

In the case of a single uncertain parameter, the number of simulations necessary to fit the model is equal to one plus the order of the polynomial. With multiple uncertain parameters, we have an extra concern; as the number of uncertain parameters which we wish to model grows, the size and complexity of the model increases, even given a constant model "order". For example, an analogous model (i.e., where we choose all possible subscript combinations involving 0 and 1) to (2.105) with n uncertain parameters requires 2^n simulations to fit the model. The model (2.106) can be extended using a different principle (i.e., the sum of the subscripts on any term must be less than or equal to 2) and a model with n parameters requires

$$1 + 2n + \left(\begin{array}{c} n \\ 2 \end{array} \right)$$

simulations.

The latter extension principle has obvious advantages over the former when we contemplate calculations with hundreds of uncertain parameters. But, even if we "only" need $\mathcal{O}(n^2)$ simulations, hundreds of uncertain parameters still present practical problems. Our reduced-order models certainly may have hundreds of parameters, if the original model is quite large. We return to this issue in a later chapter, where we discuss ways in which to limit the number of parameters which we treat as uncertain. This problem is the motivation for trying to identify "critical" uncertain parameters in our reduced-order model.

2.4 Summary

In this chapter, we outline the basic theory underlying the probabilistic collocation method and present the practical implementation of PCM. While the method is heuristic, the principles of orthogonal polynomials and Gaussian quadrature integration provide some justification for at least the single uncertain parameter case. In the context of a single uncertain parameter, we explore a number of examples which illustrate the power and flexibility of PCM, as compared to brute force Monte Carlo, variance reduction techniques or other polynomial modeling techniques. The method can also be extended to multiple uncertain parameters, and we concentrate here on independent parameter uncertainties.

The probabilistic collocation method (both single and multiple independent uncertain parameter) as well as the theory underlying the single parameter case were previously developed by researchers in the field of global climate change. We have successively applied single parameter PCM to a small power system example as well as outlined a direction for extension to multiple dependent uncertain parameters, which we use here to better justify the previously developed independent uncertain parameter results.

Of course, the method has limitations. If the relationship between the uncertain parameters and the output of interest cannot be well modeled by a polynomial model, PCM is obviously not a good choice for the uncertainty analysis. Furthermore, even if a polynomial model is appropriate, we have yet to discover an acceptable method for definitively determining what the correct order for the polynomial model is (assuming of course that the underlying model is unknown).

A final note is that as the number of uncertain parameters grows so does the number of simulations needed to fit the PCM model. Our objective is to absolutely minimize the required number of simulations of the power system model. Therefore, we need to determine an effective way to identify which parameters should be modeled as uncertain. Otherwise, even with model reduction and PCM, uncertainty analyses will remain out of reach. This topic is the focus of a later chapter.

Chapter 3

Synchronic Modal Equivalencing - a primer

In this chapter, we present a short introduction to synchronic modal equivalencing (SME). At least a basic understanding of the relevant theory is necessary to appreciate our work and its limitations. We attempt to outline the salient features in this chapter, but the description is incomplete at best, due to space limitations. The interested reader may also benefit from a number of extant references. For a thorough description of the SME algorithm (excluding recent developments in load bus grouping algorithms), we highly recommend Ganesh Ramaswamy's thesis [12]. A less thorough and consequently more compact treatment of the same material can be found in [10] and [11]. The necessary extensions required to group load buses as well as generator buses are presented in [13] and [14]. Our presentation here borrows heavily from all of these sources. (The literature related to "slow-coherency" reduction, which is a technique with many similarities to SME, may also be of interest. Two of the standard references are [36] and [37]).

3.1 Basic Concepts and Components

SME is a heuristic approach to dynamic equivalencing of power systems which uses the modal structure of a linearization of the system to group the system nodes; one (or a few) of these groups are selected to form a study area, and all other groups are replaced by significantly smaller equivalents. This one sentence is sufficient to help us understand some of the strengths and weaknesses of the SME algorithm, which we explore by way of introduction to the main features of the algorithm.

One important aspect to note first is that SME is a heuristic algorithm. While many of the concepts underlying the details of the algorithm are rigorously developed and the approach has been guided by good engineering intuition, the algorithm itself has not been shown to be "optimal" in any sense and a real power system model is unlikely to have the exact structures under which we could guarantee that the algorithm would work. However, many power system models do exhibit structure which approximately satisfies the requirements for the underlying theory, so the algorithm is certainly not without basis. In addition, the original and equivalenced models are both nonlinear, but the theorems we have are based on a linear system analysis, which is why we must linearize the system to do the analysis. Nevertheless, very good equivalents (in terms of reproducing the behavior of the original model) have been obtained for nonlinear models, although the theoretical development in this area has lagged significantly behind the practice.

Despite these limitations, SME has features which make it appealing, generally and in particular for our work on uncertainty analyses. Though not theoretically as well grounded as one might wish, the algorithm is easy to extend and use on nonlinear models, and the results have been promising, as noted above. Realistic power system models are nonlinear, so the ability to accommodate nonlinear models is key. Another important property of the SME framework is that the resulting equivalent can be easily implemented using commercial power system simulation software. The ability to implement the equivalent easily is directly a result of the algorithm's insistence on retaining physically meaningful variables throughout the process. As we proceed to explain the details of the algorithm, it is important to keep this fact in mind. Techniques such as balanced realizations require a change of variables which creates variables with no physical interpretation; the resulting equivalent may not be realizable as a power system, using standard power system component models. A purely mathematical model without physical basis would preclude an extension to nonlinear models and would limit its acceptance within the power system community, where physically meaningful variables and the intuition we derive from them are valued.

We further discuss the merits of SME and its limitations as we encounter these issues in the course of the discussion, so as to not burden the reader with too much discussion of an algorithm that has not been properly explained in this thesis yet. But, before we work through the details of the various components of the algorithm, a brief overview of the major components of the algorithm may be instructive.

First, a linearization of the original nonlinear power system model is calculated. Typically, this linearized system is in the form of a set of linear differential algebraic equations (DAE), where the differential variables correspond to generators and those loads that are modeled dynamically and the algebraic variables are typically voltages and currents which describe the interconnections in the system. A simplified model is extracted from these equations, and the eigenvalues and eigenvectors of this model are calculated. A subset of these eigenvalues are selected (roughly speaking, the eigenvalues are selected based on how "extensive" the corresponding mode is; we explain these concepts more fully later), referred to as the "chord", and the right and left eigenvectors corresponding to these modes are used to select a set of "basis generators". Subsequently, the basis generators are used to assign every node in the system to a group, so that every node is in exactly one group and every group has exactly one basis generator.

At this point, the user decides which group (or groups) to select as the "study area". In the equivalent, the details of all other groups are greatly reduced, but the full detail of the study area is retained. Normally, the study area contains the event of interest, as well as the nodes nearest to that event. The final step is to calculate the equivalent, based on the study area selected; though the partitioning and equivalencing are performed on the linearized model, the final equivalenced model is nonlinear because the full nonlinear detail of the study area is retained. We believe that such equivalents are reasonable, based on the assumption that a given event in the system is not highly sensitive to the details of every component in the system; while the effect of "distant" generators, loads and/or transmission lines must certainly be taken into account, detailed models of all of these components are not necessary for accurate simulation results.

3.2 Linearization and Extraction of Simplified Model

A realistic power system model is not only nonlinear, but it may also contain as many as ten or fifteen state variables for every generator. Since SME is based on a modal analysis of the system, we require a linearization of the system, which may be generated by the power system simulation package used for the simulations (in our case EUROSTAG [38] [39]). While the generation of such a linearization is straightforward, extracting the simplified model requires some care. With several state variables per generator, the original linearized model is unwieldy for several reasons. First and foremost, the amount of time needed to compute a complete eigenanalysis for a large power system is prohibitive if we retain the full detail. Additionally, the grouping algorithm was formulated under the assumption that each node in the system is represented by exactly one variable in the system matrix.

3.2.1 Ideal Simplified Model for Generator Partitioning

Without reiterating the justification from the references here, we assert that a linearized, undamped "swing-equation" model (also known as a "classical" model) suffices to partition the system into groups. We first derive the desired form for the linearization and then briefly describe how we attempt to obtain this form. Suppressing the load buses for the moment (in other words, we have algebraically eliminated these equations), the nonlinear swing-equation model is given by:

$$\mathbf{M}\ddot{\delta} = P_m - P \tag{3.1}$$

where $\mathbf{M}$ is a diagonal matrix of generator inertias, P_m is a vector of the mechanical input power to each of the generators, P is a vector of the electrical output power of each of the generators, and δ is a vector comprising the generators' internal angle deviations from a synchronously rotating reference frame. These equations could analogously represent a set of masses interconnected by springs, where $\mathbf{M}$ represents the masses, δ represents their displacements and the P's are force terms. We assume that the mechanical input power to each generator is constant and concentrate on the electrical equations (analogously, each mass has a constant force acting on it and we concentrate on the forces exerted by the springs).

Assuming that all of the loads in the system are can be represented as impedances, we can denote the admittance matrix of the system (after the load buses are eliminated) by $\mathbf{Y} = \mathbf{G} + j\mathbf{B}$. We follow the following convention for reduced admittance matrices: each off-diagonal term is the negative of the admittance connecting the corresponding two nodes and each diagonal term is the negative of the sum of the off-diagonal terms on its row plus any admittance to ground. We can now write equations for the real and reactive power balance at node i in an n generator system:

$$P_i + jQ_i = V_i I_i^* \tag{3.2}$$

where Q_i is the reactive power output of the generator, V_i is the complex voltage at node i and I_i is the complex current flowing out of the generator at node i. We can express these equations entirely in terms of voltages using the following equation for the current:

$$I_i = \sum_{j=1}^{n} Y_{ij} V_j \tag{3.3}$$

Using (3.2) and (3.3), we obtain:

$$\begin{aligned} P_i &= |V_i|^2 G_{ii} + \sum_{j=1, j\neq i}^{n} |V_i||V_j|(B_{ij}\sin(\theta_i - \theta_j) + G_{ij}\cos(\theta_i - \theta_j)) && (3.4) \\ Q_i &= -|V_i|^2 B_{ii} + \sum_{j=1, j\neq i}^{n} |V_i||V_j|(G_{ij}\sin(\theta_i - \theta_j) - B_{ij}\cos(\theta_i - \theta_j)) && (3.5) \end{aligned}$$

where the θ denote the angles of the voltages at the buses to which the generators are connected.

At this point, an approximate decoupling between ($\mathbf{P}$,δ) and ($\mathbf{Q}$,$|V|$) is often assumed (for example in [13]). However, we do not do this in practice when calculating the equations we use for grouping, so we do not assume such a decoupling here; we only mention the possibility to avoid confusion for the reader who is familiar with the literature. We are now prepared to calculate the linearization and eliminate the reactive power equations. Linearizing with respect to the internal angles and the voltage magnitudes, we obtain the following equations (assuming that the mechanical input power to the generators is constant and eliminating the generator terminal buses):

$$\begin{aligned} \mathbf{M}\Delta\ddot{\delta} &= \mathbf{C_1}\Delta\delta + \mathbf{C_2}\Delta|V| && (3.6) \\ 0 &= \mathbf{C_3}\Delta\delta + \mathbf{C_4}\Delta|V| && (3.7) \end{aligned}$$

where the constant matrices are defined as follows:

$$(C_1)_{ij} = \left\{ \begin{array}{ll} |\bar{V}_i||\bar{V}_j|(B_{ij}\cos(\bar{\delta}_i - \bar{\delta}_j) - G_{ij}\sin(\bar{\delta}_i - \bar{\delta}_j)) & j \neq i \\ -\sum_{k=1,k\neq i}^{n}(C_1)_{ik} & j = i \end{array} \right. \tag{3.8}$$

$$(C_2)_{ij} = \left\{ \begin{array}{ll} -|\bar{V}_i|(B_{ij}\sin(\bar{\delta}_i - \bar{\delta}_j) + G_{ij}\cos(\bar{\delta}_i - \bar{\delta}_j)) & j \neq i \\ -2|\bar{V}_i|G_{ii} - \sum_{k=1,k\neq i}|\bar{V}_k|(B_{ik}\sin(\bar{\delta}_i - \bar{\delta}_k) + G_{ik}\cos(\bar{\delta}_i - \bar{\delta}_k)) & j = i \end{array} \right. \tag{3.9}$$

$$(C_3)_{ij} = \left\{ \begin{array}{ll} -|\bar{V}_i||\bar{V}_j|(G_{ij}\cos(\bar{\delta}_i - \bar{\delta}_j) + B_{ij}\sin(\bar{\delta}_i - \bar{\delta}_j)) & j \neq i \\ -\sum_{k=1,k\neq i}^{n}(C_3)_{ik} & j = i \end{array} \right. \tag{3.10}$$

$$(C_4)_{ij} = \left\{ \begin{array}{ll} |\bar{V}_i|(G_{ij}\sin(\bar{\delta}_i - \bar{\delta}_j) - B_{ij}\cos(\bar{\delta}_i - \bar{\delta}_j)) & j \neq i \\ -2|\bar{V}_i|B_{ii} + \sum_{k=1,k\neq i}|\bar{V}_k|(G_{ik}\sin(\bar{\delta}_i - \bar{\delta}_k) - B_{ik}\cos(\bar{\delta}_i - \bar{\delta}_k)) & j = i \end{array} \right. \tag{3.11}$$

where we use a bar notation to denote nominal values.

We are now in a position to create a model with one variable per generator, namely its internal angle. We define the matrix **F** and the system we use for grouping as follows:

$$\mathbf{F} = \mathbf{C_1} - \mathbf{C_2}\mathbf{C_4}^{-1}\mathbf{C_3} \tag{3.12}$$

$$\Rightarrow \mathbf{M}\Delta\ddot{\delta} = \mathbf{F}\Delta\delta \tag{3.13}$$

We use the matrix $\mathbf{M}^{-1}\mathbf{F}$ (often referred to as the "core" matrix) to select our chord and the basis generators as well as to partition the remaining generators. Although this equation is not in state-space form, the natural frequencies of the system are closely related to the eigenvalues of the core matrix and the eigenvectors of the core matrix describe the motion associated with these natural frequencies. The matrix **F** has particular structure. First, it has a zero eigenvalue with a corresponding right eigenvector of all ones, which is easily verified by noting that both $\mathbf{C_1}$ and $\mathbf{C_3}$ have the property that every row in the matrix sums to zero; consequently, $\mathbf{M}^{-1}\mathbf{F}$ also has these properties. Unfortunately, we cannot make any other statements without making further assumptions about **F**.

One nice property to have would be symmetry, and we have observed that many of our core matrices are approximately symmetric when left multiplied by **M**. If **F** were symmetric, its left eigenvector corresponding to the zero eigenvalue would also consist of all ones; consequently, the core matrix would have a corresponding left eigenvector comprising the generator inertias. Again, these properties have been observed on actual core matrices. In fact under reasonable limiting assumptions, **F** is symmetric.

The first and most limiting assumption is that **G** be identically zero. While this is certainly not unreasonable for the transmission lines, where the inductance is substantially larger than the conductance, this assumption is difficult to justify for the loads. A power system in which no real power is consumed would not be a terribly useful power system; why would we want to distribute power if nobody was using it? The load conductance terms appear not only on the diagonal of **G** but also in off-diagonal terms because the nodes without generators have been algebraically eliminated using a standard Ward-type reduction [40]. Nonetheless, the effect of the conductance terms do not appear to dominate in practical power system models. The resulting (simplified) $\mathbf{C_i}$ matrices are as follows:

$$(C_1)_{ij} = \begin{cases} |\bar{V}_i||\bar{V}_j|B_{ij}\cos(\bar{\delta}_i - \bar{\delta}_j) & j \neq i \\ -\sum_{k=1,k\neq i}^{n}(C_1)_{ik} & j = i \end{cases} \tag{3.14}$$

$$(C_2)_{ij} = \begin{cases} -|\bar{V}_i|B_{ij}\sin(\bar{\delta}_i - \bar{\delta}_j) & j \neq i \\ -\sum_{k=1,k\neq i}|\bar{V}_k|B_{ik}\sin(\bar{\delta}_i - \bar{\delta}_k) & j = i \end{cases} \tag{3.15}$$

$$(C_3)_{ij} = \begin{cases} -|\bar{V}_i||\bar{V}_j|B_{ij}\sin(\bar{\delta}_i - \bar{\delta}_j) & j \neq i \\ -\sum_{k=1,k\neq i}^{n}(C_3)_{ik} & j = i \end{cases} \tag{3.16}$$

$$(C_4)_{ij} = \begin{cases} -|\bar{V}_i|B_{ij}\cos(\bar{\delta}_i - \bar{\delta}_j) & j \neq i \\ -2|\bar{V}_i|B_{ii} - \sum_{k=1,k\neq i}|\bar{V}_k|B_{ik}\cos(\bar{\delta}_i - \bar{\delta}_k) & j = i \end{cases} \tag{3.17}$$

We note that $\mathbf{C_1}$ is now symmetric. If $\mathbf{C_4}$ were symmetric and $\mathbf{C_2} = c\mathbf{C_3}^T$ (where c is any scalar), then **F** would also be symmetric. However, we require a further, albeit reasonable, assumption to guarantee these properties. All of the voltages are measured in per unit, so they all have approximately unit magnitude. Under the further assumption that all of the voltage magnitudes are equal, we note that $\mathbf{C_4}$ is symmetric and obtain the following form for $\mathbf{C_2}$ and $\mathbf{C_3}$:

$$(C_3)_{ij} = |\bar{V}_1|(C_2)_{ji} = \begin{cases} -|\bar{V}_1|^2 B_{ij}\sin(\bar{\delta}_i - \bar{\delta}_j) & j \neq i \\ -\sum_{k=1,k\neq i}^{n}(C_3)_{ik} & j = i \end{cases} \tag{3.18}$$

3.2.2 Simplified Model Extraction and Load Partitioning

Now that we understand the model used to do the generator partitioning, the generalization allowing us to also partition load nodes is relatively straightforward. However, the details of this simplified model are somewhat intertwined with the particular way in which we extract the model from the EUROSTAG linearization, so we present both topics in this subsection.

We seek a system of equations which is analogous to (3.13), but with one important difference, the nodes that do not have a generator attached to them have zero inertia. The derivation of this model is identical to that above, except we start with an admittance matrix with the load nodes still present; we do not repeat those manipulations. We present the resulting set of equations here (where θ denotes the voltage angles only at the load nodes):

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\ddot{\delta} \\ \Delta\ddot{\theta} \end{bmatrix} = \begin{bmatrix} \mathbf{F_1} & \mathbf{F_2} \\ \mathbf{F_3} & \mathbf{F_4} \end{bmatrix} \begin{bmatrix} \Delta\delta \\ \Delta\theta \end{bmatrix} \tag{3.19}$$

where we also make the following definitions for simplicity:

$$\mathbf{M_{ext}} = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.20}$$

$$\mathbf{F_{ext}} = \begin{bmatrix} \mathbf{F_1} & \mathbf{F_2} \\ \mathbf{F_3} & \mathbf{F_4} \end{bmatrix} \tag{3.21}$$

The similarity between these equations and those of the core matrix can be even more readily recognized when we note the following relationship between $\mathbf{F}$ and the submatrices of $\mathbf{F_{ext}}$:

$$\mathbf{F} = \mathbf{F_1} - \mathbf{F_2}\mathbf{F_4}^{-1}\mathbf{F_3} \tag{3.22}$$

In practice, this is in fact exactly how we compute $\mathbf{F}$; we first compute $\mathbf{F_{ext}}$ and then further reduce the system to obtain $\mathbf{F}$.

Equation (3.19) does differ from (3.13) in one important respect: (3.19) is not uniquely defined. The algebraic equations can be left multiplied by any arbitrary invertible matrix and the resulting set of equations is still valid. In fact, if we algebraically eliminate the algebraic equations from any such set of equations, the same core matrix results. We will be performing a generalized eigenanalysis on these equations (see [41] for an explanation of generalized eigenvectors), and while the resulting eigenvalues and right generalized eigenvectors are not affected by such a left multiplication, the left generalized eigenvectors are affected. As we will show later, both the left and right generalized eigenvectors play a role in partitioning the load nodes, so a unique set of equations must be determined.

Before determining the appropriate form of the equations, we verify that the generalized eigenvectors are affected by such a left multiplication. We use $\mathbf{X}$ to represent an arbitrary invertible matrix. The right eigenvector problem for a finite eigenvalue λ can be written as follows:

$$\left\{ \begin{array}{rcl} \lambda \mathbf{M} v_\delta & = & \mathbf{F_1} v_\delta + \mathbf{F_2} v_\theta \\ 0 & = & \mathbf{X F_3} v_\delta + \mathbf{X F_4} v_\theta \end{array} \right. \tag{3.23}$$

$$\Rightarrow \left\{ \begin{array}{rcl} \lambda \mathbf{M} v_\delta & = & \mathbf{F} v_\delta \\ v_\theta & = & -(\mathbf{F_4}^{-1} \mathbf{F_3}) v_\delta \end{array} \right. \tag{3.24}$$

So, the $\mathbf{X}$ cancels out and does not affect the computation. However, this is not the case for the left eigenvector problem although the computations are similar, as we see below:

$$\left\{ \begin{array}{rcl} \lambda w_\delta^{\mathrm{T}} \mathbf{M} & = & w_\delta^{\mathrm{T}} \mathbf{F_1} + w_\theta^{\mathrm{T}} \mathbf{X F_3} \\ 0 & = & w_\delta^{\mathrm{T}} \mathbf{F_2} + w_\theta^{\mathrm{T}} \mathbf{X F_4} \end{array} \right. \tag{3.25}$$

$$\Rightarrow \left\{ \begin{array}{rcl} \lambda w_\delta^{\mathrm{T}} \mathbf{M} & = & w_\delta^{\mathrm{T}} \mathbf{F} \\ w_\theta^{\mathrm{T}} & = & -w_\delta^{\mathrm{T}} (\mathbf{F_2 F_4}^{-1} \mathbf{X}^{-1}) \end{array} \right. \tag{3.26}$$

It should be apparent that $\mathbf{X}$ cannot "cancel", and, therefore, we must determine the appropriate form for the equations.

As a starting point, we explore the extraction of these equations from the original linearization and see in what form these equations are when the load partitioning equations are initially created. Since we desire an approximation to a linearized, undamped swing-equation model, the first step is to ignore all of the entries in the matrix corresponding to differential variables other than those relating the derivative of speed ($\dot{\omega} = \ddot{\delta}$) to internal angle. In other words, we completely ignore all other differential variables and set the damping terms to zero (damping terms relate the derivative of speed to the speed variable). The algebraic constraints are output by EUROSTAG in the form of current balance equations with real and imaginary components of the voltages at the nodes as the algebraic variables. In summary, we have a set of equations in the form:

$$\begin{bmatrix} \Delta \ddot{\delta} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{M}^{-1}\mathbf{F_1} & \mathbf{M}^{-1}\mathbf{F_5} & \mathbf{M}^{-1}\mathbf{F_6} \\ \mathbf{F_7} & \mathbf{Y_1} & \mathbf{Y_2} \\ \mathbf{F_8} & \mathbf{Y_3} & \mathbf{Y_4} \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta \Re\{V\} \\ \Delta \Im\{V\} \end{bmatrix} \tag{3.27}$$

where $\begin{bmatrix} \mathbf{Y_1} & \mathbf{Y_2} \\ \mathbf{Y_3} & \mathbf{Y_4} \end{bmatrix}$ forms a real matrix realization of the complex admittance matrix describing the system.

We would like to convert the real and imaginary voltages to voltage magnitudes and angles and then algebraically eliminate the voltage angles at nodes with generators attached to them. Then, if we were to follow our earlier development, we would also algebraically eliminate the voltage magnitudes using the reactive power balance equations, but we don't have any reactive power balance equations here! Therein lies the problem and our solution to the puzzle of which of the possible realizations of the algebraic equations to use.

One of the first papers related to load bus partitioning in an SME or slow-coherency framework using generalized eigenvectors used algebraic equations that represented real current balance equations [42]. However, in [14] and [13], the algebraic equations are presented as real power balance equations. We feel that a real power balance is more consistent, given that the differential equations also represent a real power balance. The necessary transformation from current balance to power balance is easy to derive.

The algebraic equations of (3.27) are really just expressing Kirchhoff's current law for the real and imaginary part of the current at each of the system nodes in the linearization. Before the linearization, we can express this current balance as follows:

$$\sum_{j=1}^{n} I_{ij} = 0 \tag{3.28}$$

where I_{ij} is the complex current into bus i from node j. We obtain a similar relationship for power by applying our observation from (3.2) here:

$$V_i \sum_{j=1}^{n} I_{ij}^{*} = 0 \tag{3.29}$$

However, now we have a summation with units of complex power instead of complex current, as desired. We only need to apply our linearization to obtain the desired **X**:

$$\Delta V_i \underbrace{\sum_{j=1}^{n} \bar{I}^*_{ij}}_{=0} + \bar{V}_i \sum_{j=1}^{n} \Delta I^*_{ij} = 0 \tag{3.30}$$

$$\bar{V}_i \sum_{j=1}^{n} \Delta I^*_{ij} = 0 \tag{3.31}$$

Our original equations are in the form of real and imaginary current balances and we need to also write our new complex equations in terms of real and reactive power balances:

$$\Im\{\bar{V}_i\} \sum_{j=1,j\neq i}^{n} \Im\{\Delta I_{ij}\} + \Re\{\bar{V}_i\} \sum_{j=1,j\neq i}^{n} \Re\{\Delta I_{ij}\} = 0 \tag{3.32}$$

$$\Im\{\bar{V}_i\} \sum_{j=1,j\neq i}^{n} \Re\{\Delta I_{ij}\} - \Re\{\bar{V}_i\} \sum_{j=1,j\neq i}^{n} \Im\{\Delta I_{ij}\} = 0 \tag{3.33}$$

In summary, the transformation applied to any particular node i can be represented by the following matrix multiplication:

$$\begin{bmatrix} |\bar{V}_i| \cos \bar{\delta}_i & |\bar{V}_i| \sin \bar{\delta}_i \\ |\bar{V}_i| \sin \bar{\delta}_i & -|\bar{V}_i| \cos \bar{\delta}_i \end{bmatrix} \begin{bmatrix} \sum_{j=1}^{n} \Re\{\Delta I_{ij}\} \\ \sum_{j=1}^{n} \Im\{\Delta I_{ij}\} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.34}$$

After applying this transformation, the algebraic equations are in terms of real and reactive power balances, and the reactive power balance equations can be used to eliminate the voltage magnitudes, as desired. With linearized system equations output from EUROSTAG, we found that the resulting simplified differential algebraic equations (DAE) had properties consistent with those presented for the core matrix. (In other words, the zero eigenvalue properties were apparent and the approximate symmetry was also present, with one caveat. In order to observe approximate symmetry, we needed to scale the algebraic equations by $-\frac{5}{6}f$, where f is the system frequency in Hertz. We believe that this discrepancy is the result of a per unit normalization; however, this constant does not overly concern us since it does not affect the chord selection or partitioning algorithms). Assuming that (3.19) is formed in a way consistent with our discussion here, we refer to the following matrix as the extended core matrix:

$$\begin{bmatrix} \mathbf{M}^{-1} & \mathbf{0} \\ \mathbf{0} & -\frac{5}{6}f\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{F_1} & \mathbf{F_2} \\ \mathbf{F_3} & \mathbf{F_4} \end{bmatrix} \tag{3.35}$$

3.3 Chord and Basis Generator Selection

The core matrix is used to select both the chord for the system and the basis generators. In both cases, the selection algorithms are the subject of current research. For this reason and also because our research only peripherally involves the details of these algorithms, we do not present a thorough analysis of them. Nevertheless, understanding the process of chord and basis generator selection is important for obtaining a complete picture of how SME works, so we do describe the objectives of each.

3.3.1 Chord Selection

Of all the details in the SME algorithm, the selection of the chord can have the largest overall impact on the form of the reduced model. If we step back for a moment and examine the big picture, the reasons for this should become clear.

The basic premise underlying SME is that the entire detailed model description is not necessary to obtain reasonable results for a particular simulation or set of simulations. Of course, it is not possible to reduce a power system model and retain complete fidelity for every possible simulation, unless the original system model contained some unnecessary redundancy. One can either create a reduced model which will reproduce as accurately as possible every behavior under some optimality criterion or create a reduced model which will reproduce as accurately as possible a particular subset of the possible system behaviors. SME is the latter type of model reduction and a key part of defining what system behavior to retain is the selection of the chord.

Since SME is a modal technique, it is natural that the portion of the system behavior to "retain" and the portion for which we are willing to sacrifice some fidelity are defined in terms of modes of the system. The underlying assumption is that the system modes can be split into "extensive" and "local" modes. If we are primarily interested in a particular event or set of events and the system response to them in some portion of the model, we need to retain the modes local to that part of the system as well as the extensive modes that extend throughout the system. We try to select the chord such that it contains the most important of the extensive modes of the system.

One good example of an extensive mode is the zero-mode of the system. Every generator in the system substantially participates in this mode, as demonstrated by the equal entries in the right eigenvector associated with this mode. Unfortunately, we have not yet settled on a way to identify such an extensive mode which gives "good" results in every situation. One method, which is probably the most justifiable technique, is described in [12]. We describe an alternate method here because it allows us to discuss "participation factors", which are important for other parts of the SME algorithm and our work in this thesis.

As a starting point we note that we justified our statement that the zero-mode is extensive by referring to the structure of the associated right eigenvector. While using the right eigenvector is certainly reasonable, some problems with this approach do exist. Most importantly, the right eigenvector is unit dependent; each entry of the right eigenvector has the units of the corresponding variable in the system. As a consequence, the right eigenvector shape is not scale-invariant. This limitation is particularly troublesome for power systems because the variables are often normalized to take advantage of simplifications that arise from analyzing a system in "per-unit". If a different base is used for the normalization, the right eigenvector shape may change. Participation factors are one reasonable, scale-invariant alternative to right eigenvector entries.

To define participation factors and bring out their connections to eigenvectors, let us examine a simple state-space model [43] [44]:

$$\dot{x} = \mathbf{A}x \tag{3.36}$$

We define $\mathbf{V}$ and $\mathbf{W}^{\mathrm{T}}$ to be the right and left eigenvector matrices, such that the columns of $\mathbf{V}$ and $\mathbf{W}$ are the right and left eigenvectors of the system matrix $\mathbf{A}$:

$$\mathbf{V} = \begin{bmatrix} | & & | \\ v_1 & \cdots & v_n \\ | & & | \end{bmatrix} \tag{3.37}$$

$$\mathbf{W}^{\mathrm{T}} = \begin{bmatrix} - & w_1^{\mathrm{T}} & - \\ & \vdots & \\ - & w_n^{\mathrm{T}} & - \end{bmatrix} \tag{3.38}$$

where we scale the eigenvectors such that $\mathbf{W}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$. The response of this system to initial condition $x(0)$ is easily written in terms of eigenvectors and eigenvalues:

$$x(t) = \sum_{i=1}^{n} w_i^{\mathrm{T}} x(0) v_i e^{\lambda_i t} \tag{3.39}$$

If we further limit ourselves by restricting x(0) to be the kth standard unit vector (all zeroes except entry k), this expression reduces to a description of the response of the system to an initial disturbance of variable k:

$$x(t) = \sum_{i=1}^{n} (w_i^{\mathrm{T}})_k (v_i)_k e^{\lambda_i t} \tag{3.40}$$

We define the participation of the ith mode in the kth variable, p_{ki} as those constant terms which appear in (3.40), namely:

$$p_{ki} = (w_i^{\mathrm{T}})_k (v_i)_k = W_{ki} V_{ki} \tag{3.41}$$

With our normalization of the eigenvectors, the participation factors corresponding to a particular mode or corresponding to a particular variable sum to unity:

$$\sum_{i=1}^{n} p_{ki} = \sum_{k=1}^{n} p_{ki} = 1 \tag{3.42}$$

In addition, for a symmetric matrix $\mathbf{A}$ the participation factors are just the squares of the entries in the right eigenvector matrix. In that case, the unity sum is a very strict normalization result. However, in general, the participation factors may be positive, negative or even complex, which limits the practical significance of this result. Nevertheless, as promised, the participation factors have no units and are scale-invariant.

So, if the magnitude of p_{ki} is a good measure of the strength of interaction between variable k and mode i, we might look at how much the magnitude of p_{ki} varies with k for a given mode i to determine whether mode i is extensive. If the participation factors are evenly distributed over all of the variables, we could reasonably conclude that that mode is extensive. We name this approach the "variance of participation factors" approach. We take the magnitude of the participation factors for every mode, renormalize them since the magnitudes do not necessarily sum to unity and then calculate the sample variance of the result. If the sample variance is low, we conclude that the mode is extensive, since that designates a mode with roughly equal participation from every variable (generator).

It is important to note that the system matrix $\mathbf{A}$ in our application is the previously derived "core" matrix $\mathbf{F}$ and not the full system matrix. The size of $\mathbf{F}$ does not make a full eigenanalysis prohibitively expensive.

3.3.2 Basis Generator Selection

Once we have selected a chord, we choose a set of basis generators using only the dynamics described by those modes. The reasoning is as follows. The modes of the system can be split into roughly two groups: those that are extensive and those that are local. When we create the equivalent, we only want to retain those local modes that are local to our study area; the reduction is ideally a result of ignoring all of the other local modes, which constitute a major percentage of the simulation burden but only have a minimal impact on studies wholly contained within the study area. We also want to capture the effect of the extensive modes, since they extend throughout the system and have an impact on almost any simulation performed. The chord ideally contains the modes that comprise these extensive dynamics and we attempt to select basis generators such that the response of any other generator within the system can be described as a linear combination of the responses of the basis generators, when we only consider the extensive modes, hence, the term *basis* generator.

If a generator's response is a scalar multiple of some other generator's response in the chord ν, we say that those two generators exhibit *synchrony*. Synchrony is a natural generalization of *coherency*, which requires that the responses be identical. Synchrony can be further generalized to encompass the case where one generator's response is a linear combination of several other generator's responses. So, when we just look at the dynamics of the chord ν, we desire basis generators such that every non-basis generator exhibits synchrony with respect to the basis generators at least approximately.

Reordering our right eigenvector matrix such that the modes in the chord appear first, we can then easily separate out those modes:

$$\mathbf{V} = \left[\begin{array}{c|c} \mathbf{V}_{\nu} & \mathbf{V}_{\nu^C} \end{array} \right] \tag{3.43}$$

where the subscript ν denotes eigenvectors associated with eigenvalues in the chord and the subscript ν^C denotes the complement of this set (i.e., the other eigenvectors). Let us assume that the first two generators are the basis generators and that there are two modes in our chord. Then, if there was exact multi-dimensional synchrony in the system, the following relationship would hold:

$$\left[\begin{array}{ccc} V_{i,1} & \cdots & V_{i,p} \end{array} \right] = k_1 \left[\begin{array}{ccc} V_{1,1} & \cdots & V_{1,p} \end{array} \right] + k_2 \left[\begin{array}{ccc} V_{2,1} & \cdots & V_{2,p} \end{array} \right], \text{ for all } i \tag{3.44}$$

Naturally, most systems do not exactly have such structure. We seek basis generators, however, that help bring out as much of this type of structure as possible. Without presenting the details of the algorithm (which are not necessary to understand the present thesis), we would like to present two of the principles that the basis generator selection algorithm uses. First, the algorithm seeks basis generators whose corresponding rows in $\mathbf{V}_{\nu}$ are as mutually orthogonal as possible. The reason for this should be fairly clear. If we wish to represent every other row as a linear combination of these rows, having an orthogonal basis allows us to "cover" as large a subspace as possible. Second, the basis generators selected by the algorithm have large participation in the chord. This goal is reasonable from a physical standpoint. If we seek to equivalence an entire section of the system and essentially replace it with one (basis) generator, we do not want that (basis) generator to be highly coupled into the local dynamics nor do we want it to have minimal participation in the extensive modes of the system, which are what we are trying to retain.

3.4 Partitioning the System

In order to partition the entire system (generators and load nodes), we switch from using the core matrix to the extended core matrix at this point. Consequently, while we are still concerned with the modal structure of the system, we also need to use generalized eigenanalysis, as opposed to the more familiar regular eigenanalysis. Despite these changes, the reasoning underlying the partitioning algorithm is the same as that for the basis generator selection. Namely, we seek to group each load and generator with one of the basis generators such that the behavior of the generator (in terms of internal angle) or load node (in terms of voltage angle at the bus) is well-represented by some scalar multiple of the behavior of that basis generator.

As with the basis generator selection, we are not as concerned with the particulars of the algorithm as with the goals of the algorithm. A few points are however important to note.

First, the partitioning algorithm essentially seeks to maximize the one-dimensional synchrony within each group and minimize such synchrony between groups. But, as we shall see later, the equivalencing is based on multi-dimensional synchrony. Said differently, the partitioning algorithm looks for the single basis generator which most closely matches a particular generator or load node and groups the generator or load node with that basis generator, but the equivalent will be based on representing each non-basis generator and load node as a linear combination of *all* of the basis generators outside of the study area. This apparent disconnect between the partitioning and the equivalencing has a practical explanation. When the partitioning is being formed, the study area has not yet been determined. So, it is not clear at that point which of the basis generators will be external to the study area and which one(s) will be inside the study area.

No other partitioning schemes have been seriously explored, but we can imagine at least two alternate approaches to partitioning based on our selected chord and basis generators. The first relies on *a priori* knowledge of the approximate outline of the desired study area, since we know which event(s) we would like to simulate. If we know which basis generators are inside the study area and which are outside, an alternate idea would be to formulate an algorithm which seeks to place nodes outside the study area which are well-represented using multi-dimensional synchrony with respect to the external basis generators. In practice under the current algorithm, once the study area is selected all other external partitions are ignored. The only division which is used in any way is that between nodes in the study area, those external to the study area, and the basis generators. A similar idea was briefly mentioned and dismissed in [12], but we feel that further exploration may be fruitful.

Another radically different idea would be to use the local modes to perform the partitioning. As far as we know, this idea has never been considered, and it may be that it is not feasible or just not a good idea. However, we note that the equivalent model seeks to retain two general classes of dynamics: the modes local to the study area and the extensive modes of the system. So far all of our equivalencing procedures have concentrated solely on the extensive modes, with the belief that by retaining the study area in detail the local modes would only be minimally impacted. However, it is quite possible that a particular local mode dominates for the desired event and has a high degree of participation with another node or generator that is grouped outside the study area. There is nothing whatsoever to preclude this from happening. Perhaps, fruitful partitions could be obtained by exploring the extent of the various local modes and trying to divide the system based on those observations.

Second, the partitioning algorithm presupposes that the desired number of partitions is already known. Obviously, if we partition the system into four groups instead of five groups, the groups themselves will have to be different. Some partitionings may be "better" than others, where "good" means a small study area while sacrificing as little accuracy as possible. However, we do not yet have a well-defined set of measures to determine whether any particular partitioning is a good partitioning without doing extensive simulation testing. This area is the subject of active research.

On a third and final note, we mention the connectivity of the groups. For reasons which will become apparent when we discuss the external network reduction step of equivalencing, we would like to have each group be well-connected. By that we mean that one could travel along lines in the system from any node in a group to any other node in the same group while only traversing nodes within that group. As this connectivity worsens, the number of new lines and current injectors in the equivalent may increase, which leads to less of a reduction.

3.5 Study Area Selection and System Reduction

After the system has been analyzed and partitioned, we can begin to actually carry out the reduction, based on the study area which the user has selected. The study area selection is largely determined by the simulations to be done. If a line is faulted or a generator is tripped as part of the simulation, they must be contained within the study area. Otherwise, that generator may be equivalenced or that line may be removed, which creates obvious difficulties for simulation. In addition, if the generators and/or lines to be studied are near the border of two groups, both groups may need to be included in the study area for accurate results.

At this point, the variables in the system can be separated into several classes. The first division is between differential variables and algebraic variables, which are denoted by x and u, respectively (we algebraically eliminate any algebraic variable which is not a voltage). Another division is between those variables which are in the study area, those variables associated with basis generators, and all other variables external to the study area. A last division is between those external non-basis generator variables associated with nodes which are close to the study area (connected to the study through exactly one transmission line or transformer) and those which are further away.

We use a system of subscripts to denote the various divisions (other than between algebraic and differential). To illustrate, we refer to Figure 3.1.

ri - **relevant variables in the study area** All variables associated with study area buses, such as buses 1, 2, 8 and 9.

ro - **relevant variables outside the study area** All variables associated with basis generator buses outside the study area, such as buses 3 and 4.

r - **all of the above relevant variables**

zc - **less-relevant variables close to the study area** All variables associated with buses which are not in the study area or attached to a basis generator but are directly connected to a study area bus, such as bus 10.

zf - **less-relevant variables far from the study area** All variables associated with buses

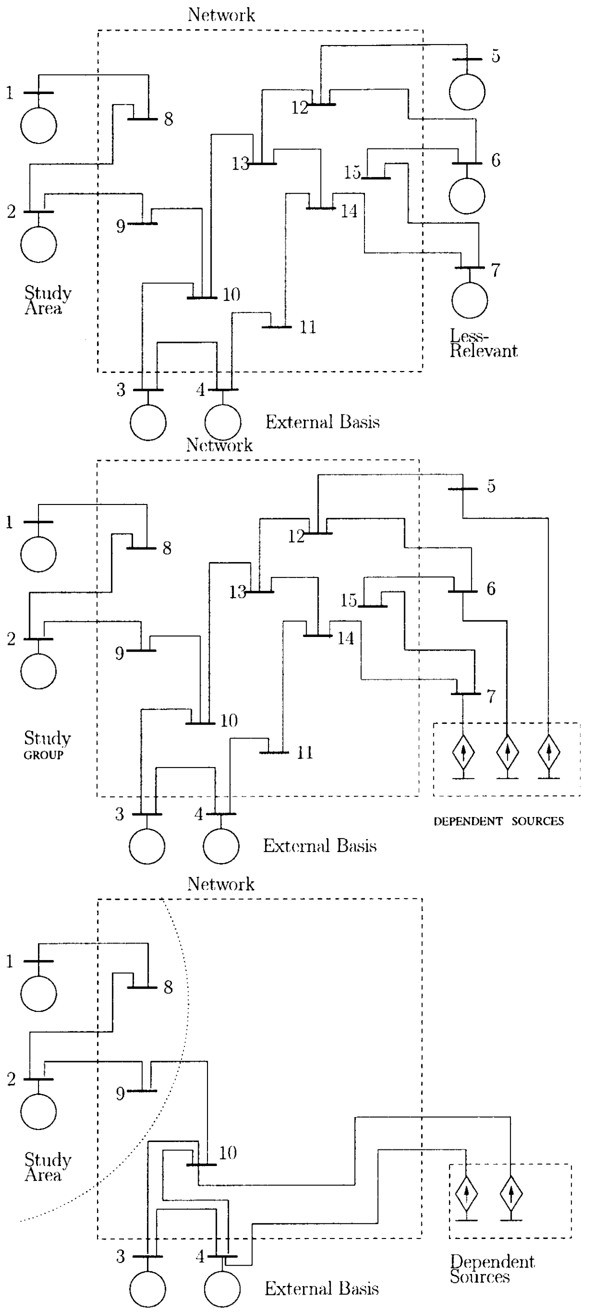


Figure 3.1: **Schematic of Reduction.** The top figure shows a schematic of a power system before reduction, but with the three types of generators (study area, basis and external) already determined. The middle figure shows the reduction of the external generators (the dependent sources are functions of generators three and four). The bottom figure shows the final reduced system where most of the external network has been equivalenced.

which are not in the study area or attached to a basis generator and which are not "close" to the study area, such as buses 5, 6, 7, 11, 12, 13, 14 and 15.

z - **all of the above less-relevant variables**

k - **kept variables in the external reduction** All of the variables external to the study area which are retained (i.e., all ro and zc variables).

Using this notation, we can describe the original system linearization as follows:

$$\begin{bmatrix} \dot{x}_{ri} \\ \dot{x}_{ro} \\ \dot{x}_{z} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{A_{ri,ri}} & \mathbf{0} & \mathbf{0} & \mathbf{B_{ri,ri}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A_{ro,ro}} & \mathbf{0} & \mathbf{0} & \mathbf{B_{ro,k}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A_{z,z}} & \mathbf{0} & \mathbf{B_{z,k}} & \mathbf{B_{z,zf}} \\ \mathbf{C_{ri,ri}} & \mathbf{0} & \mathbf{0} & \mathbf{J_{ri,ri}} & \mathbf{J_{ri,k}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_{k,ro}} & \mathbf{C_{k,z}} & \mathbf{J_{k,ri}} & \mathbf{J_{k,k}} & \mathbf{J_{k,zf}} \\ \mathbf{0} & \mathbf{0} & \mathbf{C_{zf,z}} & \mathbf{0} & \mathbf{J_{zf,k}} & \mathbf{J_{zf,zf}} \end{bmatrix} \begin{bmatrix} x_{ri} \\ x_{ro} \\ x_{z} \\ u_{ri} \\ u_{k} \\ u_{zf} \end{bmatrix} \tag{3.45}$$

3.5.1 Generator Equivalencing

As suggested in our discussion of the basis generator selection, the first step in the equivalencing procedure is to replace all of the less-relevant generator dynamics (all generators except those within the study area and the basis generators) with linear algebraic functions of the basis generator variables. In other words, we create a matrix $\mathbf{K_b}$ and replace x_z using the following relationship:

$$x_z = \mathbf{K_b} x_{ro} \tag{3.46}$$

which is actually a restricted form of a more general possibility:

$$x_z = \mathbf{K} x_r = \begin{bmatrix} \mathbf{K_s} & \mathbf{K_b} \end{bmatrix} \begin{bmatrix} x_{ri} \\ x_{ro} \end{bmatrix} \tag{3.47}$$

We restrict ourselves to only combinations of the basis generator variables for two reasons. First, it leads to a smaller, less-complex model. Second, the study area generators are likely to have their local modes excited by an event within the study area. If we build an equivalent based on these generators, the effect of these local modes is likely to be spread throughout the system, which is undesirable.

We would ideally like a $\mathbf{K_b}$ such that the right eigenvector entries (in the chord) of the retained variables are accurately retained (in the spirit of (3.44)), in other words a $\mathbf{K_b}$ that satisfies:

$$\mathbf{V_{\nu z}} = \mathbf{K_b}\mathbf{V_{\nu ro}} \tag{3.48}$$

With the belief that the extensive modes of the system are the electromechanical modes associated with δ and ω, we only use the rows of $\mathbf{V}$ which correspond to these generator variables to construct our equivalent. Therefore, $\mathbf{K_b}$ is a matrix with two times the number of less-relevant generators columns and a number of rows equal to the sum total of the number of differential variables in all of the less-relevant generators. Though the number of basis generators is equal to the number of modes in the chord (note that each eigenvalue of the core matrix corresponds to two eigenvalues in the full system), the set of equations (3.48) is an underconstrained system since we do not use the basis generator(s) in the study area to build our equivalent.

Due to practical considerations, we limit ourselves even further. Instead of using the full chord ν, we only use the zero-mode. We originally used the core matrix to do the chord and basis generator selection as well as the partitioning because an eigenanalysis of the full set of equations can be very computationally intensive. We do however know the zero-mode left and right eigenvectors *a priori*, based on physical principles (in particular, the right eigenvector has equal entries for every δ variable and is zero everywhere else). Additionally, even if we could perform a full eigenanalysis of the system in a reasonable amount of time, finding those modes of the full system which correspond to the modes of the core matrix is not a trivial problem. The results using this *ad hoc* correction matrix have been quite good, and we will discuss in the appendices ways in which the extra degrees of freedom created by only using the zero-mode to create $\mathbf{K_b}$ can be used constructively. For now, we construct $\mathbf{K_b}$ using the pseudo-inverse:

$$\mathbf{K_b} = \mathbf{V_{0z}}\mathbf{V_{0b}}^{\dagger} \tag{3.49}$$

where the subscript b on $\mathbf{V_{0b}}$ denotes that we only consider the δ and ω variables of the basis generators.

Using the matrix $\mathbf{K_b}$ we can rewrite our system equations and eliminate the x_z variables:

$$\begin{bmatrix} \dot{x_{ri}} \\ \dot{x_{ro}} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{A_{ri,ri}} & \mathbf{0} & \mathbf{B_{ri,ri}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A_{ro,ro}} & \mathbf{0} & \mathbf{B_{ro,k}} & \mathbf{0} \\ \mathbf{C_{ri,ri}} & \mathbf{0} & \mathbf{J_{ri,ri}} & \mathbf{J_{ri,k}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_{k,ro}} + \mathbf{C_{k,z}K_b} & \mathbf{J_{k,ri}} & \mathbf{J_{k,k}} & \mathbf{J_{k,zf}} \\ \mathbf{0} & \mathbf{C_{zf,z}K_b} & \mathbf{0} & \mathbf{J_{zf,k}} & \mathbf{J_{zf,zf}} \end{bmatrix} \begin{bmatrix} x_{ri} \\ x_{ro} \\ u_{ri} \\ u_k \\ u_{zf} \end{bmatrix} \tag{3.50}$$

The study area has not been changed, and the system topology has not been disturbed. We can implement this new system by retaining our old system description and replacing each of the less-relevant generators by a current injector (a controlled current source).

3.5.2 Network Reduction

While the generator equivalencing does substantially reduce the number of differential equations in our system, the system topology remains the same and there is no reduction in the number of algebraic constraints within the system. The ability to group the load nodes in our partitioning scheme opens up further possibilities for reduction. If we restrict ourselves to the study area, the details of the rest of the system only concern us in so far as they impact the study area. So, we can create an equivalent for this external area using a Ward-type reduction and the resulting linear model performs identically to the generator-equivalenced model. Using such a reduction can radically reduce the number of transmission lines, transformers and nodes external to the study area, with a corresponding reduction in the number of algebraic variables. The reduction is only strictly equivalent for a linear system, but since we base our entire method on a linearization, this further reliance on a linear system seems reasonable.

The first step in the network reduction is to algebraically solve for u_{zf} in (3.50):

$$u_{zf} = -\mathbf{J_{zf,zf}}^{-1}(\mathbf{C_{zf,z}K_b}x_{ro} + \mathbf{J_{zf,k}}u_k) \tag{3.51}$$

We only eliminate the less-relevant, far nodes so as to not disturb the study area or our generator equivalencing. We cannot remove nodes close to the study area because that might affect lines directly connected to a study area node; we cannot remove nodes to which basis generators are connected, since the basis generators are not equivalenced. Substituting (3.51) into (3.50), we get:

$$\begin{bmatrix} \dot{x}_{ri} \\ \dot{x}_{ro} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{A_{ri,ri}} & \mathbf{0} & \mathbf{B_{ri,ri}} & \mathbf{0} \\ \mathbf{0} & \mathbf{A_{ro,ro}} & \mathbf{0} & \mathbf{B_{ro,k}} \\ \mathbf{C_{ri,ri}} & \mathbf{0} & \mathbf{J_{ri,ri}} & \mathbf{J_{ri,k}} \\ \mathbf{0} & \mathbf{C_{k,ro}} + \tilde{\mathbf{C}}_{\mathbf{k,z}}\mathbf{K_b} & \mathbf{J_{k,ri}} & \tilde{\mathbf{J}}_{\mathbf{k,k}} \end{bmatrix} \begin{bmatrix} x_{ri} \\ x_{ro} \\ u_{ri} \\ u_k \end{bmatrix} \qquad (3.52)$$

where

$$\tilde{\mathbf{C}}_{\mathbf{k,z}} = \mathbf{C_{k,z}} - \mathbf{J_{k,zf}}\mathbf{J_{zf,zf}}^{-1}\mathbf{C_{zf,z}} \qquad (3.53)$$

$$\tilde{\mathbf{J}}_{\mathbf{k,k}} = \mathbf{J_{k,k}} - \mathbf{J_{k,zf}}\mathbf{J_{zf,zf}}^{-1}\mathbf{J_{zf,k}} \qquad (3.54)$$

Again, we have not affected the study area in any way.

The resulting changes to the system can be represented physically using transmission lines and controlled current sources. All of the less-relevant far nodes have been removed, along with all lines which are connected to a less-relevant far node and any current injectors or loads which were connected at these nodes. Equivalent lines connecting each pair of kept nodes which were previously connected through a path involving only less-relevant far nodes can be computed using $\tilde{\mathbf{J}}_{\mathbf{k,k}}$. Every off-diagonal 2x2 in this matrix represents a new line (remembering that all of our matrices are real so complex admittances are represented in real form using 2x2 blocks). A true complex admittance (which we can represent as a line) has the form:

$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix} \qquad (3.55)$$

However, many of the entries of $\tilde{\mathbf{J}}_{\mathbf{k,k}}$ only approximately have this structure because the original $\mathbf{J}$ matrices have some terms without this structure on their diagonals. Loads which are not modeled as impedances will yield a linearization with a different structure. As noted earlier, the EUROSTAG linearization yields current balance equations and the complex voltage is expressed using its real and imaginary parts. One standard way to model loads is to use the following representation:

$$
\begin{aligned}
P &= P_0 \left|V\right|^{\alpha} && (3.56)\\
Q &= Q_0 \left|V\right|^{\beta} && (3.57)
\end{aligned}
$$

where P and Q are the real and reactive power consumed by the load, P_0 and Q_0 are the nominal real and reactive power consumptions calculated by the load flow algorithm, V is the voltage at the corresponding bus in per unit and α and β determine the type of load modeled. If α and β are both two, we refer to the load as an impedance load. With some algebraic manipulation, we can write the following equation for the contribution of such a load to the current at its bus:

$$
\Re\{I\} - j\Im\{I\} = P_0 V^* \left|V\right|^{\alpha-2} + jQ_0 V^* \left|V\right|^{\beta-2} \tag{3.58}
$$

Taking derivatives of the real and reactive currents with respect to the real and reactive voltages, we derive the entries in the real "admittance" matrix corresponding to such a load:

$$
\begin{aligned}
\frac{\partial\Re\{I\}}{\partial\Re\{V\}} &= P_0 \left|V\right|^{\alpha-2} + (\alpha-2)P_0\Re\{V\}^2 \left|V\right|^{\alpha-4} + \\
&\quad (\beta-2)Q_0\Im\{V\}\Re\{V\} \left|V\right|^{\beta-4} && (3.59)\\
\frac{\partial\Re\{I\}}{\partial\Im\{V\}} &= Q_0 \left|V\right|^{\beta-2} + (\beta-2)Q_0\Im\{V\}^2 \left|V\right|^{\beta-4} + \\
&\quad (\alpha-2)P_0\Im\{V\}\Re\{V\} \left|V\right|^{\alpha-4} && (3.60)\\
\frac{\partial\Im\{I\}}{\partial\Re\{V\}} &= -Q_0 \left|V\right|^{\beta-2} - (\beta-2)Q_0\Re\{V\}^2 \left|V\right|^{\beta-4} + \\
&\quad (\alpha-2)P_0\Im\{V\}\Re\{V\} \left|V\right|^{\alpha-4} && (3.61)\\
\frac{\partial\Im\{I\}}{\partial\Im\{V\}} &= P_0 \left|V\right|^{\alpha-2} + (\alpha-2)P_0\Im\{V\}^2 \left|V\right|^{\alpha-4} + \\
&\quad (\beta-2)Q_0\Im\{V\}\Re\{V\} \left|V\right|^{\beta-4} && (3.62)
\end{aligned}
$$

The appropriate structure is present if both α and β are equal to two. An additional problem is the small signal "admittance" of the less-relevant generators. These terms also appear on the diagonal of the **J** matrices and are not in general true admittances, so they also will not have the appropriate structure. Nevertheless, in practical cases, we have found that most 2x2 blocks do have approximately the appropriate structure, so we take an arithmetic average of the terms and create the transmission lines using these averaged terms.

The current injectors comprise three terms, as shown in the following equation:

$$I = I_0 + \tilde{\mathbf{D}}u_k + \tilde{\mathbf{C}}_{\mathbf{k},\mathbf{z}}\mathbf{K}_\mathbf{b}x_{ro} \tag{3.63}$$

where I_0 is a vector of the nominal current injections at each node, $\tilde{\mathbf{D}}$ is the small signal admittance of the current injector and $\tilde{\mathbf{C}}_{\mathbf{k},\mathbf{z}}\mathbf{K}_\mathbf{b}$ is the previously discussed dynamic equivalent. The dynamic equivalent comprises any less-relevant generators which were originally close to the study area as well as the effect of the less-relevant generators at nodes which were removed by the network reduction, since those current injectors were also removed. The small signal admittance term can be calculated using the diagonal 2x2 blocks of $\tilde{\mathbf{J}}_{\mathbf{k},\mathbf{k}}$. Those blocks of $\tilde{\mathbf{J}}_{\mathbf{k},\mathbf{k}}$ comprise three terms. First, the negative of the sum of the admittances of the new lines is present, as in an admittance matrix. Second, an additional set of terms is present, which correspond to the compensation necessary to account for the removal of lines. The original $\mathbf{J}_{\mathbf{k},\mathbf{k}}$ had terms on its diagonal which corresponded to the negative of the sum of the admittances connected to each node; however, many of these lines are no longer present, which must be reflected in the diagonal as well as the off-diagonal terms. Third, we have our term $\tilde{\mathbf{D}}$, which in a manner similar to the dynamic equivalent term reflects small-signal admittances of removed less-relevant generators.

As an example, we refer to Figure 3.1. Nodes 5, 6, 7, 11, 12, 13, 14 and 15 are all less-relevant far nodes, so they are removed along with all of the lines connected to them. A new line is created connecting node 4 and node 10. These two nodes were previously connected through at least one path involving only removed nodes (e.g., through 11, 13 and 14), so a new line is necessary to compensate for the removal of the far nodes. No other new lines are necessary since neither the combination of 3 and 4 nor the combination of 3 and 10 were previously connected through a path involving only removed nodes. The network reduction removes all of the nodes which previously had current injectors connected to them. Two new current injectors (at node 10 and node 4) are necessary to compensate. Node 3 does not have a current injector because it is not connected to a removed current injector through a path involving only removed nodes.

On a final note, we return to an issue mentioned in Section 3.4, namely the connectivity of the study area. A cursory examination of the network reduction technique shows that if we have fewer kept nodes in the external area we also have fewer new transmission lines and fewer current injectors. The number of groups determines the number of basis generators, but the shape and size of the study area are the primary factors in determining how many less-relevant close nodes there are; we also note that in a large system the less-relevant close nodes greatly outnumber the number of basis generators. A reasonable assertion which is often but not always true is that a connected study area leads to fewer less-relevant close nodes. A few extreme examples should illustrate the point. If the study area was completely disconnected, each of the study area nodes could be associated with at least one less-relevant close node, since any node connected to a study area node would be close to but not in the study area by construction. On the other hand, many of the study area nodes in a completely connected study area are only connected to other study area nodes; consequently, they do not add to the number of close nodes. Because of our definition of connectivity, counter-examples are easy to create, but in practice this principle seems to hold.

3.6 Summary

We have attempted to present the salient features of the SME algorithm in this chapter. First, a simplified, linearized model is derived, which is amenable for eigenanalysis and has nice properties. Then, this eigenanalysis is used to classify the modes of the system. Some of the modes of the system are used to select basis generators, which are used to partition the system, such that every node in the system belongs to exactly one group with an associated basis generator. Based on the partitioning, a study area is selected; the study area is not directly affected by our equivalencing procedure. The dynamics of all of the non-basis generators external to the study area are replaced by controlled current sources, which are functions of the basis generators. In addition, most of the detail in the external network is replaced by a substantially smaller equivalent (in the linearization) network. The resulting equivalent model can be implemented using physical power system components.

The reliance on a linearized model is a disadvantage, since our simulations are performed on a nonlinear model, but the advantages of having the tools of linear system analysis at our disposal seem to outweigh this disadvantage. Another major disadvantage is the heuristic nature of the algorithm, since this makes analysis of the algorithm difficult. However, we have found the results of our testing so far to be very promising; hopefully, many of the theoretical issues can be addressed with time. In addition to our promising results, an overwhelming advantage of this approach is that it can be implemented using commercially available software, namely MATLAB and EUROSTAG. We hope that this will lead to greater acceptance within the power community, where a great deal of emphasis is placed upon physical intuition (which may be lost if the system variables are not retained).

Most of our discussion is derived from previously published work, but we do note that some of the work in this chapter is original. In particular, our analysis in Section 3.2 had not been previously presented. We had developed the model extraction procedure without examining closely how well it corresponded to our assumed model, the swing-equation model. Our analysis shows that the core model and extended core model are not Laplacian matrices though they do have some of the desired properties exactly and others approximately. This work may be helpful for current research into the connections between model reduction and graph theory. In addition, the need for power balance equations consistently throughout the extended core matrix was previously noted, but the connection to the actual core matrix used in practice had not been made.

Chapter 4

Uncertain Parameters - Modeling and Prioritizing

So far, we have discussed reducing the size of a nominal (certain) power system model and uncertainty analyses of a (reduced-order) power system model given that the parameter uncertainties are already well-defined. However, we have not yet discussed how we transition from the initial parameter uncertainty descriptions for the full model to similar descriptions for the reduced-order model. In addition, we do not want to have to treat every system parameter as uncertain because the uncertainty analysis quickly becomes infeasible for large parameter sets, as pointed out in our probabilistic collocation method discussion. A system by which we can prioritize the uncertainties is needed. The present chapter presents these topics and essentially completes the description of our entire uncertainty analysis approach.

All of the study area and also the basis generators are not modified in any way by the model reduction. Therefore, we retain the uncertainty descriptions of these elements from the original full model. On the other hand, the less-relevant far nodes, the lines connected to these nodes and any non-basis generators outside of the study area are removed. In place of these, the reduced-order model has new lines and current injectors. Each of these new elements is a function of a large subset of the removed elements. Therefore, the uncertainties of these new elements' parameters can be quite difficult to compute and are likely to be dependent on each other (in the usual probability sense); both of these properties are undesirable.

Fortunately, though each of the new elements is a function of a large subset of the removed elements, the new element's parameters are largely determined by a small set of the original model parameters, with the remaining functional dependence having a minimal effect. Our job is to determine which of the original parameters have the greatest impact on any given new parameter. This approach can also be used to solve the issue of dependency among the new parameters. For each of the new parameters, we determine a list of the original parameters which affect it. We then use the most "important" original parameters for each of the new parameters in our uncertainty analysis by changing these original parameters and computing the resulting new parameter values. In other words, if $\alpha(x, y)$ is a parameter in the reduced-order model which is primarily affected by original parameters x and y, we simply create our PCM model using x and y as our uncertain parameters and compute the appropriate α's based on the x and y collocation points.

A similar rank-ordering of uncertain parameters is necessary for the entire reduced-order model. We propose a method of prioritizing the uncertainties based on the eigenvalue sensitivities of the system and the standard deviation of each uncertainty. The eigenvalue sensitivities of the system are helpful in indicating which of the parameters have the greatest impact on the system behavior [45], while the standard deviation of each uncertainty indicates which of the uncertainties actually varies over a large range. We seek parameters that are relatively uncertain and also have an impact on system behavior.

4.1 Analysis of the External Reduction

We begin by reiterating the equations for the linear reduced-order system (3.52):

$$\begin{bmatrix} \dot{x_{ri}} \\ \dot{x_{ro}} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{A_{ri,ri}} & \mathbf{0} & \mathbf{B_{ri,ri}} & \mathbf{0} \\ \mathbf{0} & \mathbf{A_{ro,ro}} & \mathbf{0} & \mathbf{B_{ro,k}} \\ \mathbf{C_{ri,ri}} & \mathbf{0} & \mathbf{J_{ri,ri}} & \mathbf{J_{ri,k}} \\ \mathbf{0} & \mathbf{C_{k,ro}} + \mathbf{\tilde{C}_{k,z}K_b} & \mathbf{J_{k,ri}} & \mathbf{\tilde{J}_{k,k}} \end{bmatrix} \begin{bmatrix} x_{ri} \\ x_{ro} \\ u_{ri} \\ u_k \end{bmatrix} \tag{4.1}$$

where

$$\mathbf{\tilde{C}_{k,z}} = \mathbf{C_{k,z}} - \mathbf{J_{k,zf}J_{zf,zf}}^{-1}\mathbf{C_{zf,z}} \tag{4.2}$$

$$\mathbf{\tilde{J}_{k,k}} = \mathbf{J_{k,k}} - \mathbf{J_{k,zf}J_{zf,zf}}^{-1}\mathbf{J_{zf,k}} \tag{4.3}$$

The system reduction is summarized in $\mathbf{\tilde{C}_{k,z}}$ and $\mathbf{\tilde{J}_{k,k}}$ and all of the current injectors and new transmission lines are created based solely on them. These matrices are algebraic representations of well-known circuit reduction techniques; for example, $\mathbf{\tilde{J}_{k,k}}$ could be calculated directly from a system schematic (one-line diagram) by finding the equivalent impedance between each of the kept external nodes (with a few extra conditions which we explain later).

Based on this knowledge, we see at least two ways to proceed with the analysis of the network reduction. First, we can attempt to identify which of the parameters in the original full model have the greatest effect on any given new parameter using a topological analysis of the system. In other words, by studying the interconnections among the nodes of the system, we gain a lot of information without directly referring to matrices. An advantage of this approach is that we avoid (costly) matrix inverses in our analysis. However, we have found that a direct computation approach based on the matrices is actually faster. But, we still outline our investigations in this area because the work is interesting and helps explain why almost every new line and current injector in the reduced-order system is only highly sensitive to a small subset of the original parameters in the system.

A second approach is to simply analyze these matrices, without relying explicitly on circuit or topological considerations. We can always compute the relative sensitivity of each of the new parameters to each of the original parameters, but since there are tens or even hundreds of new parameters and hundreds or even thousands of original parameters, this task is daunting. (See [46] for an introduction to sensitivity analysis). Nevertheless, we compute and sort all of this information as part of our uncertainty analysis technique in a relatively short period of time. To accomplish this, we repetitively use a well-known result about matrix inverses, which we discuss later.

4.1.1 Topological Approach

Before outlining feasible approaches to analyzing the external reduction topologically, we present a simple example which illustrates the different types of nodes and some important characteristics of the network. Our example is shown schematically in Figure 4.1.

The external far nodes are removed and new lines are created to replace those removed. The resulting system is shown in Figure 4.2. We wish to draw the reader's attention to a few properties illustrated here. First, a new line is created between node pairs such as A and B, where these kept nodes were previously connected through a path involving only removed nodes. New lines are not created between node pairs such as D and B because

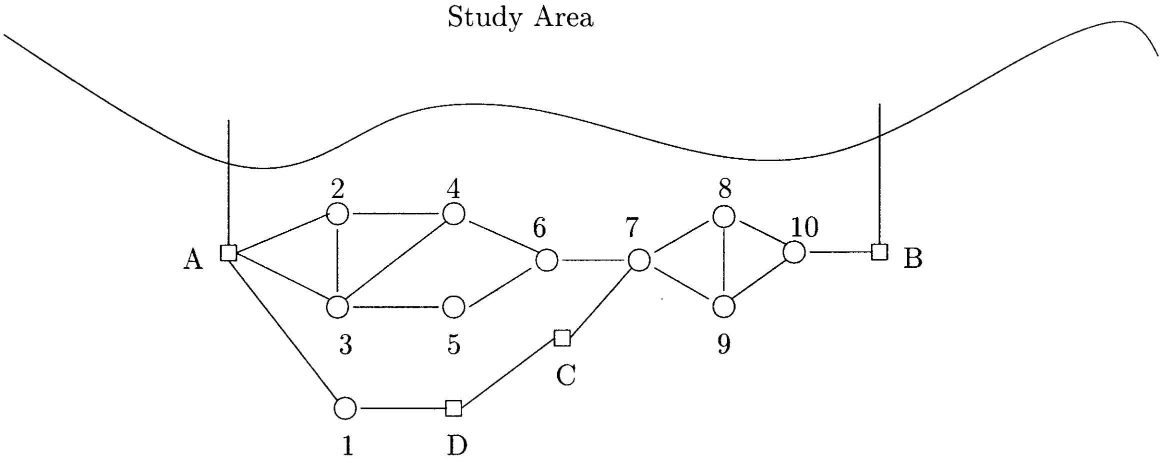


Figure 4.1: **External Network Before Reduction.** This figure is a very simple schematic of the interconnections in the portion of an example system external to the study area. The squares represent nodes that will be kept (i.e., A and B are close to the study area and C and D are basis generator nodes). The circles represent all other external nodes.

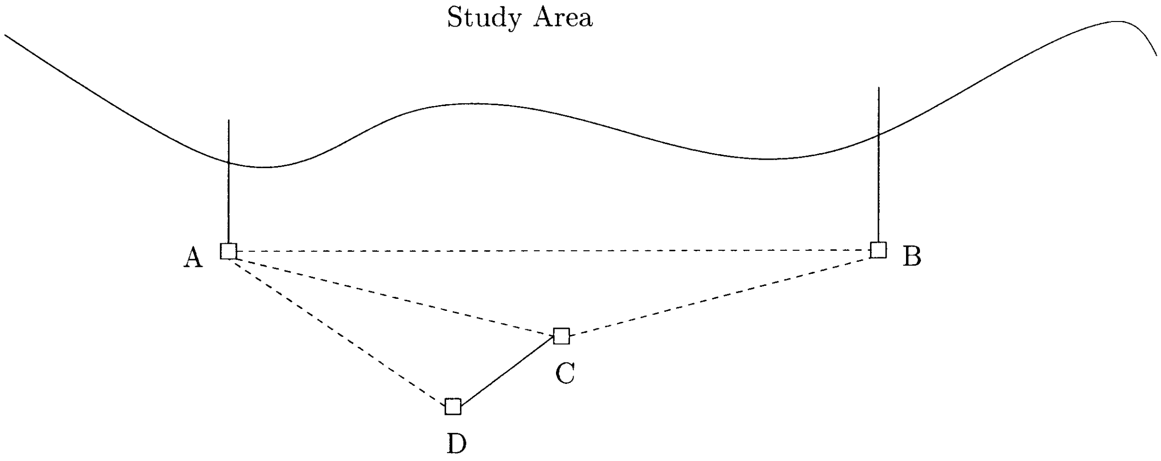


Figure 4.2: **External Network After Reduction.** This system is the one in Figure 4.1 but after the external network has been equivalenced. The dashed lines represent new transmission lines created by the reduction.

these nodes are only connected through paths involving at least one node which is retained. Second, the line connecting C and D is not disturbed by the reduction and this line does not influence any of the new parameters. If C and D were also connected through some path involving only removed nodes, a new line would be connected in parallel with the existing line. Third, though we cannot show it graphically on the diagram, the new line between nodes A and D is only dependent on the removed lines between nodes A and 1 and nodes D and 1, as well as any load or current injector at node 1. Fourth, the removed line between nodes C and 7 influences the new lines between nodes A and B, nodes B and C and nodes A and C. From the perspective of the new line between nodes A and B, the removed line between nodes C and 7 looks like a transmission line to ground.

Let us concentrate for a moment on the new line between nodes A and B. Several possible topological criteria for sorting the original lines which influence this new line could be imagined. (A problem which runs throughout our topological discussions is the lack of an adequate way to treat loads and current injectors. Almost all of our analyses in this section are limited to the influence of removed lines on new lines). One approach is to calculate the shortest path between nodes A and B in Figure 4.1 (based either on impedances of the lines or just by "hops") and conclude that the removed lines and nodes on this path have the greatest influence on the parameters of the new line. Efficient algorithms (for example, Dijkstra's algorithm) exist for finding this shortest path [47]. In an electrical network context, little justification for such an approach exists because it ignores the interaction of parallel paths, and we have obtained dismal results using such an approach on our power system examples.

Another approach is inspired by the theory of localized response in power systems. (See [48], [49] and [50] for discussions of localized response). Those results are in an entirely different context than ours and the theory as formulated is not relevant for our purposes. Nevertheless, we borrow the idea and claim that the removed nodes and lines closest to either of nodes A and B have the greatest influence on the parameters of the new line connecting these nodes. In our context, the removed lines near either node A or B are less likely to be in parallel with a large number of other removed lines than are other more distant removed lines, which we might hope dilutes the effect of these near removed lines less.

Let us define tiers for the system in a similar way to the definition used in the localized

Tier	Line(s) and Node(s)
1	A, B
2	A-2, A-3, B-10
3	2, 3, 10, 2-3
4	2-4, 3-4, 3-5, 10-8, 10-9
5	4, 5, 8, 9, 8-9
6	4-6, 5-6, 8-7, 9-7
7	6, 7, 6-7
8	7-C

Table 4.1: **Tier Example.** We apply our tier designations to the new line connecting nodes A and B in our example from Figures 4.1 and 4.2.

response context. Our definition of tier encompasses both nodes and lines and is based on the original system topology. The two nodes to which the new line is connected are designated as tier one. The lines directly connected to these nodes are designated as tier two. The nodes to which these lines are connected as well as lines which are connections between such nodes are designated as tier three. The definition continues in this fashion. In all cases, the minimum such tier numbering which applies to a given node or line is the one used; this issue arises because the tier could be calculated from either of the two original nodes. We refer to Table 4.1 for the designations of all tiers for the new line connecting nodes A and B in our example. We neglect all removed lines and nodes which are not on a path connecting nodes A and B involving only less-relevant far nodes.

If we look at the original algebraic equations relevant to a particular new line (that is, the equations before we perform the network reduction), this tier definition emphasizes certain structures in these equations. In particular, letting $\mathbf{J_i}$ represent the parameters associated with lines or nodes in tier i, we see that the following equations for a 5-tier example are quite structured (the generalization to n-tier should be apparent):

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{J_1} & \mathbf{J_2} & \mathbf{0} \\ \mathbf{J_2}^{\mathrm{T}} & \mathbf{J_3} & \mathbf{J_4} \\ \mathbf{0} & \mathbf{J_4}^{\mathrm{T}} & \mathbf{J_5} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{4.4}$$

where the u_i are appropriately ordered algebraic variables in the system.

We are now in a position to compare the sensitivity of the new line to changes in different tiers of the original system. For example, we can compute the effect of changes to the third and fifth tiers of (4.4), $\mathbf{\Delta_3}$ and $\mathbf{\Delta_5}$, and see if the system is relatively more sensitive to changes to tier three than it is to changes to tier five. We present the matrices which represent the changes to the new system description caused by such perturbations of the original system (to first-order):

$$\begin{aligned} \mathbf{\Delta_3} &\Rightarrow \mathbf{J_2}\left(\mathbf{J_3}-\mathbf{J_4}\mathbf{J_5}^{-1}\mathbf{J_4}^{\mathrm{T}}\right)^{-1}\mathbf{\Delta_3}\left(\mathbf{J_3}-\mathbf{J_4}\mathbf{J_5}^{-1}\mathbf{J_4}^{\mathrm{T}}\right)^{-1}\mathbf{J_2}^{\mathrm{T}} && (4.5)\\ \mathbf{\Delta_5} &\Rightarrow \mathbf{J_2}\left(\mathbf{J_3}-\mathbf{J_4}\mathbf{J_5}^{-1}\mathbf{J_4}^{\mathrm{T}}\right)^{-1}\mathbf{J_4}\mathbf{J_5}^{-1}\mathbf{\Delta_5}\mathbf{J_5}^{-1}\mathbf{J_4}^{\mathrm{T}}\left(\mathbf{J_3}-\mathbf{J_4}\mathbf{J_5}^{-1}\mathbf{J_4}^{\mathrm{T}}\right)^{-1}\mathbf{J_2}^{\mathrm{T}} && (4.6)\end{aligned}$$

Borrowing from the localized response literature, we assert further that:

$$\|\mathbf{J_5}^{-1}\mathbf{J_4}^{\mathrm{T}}\|_\infty < 1 \qquad (4.7)$$

Although these results are very suggestive, they do not conclusively prove much for our case since the norm of a product is not the product of the norms:

$$\|\mathbf{AB}\| \neq \|\mathbf{A}\|\|\mathbf{B}\| \qquad (4.8)$$

Our experience with this scheme has shown some promise. In many cases, the tier method accurately identifies the original parameters which have the greatest influence on the new parameters. However, in a nontrivial number of cases, this method performs poorly; we return to this issue directly. In addition, accurately delineating all of the tiers is time-consuming; determining from which direction the tier of a particular node or line should be computed (with respect to node A or node B in our example) requires some time and effort. An alternate approach is to determine every removed node and line which is on a tier less than n; such a computation is extremely fast, but it does not resolve the performance issue mentioned previously.

A final topological approach attempts to address those cases where the tier method fails. After extensive testing, we concluded that the original nodes and lines to which a given new line was sensitive but which were distant from the nodes connected to the new line (i.e., on a high tier) fall into two general categories. This nice categorization of the problem elements is somewhat surprising since our tier method does not use any information about the impedances of lines or in fact any system information other than topology; it seems that simply the topology of the network contains much knowledge of interest. In any event, we describe and illustrate these two categories of problem cases using our example.

The line connecting nodes 6 and 7 embodies one of the problem cases. This line is in the second to last tier, but our experience indicates that it is likely to be very important for categorizing the new line between nodes A and B. The explanation is also topological; if we were to cut this line in the original system, node A and node B would no longer require a new line connecting them in the reduced-order model. In the language of graphs, this line is part of a small cutset of the network connecting nodes A and B. Generally, original lines through which every (or almost every) path connecting the two kept nodes must pass are important, for obvious reasons. Identifying lines which constitute a cutset by themselves (such as the one connecting nodes 6 and 7) is easy. As the size and complexity of the system increases, we may need to identify cutsets which comprise more than one line; since the identification of all cutsets in the system belongs to the class of NP-complete problems, the computation needed to identify cutsets comprising more than a few lines is impractical.

The other problem case is represented by the original line connecting nodes C and 7 in our example. Since this line is essentially a line to ground (from the perspective of the new line connecting nodes A and B), it can also have a substantial effect on the new line, even though it is in the very last tier. Identifying such lines is equivalent to determining the cutsets for the networks connecting nodes B and C and nodes A and C. As far as computation, we have exactly the same problem as we just mentioned in the case of the line connecting nodes 6 and 7.

So, a refined tier method which also addresses our "problem" lines is impractical. Nevertheless, our topological methods do perform reasonably well considering how simplistic they are. Topological methods may be developed which can perform the necessary calculations and prioritization even quicker as the connections between graph theory and power system are explored further, but at the present time, we recommend direct methods, which we describe in the next section. Fortunately, direct computation methods exist, which yield reasonable results quickly. The topological methods are however useful because they do indicate that only a subset of the original parameters have a large impact on the new parameters in the external network reduction.

4.1.2 Direct Computation of Sensitivities

By sensitivity, we mean relative sensitivity. The relative sensitivity of reduced-order system parameter $\tilde{y}$ to original system parameter y is given by the following expression:

$$\left(\frac{\partial \tilde{y}}{\partial y}\right)\left(\frac{y}{\tilde{y}}\right) \tag{4.9}$$

The first term in the product is the absolute sensitivity of $\tilde{y}$ with respect to y, while its product with the second term is useful for us because the result has no units which allows us to compare sensitivities with respect to disparate original parameters. One interpretation of this quantity is that it measures the percentage change to $\tilde{y}$ with respect to a percentage change in y. Such a measure is useful for us because the uncertainty in a parameter such as a line impedance is not absolute but rather the magnitude of the uncertainty scales with the magnitude of the line parameter.

Our goal in this section is to outline a practical method for computing such sensitivities. The particular form of the sensitivity and the method of computing it depend on both $\tilde{y}$ and y. We seek to be quite general by deriving our results using matrices; nevertheless, our actual program must differentiate among several classes of problems. We present here one of the most involved cases for illustration purposes. Afterwards, we discuss some issues particular to certain other classes of $\tilde{y}$ and y.

We restrict ourselves for now to a particular type of y, the transmission line parameters of the original lines between external kept nodes and external far nodes. These lines are removed in the external reduced network. We further limit ourselves by only looking at the effects of perturbations of such y on (4.3); in other words, we only examine the $\tilde{y}$ which represent parameters of new lines and the static parts of the current injectors. For concreteness, let us examine perturbing the conductance of a line connecting the ith kept node to the jth less-relevant far node. Such a perturbation is present in four of our system matrices:

$$\begin{aligned}
\mathbf{J_{k,k}} &\rightarrow \mathbf{J_{k,k}} + \boldsymbol{\Delta}_{\boldsymbol{k},\boldsymbol{k}} && (4.10)\\
\mathbf{J_{k,zf}} &\rightarrow \mathbf{J_{k,zf}} + \boldsymbol{\Delta}_{\boldsymbol{k},\boldsymbol{zf}} && (4.11)\\
\mathbf{J_{zf,zf}} &\rightarrow \mathbf{J_{zf,zf}} + \boldsymbol{\Delta}_{\boldsymbol{zf},\boldsymbol{zf}} && (4.12)\\
\mathbf{J_{zf,k}} &\rightarrow \mathbf{J_{zf,k}} + \boldsymbol{\Delta}_{\boldsymbol{zf},\boldsymbol{k}} && (4.13)
\end{aligned}$$

where

$$\left(\Delta_{k,k}\right)_{l,m} = \begin{cases} \Delta_0 & \text{if } l = m = i \text{ or } l = m = i+1 \\ 0 & \text{otherwise} \end{cases} \qquad (4.14)$$

$$\left(\Delta_{k,zf}\right)_{l,m} = \begin{cases} -\Delta_0 & \text{if } (l = i \text{ and } m = j) \text{ or } (l = i+1 \text{ and } m = j+1) \\ 0 & \text{otherwise} \end{cases} \qquad (4.15)$$

$$\left(\Delta_{zf,zf}\right)_{l,m} = \begin{cases} \Delta_0 & \text{if } l = m = j \text{ or } l = m = j+1) \\ 0 & \text{otherwise} \end{cases} \qquad (4.16)$$

$$\left(\Delta_{zf,k}\right)_{l,m} = \begin{cases} -\Delta_0 & \text{if } (l = j \text{ and } m = i) \text{ or } (l = j+1 \text{ and } m = i+1) \\ 0 & \text{otherwise} \end{cases} \qquad (4.17)$$

The resulting new $\tilde{\mathbf{J}}_{\mathbf{k,k}}$ can then be written in terms of these perturbation matrices:

$$\tilde{\mathbf{J}}'_{\mathbf{k,k}} = \left(\mathbf{J_{k,k}} + \boldsymbol{\Delta}_{\boldsymbol{k},\boldsymbol{k}}\right) - \left(\mathbf{J_{k,zf}} + \boldsymbol{\Delta}_{\boldsymbol{k},\boldsymbol{zf}}\right)\left(\mathbf{J_{zf,zf}} + \boldsymbol{\Delta}_{\boldsymbol{zf},\boldsymbol{zf}}\right)^{-1}\left(\mathbf{J_{zf,k}} + \boldsymbol{\Delta}_{\boldsymbol{zf},\boldsymbol{k}}\right) \qquad (4.18)$$

The term to be inverted is often quite large (a large $\mathbf{J_{zf,zf}}$ indicates a substantial system reduction), and its inversion can be quite time-consuming. However, using the following well-known result from linear algebra, the problem becomes tractable [51]:

$$\left(\mathbf{A} + \mathbf{BCD}\right)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B}\right)^{-1}\mathbf{DA}^{-1} \qquad (4.19)$$

This result is useful because $\boldsymbol{\Delta}_{\boldsymbol{zf},\boldsymbol{zf}}$ has a small rank (in our example, rank two).

Let y and z be two-column matrices such that their outer product is $\boldsymbol{\Delta}_{\boldsymbol{zf},\boldsymbol{zf}}$:

$$\boldsymbol{\Delta}_{\boldsymbol{zf},\boldsymbol{zf}} = yz^{\mathrm{T}} \qquad (4.20)$$

Using (4.19), we obtain the following result:

$$\left(\mathbf{J_{zf,zf}} + \boldsymbol{\Delta}_{zf,zf}\right)^{-1} = \mathbf{J_{zf,zf}}^{-1} - \mathbf{J_{zf,zf}}^{-1} y \left(\mathbf{I} + z^{\mathrm{T}} \mathbf{J_{zf,zf}}^{-1} y\right)^{-1} z^{\mathrm{T}} \mathbf{J_{zf,zf}}^{-1} \tag{4.21}$$

This greatly speeds up the computation of the inverse for two reasons. First, we have already computed $\mathbf{J_{zf,zf}}^{-1}$ as part of the external reduction. In other words, this inverse requires no extra computation here; we must merely save it from our earlier calculation. Second, the other inverse term $\left(\mathbf{I} + z^{\mathrm{T}} \mathbf{J_{zf,zf}}^{-1} y\right)^{-1}$ is a two by two matrix, which presents no real problems. The result (4.19) has allowed us to transform our earlier outer product of y and z into an inner product. We do not discuss the details here, but the order in which the matrix multiplications are performed also has an effect on the speed of the computation. In brief, the ordering must always be chosen to maximize the benefit of the lower dimension of each matrix.

Using these results, we can efficiently calculate (4.18). In practice, we only compute the constant and first order terms in Δ_0. The (absolute) sensitivity is then easily obtained by picking out the appropriate term(s) of $\tilde{\mathbf{J}}'_{\mathbf{zf,zf}}$ corresponding to the desired $\tilde{y}$, subtracting off the nominal $\tilde{y}$ and dividing by Δ_0. The final step is to calculate the relative sensitivity using (4.9).

In our example, we describe a change to the conductance of a particular transmission line. We are able to easily determine a suitable Δ_0 for such an example, but changes to loads (for example) cannot be modeled so easily. Similarly, if our $\tilde{y}$ is a current injector parameter, determining the appropriate parameters in $\tilde{\mathbf{J}}'_{\mathbf{zf,zf}}$ to examine is not trivial. We have not worried ourselves excessively with these details, because we seek in the end a rank-ordered list and not necessarily a 100% accurate determination of the sensitivity for each element. If we use good engineering intuition and common sense, such particulars should not be overly critical.

After we have performed these calculations for all possible y, we separate the results into groups based on the different $\tilde{y}$ and within each group the results are sorted in decreasing order. A larger relative sensitivity indicates greater importance. We can then either look through each sorted list and pick a cutoff point or arbitrarily set a uniform cutoff criterion (such as all elements with a sensitivity within 50% of the greatest sensitivity for that $\tilde{y}$); in either event, we now have a (short) list of original parameters for each of the new parameters, which was our goal.

A useful modification to this approach is to further weight the relative sensitivities based on the uncertainty in the original parameters before we sort our lists. As a simple measure without units, we use the standard deviation divided by the expected value of each of the original parameters to indicate the relative uncertainty of each of these elements. If a new parameter is highly sensitive to a particular original parameter, but that original parameter is very well-known, the original parameter may not be "crucial" to the *uncertainty* of the new parameter. Therefore, we sort our original parameters according to the following modified sensitivity formula:

$$\left(\frac{\partial \tilde{y}}{\partial y}\right)\left(\frac{y}{\tilde{y}}\right)\left(\frac{\sigma_y}{\mathrm{E}[y]}\right) \tag{4.22}$$

4.1.3 New Element Uncertainties

A lingering problem which we need to discuss at least briefly before we identify the "key" uncertainties in our entire reduced-order model is calculating the uncertainty in each of our new elements. If our definition of key uncertainty is to include both system sensitivity as well as relative uncertainty of each of the parameters, we need to know those uncertainties. Of course, this task is trivial for the study area, since we assume that the uncertainties in the original model are already known. However, for the new parameters created by the external network reduction, the uncertainties are unknown.

Unfortunately, determining even just the standard deviation and expected value for each of the new parameters is far too computationally intensive to be practical. We fall back on the list of original parameters we just generated for each of the new parameters to help us approximate these quantities. Assuming that our list of original parameters really comprises the uncertainties which have the greatest effect and that the combined effect of all of the other (less critical) original parameters does not dominate, we can calculate the expected value and standard deviation of each new parameter based on only the original parameters on its list (treating all other original parameters as if they were known with certainty). Since a matrix inverse can be written as a quotient of determinants, the functional dependence of the new parameter on this list of original parameters is a quotient of polynomials in these original parameters. (Since every possible cross-term could be present, our list of original parameters must be very short for us to be capable of identifying this quotient). We can then perform Monte Carlo "simulations" on these functions to determine the expected value and standard deviation of each of the new parameters quickly.

4.2 Identification of Key Parameter Uncertainties

At least two general approaches exist to solve the problem of identification of key parameter uncertainties. The first is to examine the actual planned simulations and determine somehow which parameter uncertainties have the greatest effect on these simulations. The second is to look at the overall system and determine which uncertainties are the most "important" in general. Of course, for a particular simulation, the first approach should have better performance, but it also obviously limits the applicability of the analysis to only that simulation. We prefer the second approach for its greater generality. In addition, we can perform analyses under the umbrella of the second approach, which are in the same spirit as those we have previously presented, as we see in this section.

The basic technique we use is inspired by recent work on the identification of key load dynamics for system damping in power systems. (See [52], [53], [45], [54] and [55] for an overview of this interesting area of research). While this research encompasses two separate approaches (which are compared in [55]), eigenvalue sensitivities are at the core of the work described by both research groups. Eigenvalue sensitivities are a natural choice for ranking loads if one is primarily interested in the effects of modeling load dynamics on system stability. If a change to a particular load model were to shift the system eigenvalues towards (or across) the y-axis, we would certainly be interested in knowing that.

Since the eigenstructure of the system completely describes its behavior (assuming that the linearization is a fair characterization of the system), eigenvalue sensitivities serve well as a general measure of the impact on the overall system of any particular parameter uncertainty. In addition, our model reduction method (SME) is based on the modal structure of the system, so using a modal technique here as well has a satisfying consistency. Practically, eigenvalue sensitivities are simple to calculate.

4.2.1 Our Method - Introduction

We again start with the equations describing the linear reduced-order model (3.52), but we write them more compactly:

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{E}} \begin{bmatrix} \dot{x} \\ \dot{u} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{J} \end{bmatrix}}_{\boldsymbol{\Psi}} \begin{bmatrix} x \\ u \end{bmatrix} \tag{4.23}$$

Let us assume that the system has n finite, non-zero eigenvalues. (Obviously, we have a generalized eigenproblem here and we return to that issue in the next section). We evaluate the following expression for each uncertain parameter y in the system:

$$\sum_{i=1}^{n} \left| \frac{\partial \lambda_i}{\partial y} \right| \left| \frac{y}{\lambda_i} \right| \left| \frac{\sigma_y}{\mathrm{E}[y]} \right| \tag{4.24}$$

where the first term in the product is the absolute sensitivity of the eigenvalue to parameter y, the second term is the quotient of the nominal value of the parameter and the nominal value of the eigenvalue, and the last term is the standard deviation of the parameter divided by its expected value. We ignore the zero eigenvalue for obvious reasons. (The zero eigenvalue should be insensitive to parameter changes anyway).

This measure is heuristic, since we do not attempt to prove that the addition over all eigenvalue sensitivities is an optimal measure of overall system sensitivity. However, this measure is a reasonable way to combine all of the sensitivities into one single scalar quantity, which can be computed quickly. The quantity is also unitless and based on both the relative sensitivity of the eigenvalues and the uncertainty in the parameter (as measured by standard deviation).

We perform this calculation for all of the uncertain parameters in the system (parameters created by the external reduction as well as those in the study area). These values can then be sorted in descending order, and the parameters with the largest values are designated "key" uncertainties. We use the key uncertainties in our uncertainty analyses and use nominal values for all other parameters. If any of these key uncertainties are from the external reduction, we simply identify the original parameters from the list of important original parameters that we generated for this new parameter and use those original parameters in our uncertainty analyses.

4.2.2 Our Method - Practical Issues

The sensitivity formula for the eigenvalues in the generalized eigenvalue problem is very similar to that for the regular eigenvalue problem. Let us denote a perturbation to $\boldsymbol{\Psi}$ in the appropriate place(s) to represent parameter y by $\boldsymbol{\Delta_y}$. Concentrating for the moment on a particular eigenvalue λ and its associated right and left eigenvectors, v and w^{T}, we have the following formula for the perturbation to λ caused by $\boldsymbol{\Delta_y}$:

$$\Delta\lambda = \frac{w^{\mathrm{T}}\boldsymbol{\Delta_y} v}{w^{\mathrm{T}}\mathbf{E}v} \tag{4.25}$$

To compute our absolute eigenvalue sensitivity, we divide this quantity by the norm of $\boldsymbol{\Delta_y}$.

Fortunately, we only need the eigenvectors associated with finite eigenvalues, which lessens the computational burden. We first compute the eigenvalues and eigenvectors of $\boldsymbol{\Psi_r}$:

$$\boldsymbol{\Psi_r} = \mathbf{A} - \mathbf{B}\mathbf{J}^{-1}\mathbf{C} \tag{4.26}$$

The eigenvalues of $\boldsymbol{\Psi_r}$ are identically the finite eigenvalues of our generalized eigenvalue problem. Let the right and left eigenvector matrices of $\boldsymbol{\Psi_r}$ be denoted $\mathbf{V_r}$ and $\mathbf{W_r}^{\mathrm{T}}$. The corresponding right and left eigenvectors of the generalized eigenvalue problem are given by:

$$\mathbf{V} = \begin{bmatrix} \mathbf{V_r} \\ -\mathbf{J}^{-1}\mathbf{C}\mathbf{V_r} \end{bmatrix} \tag{4.27}$$

$$\mathbf{W}^{\mathrm{T}} = \begin{bmatrix} \mathbf{W_r}^{\mathrm{T}} & -\mathbf{W_r}^{\mathrm{T}}\mathbf{B}\mathbf{J}^{-1} \end{bmatrix} \tag{4.28}$$

This approach to the generalized eigenanalysis is substantially faster than a brute force method (which would also compute the eigenvectors corresponding to infinite eigenvalues).

One additional issue we encountered in the course of implementing our identification method relates to zero eigenvalues in the system. Our system always has at least one zero eigenvalue, the familiar zero-mode from the swing-equation model. We feel comfortable neglecting this eigenvalue in our summation since none of our parameter changes should affect it or its eigenvectors. However, in practice, our linearized system often has a number of zero eigenvalues related to excitation and governor controls on generators. Since these zero eigenvalues are always identifiable by a row or column of all zeros in our matrix $\mathbf{\Psi}$, we can eliminate these variables before doing our eigenanalysis without disturbing the rest of the system.

A related problem could arise if the expected value of some parameter were zero, since this term appears in the denominator of our expression. We have yet to encounter such a case in practice, but if it were to arise, our measure would need to be modified. If only a limited number of parameters had an expected value of zero, the simplest solution would be to simply automatically identify all such parameters as "key".

4.3 Summary

This chapter focuses on prioritizing the parameter uncertainties in our reduced-order model and selecting some subset of these parameters for use in our uncertainty analyses. While we would certainly like to treat every possible system parameter as uncertain, such an analysis is simply impossible. In a standard brute force Monte Carlo approach to the problem, if most of the uncertain parameters were not critical (because those parameters were not important for the desired result or because the parameters did not vary sufficiently), the Monte Carlo results would simply converge quicker than expected without the necessity for any user intervention. However, a brute force Monte Carlo approach is far too computationally intensive to be practical and we use the probabilistic collocation method instead. With PCM, the number of system simulations necessary is strongly dependent on the number of parameters which we designate as uncertain; hence, this selection process is necessary.

The selection process proceeds in two phases. First, information about the parameters external to the study area and the associated uncertainty descriptions must be computed. These parameters are new, in that they do not exist in the original model. Direct computation of PDFs or even simple statistics for these parameters is practically impossible. Therefore, we instead prioritize the effects of the original system parameters on each of the new parameters. This method is reasonable, especially when we consider that statistical information about these parameters is only necessary for our larger parameter selection process. We discuss some topological approaches which are quite interesting; however, our actual program uses a direct computation approach to determine the necessary quantities.

Second, all of the uncertain parameters in the reduced-order model are sorted based on a measure we developed. This measure tries to determine the "key" uncertain parameters in the system. To accomplish this, it examines the weighted relative sensitivities of the system eigenvalues to each of the parameters. Our weighting is the standard deviation of the parameter divided by its expected value, which is designed to be a unitless measure of how much the parameter actually varies.

Using this information, we perform uncertainty analyses on our reduced-order model, but only treat a subset of the uncertain parameters as truly uncertain. The remaining uncertain parameters are essentially assumed to be certain. One minor complication occurs if we treat a parameter from the external reduction as uncertain. Instead of trying to derive an uncertainty description for this parameter, we simply (intelligently) select a small number of the original parameters used to calculate it and treat those parameters as uncertain. In other words, we recompute a small portion of our network reduction for each of our collocation points. This extra computation is not particularly burdensome and is much easier than trying to derive a PDF for the new parameter.

Throughout our analysis, we make the assumption that the variation of a small subset of the system parameters essentially dictates the uncertainty of the output variables of interest. We focus on such cases because our methods work well with them and they are of practical interest. As stated earlier, we also only focus on critical parameter identification methods which are simulation independent. However, for certain simulations or events, some parameter uncertainties may be critical, which are not critical in general. One example is that if a line is faulted in the system, the uncertainties associated with the parameters of that line may be critical for this event although they are not critical in general. Another potentially interesting uncertainty is the length of time until that fault is cleared, which cannot be analyzed using our eigenanalysis techniques but is obviously crucial. We explore some of these issues in the next chapter.

In closing, we note that the analysis in this chapter is specifically targeted to the overall goal of the thesis. In fact, it completes the analysis portion of the thesis, and the next chapter presents examples of our method in practice. Nevertheless, the issues and analysis in the present chapter may be of interest in other contexts. If a certain parameter is determined to be a "key" uncertain parameter, we use that parameter in our uncertainty analyses. However, another approach would be to measure or identify that particular parameter more accurately (if possible), in order to increase the accuracy of the nominal simulation. We do not explore such ideas here, since they are far afield from the main thrust of our work, but they certainly bear closer scrutiny.

Chapter 5

Examples - Both Great and Small

From the beginning, one of the major goals of our work has been to demonstrate that uncertainty analyses of power system simulations are feasible. This chapter is designed to fulfill a major part of that goal by demonstrating our method on power system models using commercially available software. The previous chapters of the thesis have described the tools necessary to implement a feasible uncertainty analysis for a large power system; in this chapter we present the results we have obtained by applying these tools.

The chapter comprises two major sections: a small power system example and a large power system example. Our methods are designed for the study of large power systems. Since present methods are capable of handling small power systems, the ability to perform analyses on large power systems is a novel feature of our work. Therefore, a demonstration of our technique on a large power system is essential. However, we cannot easily verify results we obtain on a large system model, since we have no means of testing the veracity of the results we obtain. Uncertainty analyses are not routinely carried out on such systems presently for good reason. Simulations of disturbances on our large model (932 buses and 262 generators) generally require between three to five minutes on a desktop personal computer (on the full model). Such a model obviously has hundreds or even thousands of parameters which we could model as uncertain. Even 10000 Monte Carlo simulations of this system require about a month of constant simulation without parallelization. The actual number of Monte Carlo simulations necessary to obtain accurate results is probably substantially higher.

Our solution is to also study our methods applied to a small power system model. As we discuss later, our method is not optimal for small systems, but the results we obtain can be compared to Monte Carlo simulations performed on the full (non-SME) model, which can be simulated in a reasonable time-frame. We start the chapter by discussing the small power system model and the results of applying our uncertainty analysis techniques to it,

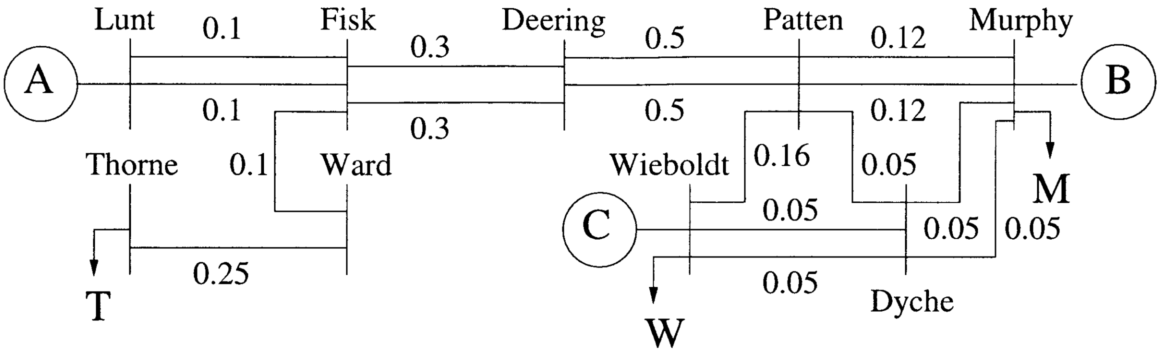


Figure 5.1: **"Small" Power System Example.** This small power system example is adapted from [56]. The transmission lines have no losses and the transmission line inductances are shown on the diagram. The arrows indicate loads and the circled letters indicate generators.

along with comparisons to accurate but time-consuming alternatives. The second section presents similar results for a large power system example but without any comparisons to other techniques.

5.1 "Small" Example

We have adapted a small power system example from [56] to test our ideas. We have modified the example from the reference in a number of ways to better suit our purposes. First, we have removed the transmission lines between Ward and Dyche and between Ward and Wieboldt; we found that the entire system was strongly coupled without these changes, which led to poor performance of the SME reduction. Second, we increased the generator inertia H of generator A to four; otherwise, generator A would not be selected as a basis generator and we would like generator A to be the sole generator in our study area. Third, we changed the transmission line parameters between Lunt and Fisk, Fisk and Deering and Deering and Patten; otherwise, the study area comprises solely Lunt and generator A, which is not very interesting for our purposes. Fourth, we increased the generation of generator A by 10MW and added a 10MW load T at Thorne; otherwise, our study area would contain no loads, and we would like to have a variety of parameters in our reduced-order model. The modified system is shown in Figure 5.1.

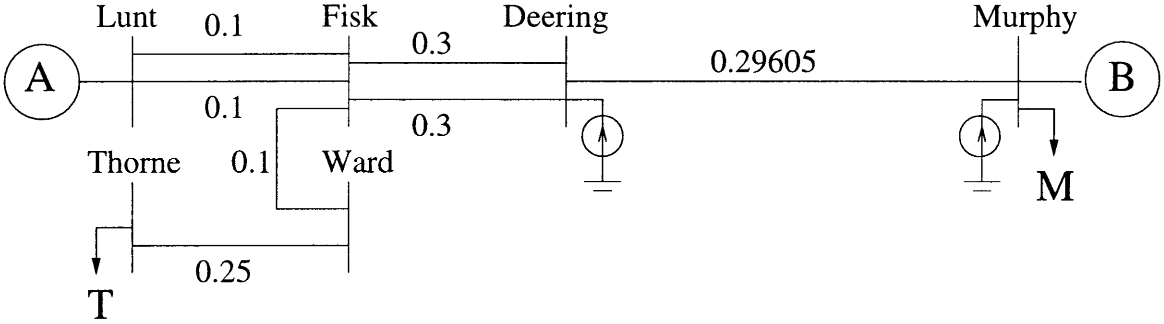


Figure 5.2: **"Small" Power System Example - SME Reduction.** The small power system example from Figure 5.1 has been reduced using SME. The circled arrows indicate controlled current sources.

5.1.1 Synchronic Modal Equivalencing

The results of applying SME to the system in Figure 5.1 are shown in Figure 5.2. The two basis generators are generators A and B. We select the area containing generator A to be our study area. The study area buses are Lunt, Fisk, Ward and Thorne. Deering is a less-relevant near node and Murphy is associated with basis generator B. A new line between Deering and Murphy is created, as well as current injectors at both Deering and Murphy. The new transmission line has a very small negative resistance associated with it along with the inductance shown on the diagram. Ten transmission lines, three buses, one load and one generator are eliminated.

Unfortunately, one limitation of using a small power system to test our ideas is that the model reduction does not make a significant difference in the simulation time for the system. In a system as small as this one, much of the simulation time is essentially "overhead", and the reduced-order model and the full model require essentially the same amount of time for any given simulation. Nevertheless, we perform the model reduction here, since we primarily seek to demonstrate the accuracy of our method with this example. The large power system example should demonstrate more effectively the time-savings of the model reduction.

As well as the system description, we also require a description of the parameter uncertainties in the system. We model the transmission line parameters and the load parameters of the full model as uncertain. Specifically, we assume that each of the transmission line inductances is equally likely to be anywhere between ±10% of its nominal value [57]. All of the transmission lines are independent, except for those in parallel which we treat as completely dependent. We model the loads using equations of the form (3.56) and (3.57). We assume that α and β are both nominally two for every load, and that the α and β of each load are always equal, but they are equally likely to be anywhere between ±25% of their nominal value. The load uncertainties are independent of each other as well as the transmission line uncertainties. The greater deviation of the loads reflects our belief that the loads have greater uncertainty in their descriptions than the transmission lines have in theirs.

5.1.2 Identification of "key" uncertain parameters

We apply our eigensensitivity approach to the reduced-order model and find that the new line between Murphy and Deering is by far the most important element. Using the measure defined in (4.24), the new line between Murphy and Deering is more than twice as "important" as the next element. Another significant gap in the rank-ordered listing does not occur until much later in the list, so we arbitrarily selected the next three most important elements: the current injector at Deering, the transmission lines between Fisk and Deering and the load T at Thorne.

Since both the line between Murphy and Deering and the current injector at Deering are created by the network reduction, we have to determine which of the original parameters have the greatest effect on them, as measured by (4.22). In both cases, the new element is most sensitive (with uncertainty weighting) to the lines between Deering and Patten. To illustrate, we present the values for the line between Murphy and Deering in Table 5.1.

Therefore, our key parameter uncertainties are all elements of the original, full model: the transmission lines between Deering and Patten, the transmission lines between Fisk and Deering and the load T. We can now proceed to analyze our system using PCM and these three parameter uncertainties. For convenience, we denote the uncertain parameters of the lines between Deering and Patten, the lines between Fisk and Deering and load T by a, b and c.

Weighted Sensitivity	Original Element
1	Deering - Patten
0.0938	Murphy - Patten
0.0658	Load W
0.0576	Murphy - Dyche
0.0226	Patten - Dyche
0.0066	Patten - Wieboldt
0.0063	Dyche - Wieboldt

Table 5.1: **Network Reduction Sensitivities for Murphy - Deering.** The results of our analysis of the external reduction for the new transmission line between Murphy and Deering. We normalize our measure such that the highest value is unity.

5.1.3 The Probabilistic Collocation Method

We begin by explicitly showing the PDF's of a, b and c:

$$f_a(a_0) = \begin{cases} 10 & 0.45 \leq a_0 < 0.55 \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

$$f_b(b_0) = \begin{cases} 16.67 & 0.27 \leq b_0 < 0.33 \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

$$f_c(c_0) = \begin{cases} 1 & 1.5 \leq c_0 < 2.5 \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

We model each output of the system y using a second-order polynomial in these uncertain parameters $\hat{y}$:

$$\hat{y} = Y_0 + Y_1 a + Y_2 b + Y_3 c + Y_4 ab + Y_5 ac + Y_6 bc + Y_7 a^2 + Y_8 b^2 + Y_9 c^2 \tag{5.4}$$

where the Y_i are constants. (We could equivalently define $\hat{y}$ using the orthogonal polynomials of a, b and c. In that case, the moments of the system could be directly computed from the Y_i. We use the present definition here for simplicity of presentation). To fit this model, we use the results of simulations performed with parameter value combinations selected from combinations of the roots of the third-order orthogonal polynomials for each of the uncertain parameters. The roots of these polynomials are presented in Table 5.2.

With the creation of our PCM polynomial model, we are ready to perform uncertainty analyses on our system. With such a small system, the computation required to create the

Uncertain Parameter	Roots
a	0.5, 0.5387, 0.4613
b	0.3, 0.3232, 0.2768
c	2, 2.3873, 1.6127

Table 5.2: **Third-Order Polynomial Roots.** The roots of the third-order orthogonal polynomials associated with uncertain parameters a, b and c are presented in the table above.

SME model, identify the key uncertain parameters and determine the collocation points for our PCM model is not significant. For example, we determine the key uncertain parameters in less than one second. To fit our polynomial model(s), we must simulate the system ten times, which by far requires the most time of all of the steps in our approach.

5.1.4 Simulations, Results and Comparisons

Our event is to short one of the lines connecting Lunt to Fisk after five seconds and then clear the fault 0.1 seconds later. To demonstrate the ability of our approach to perform uncertainty analyses of multiple output variables based on a single set of simulations, we examine the transient behavior of the internal angle and speed of generator A as well as the voltage magnitude at Lunt. Our goal is to obtain the expected value and standard deviation for each of these output variables at 100 different time points after the event. We plot these values separately as well as with our earlier "error bar" presentation.

The first set of plots in Figures 5.3, 5.4 and 5.5 are our benchmarks. We perform 1000 simulations of the full system (no SME reduction) and calculate the expected value and standard deviation of our three output variables using a brute force Monte Carlo approach. After examining the expected values and standard deviations computed using a progressively greater number of simulations, we feel that 1000 simulations give us a sufficient degree of accuracy. Of course even on such a small system, this Monte Carlo approach is time consuming; generating such figures requires over three hours of simulations.

Our second set of plots in Figures 5.6, 5.7 and 5.8 present the same outputs, but in this case we only consider the "key" uncertain parameters. We again perform Monte Carlo simulations, but we use nominal values for all of the parameters other than our key uncertain parameters. As a basis of comparison, we also present the expected values and standard

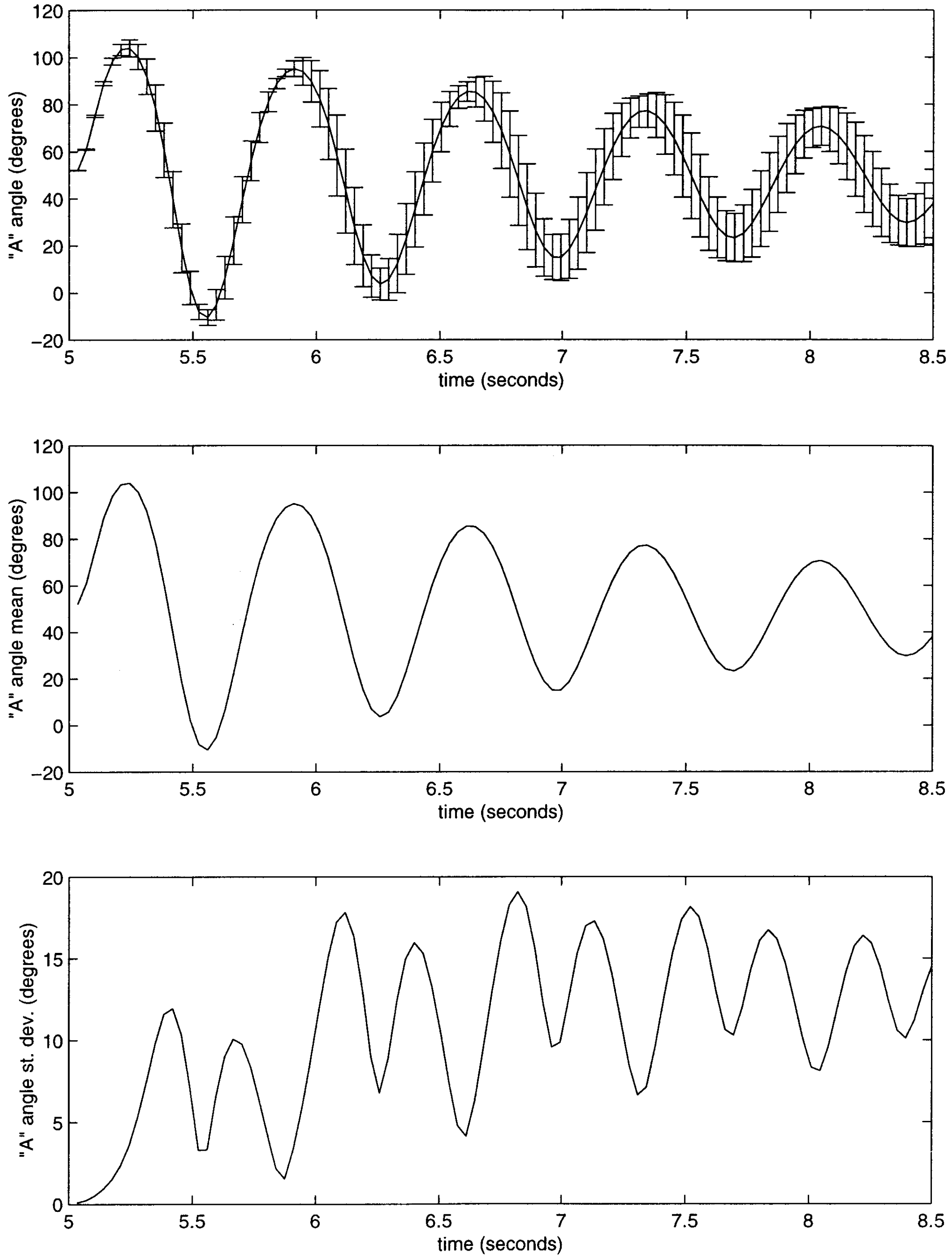


Figure 5.3: **Generator A Angle - Benchmark Uncertainty Analysis.** These results were obtained by simulating the full system model 1000 times using a brute force Monte Carlo analysis. The top figure presents the expected value and standard deviation of the internal angle of generator A. The solid line is the expected value and the "error bars" are ±1 standard deviation from the mean. The bottom two figures re-present the expected value and standard deviation separately.

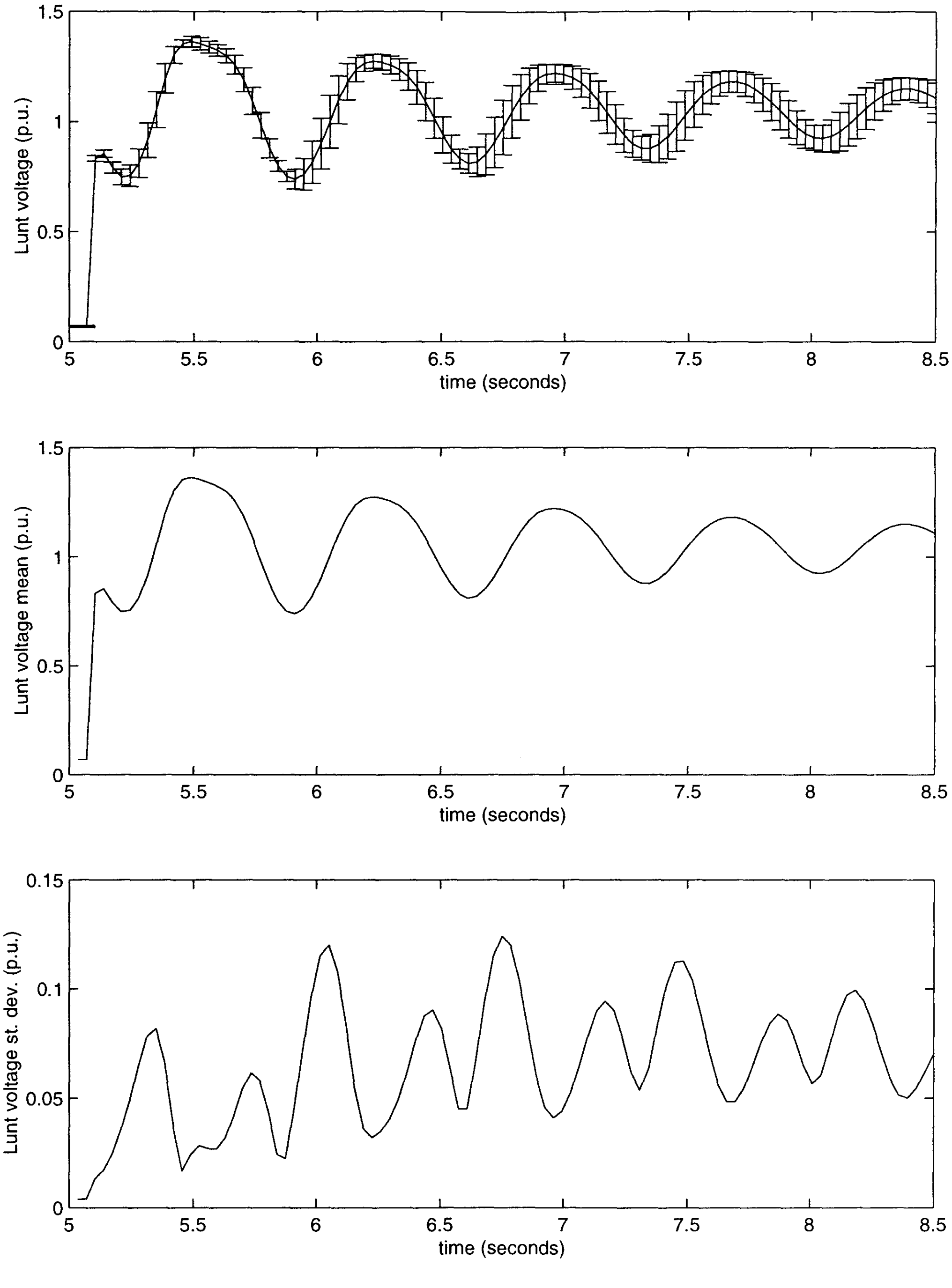


Figure 5.4: **Voltage at Bus Lunt - Benchmark Uncertainty Analysis.** These results were obtained by simulating the full system model 1000 times using a brute force Monte Carlo analysis. The top figure presents the expected value and standard deviation of the voltage at bus Lunt. The solid line is the expected value and the "error bars" are ±1 standard deviation from the mean. The bottom two figures re-present the expected value and standard deviation separately.

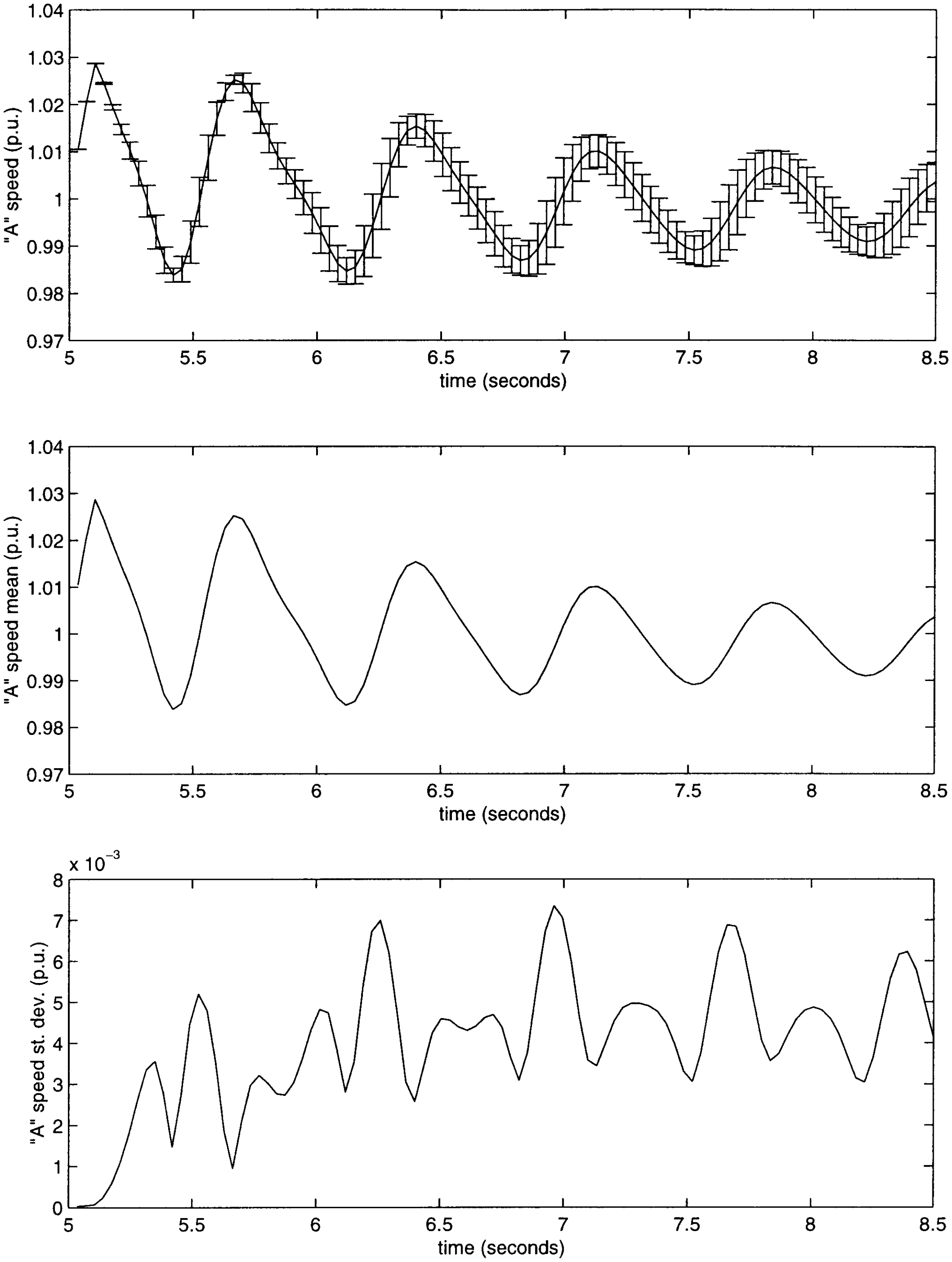


Figure 5.5: **Generator A Speed - Benchmark Uncertainty Analysis.** These results were obtained by simulating the full system model 1000 times using a brute force Monte Carlo analysis. The top figure presents the expected value and standard deviation of the speed of generator A. The solid line is the expected value and the "error bars" are ±1 standard deviation from the mean. The bottom two figures re-present the expected value and standard deviation separately.

deviations of the output variables from our benchmark analysis. The expected values have not been substantially affected and the standard deviations remain surprisingly accurate. Of course, limiting ourselves to only these key parameter uncertainties does not significantly reduce the amount of computation; roughly the same number of Monte Carlo simulations as before are necessary since the other uncertain parameters only minimally effect the outcome anyway. We remind the reader that the key parameter identification is only necessary to limit the number of simulations needed to fit our PCM polynomial model.

Our third set of plots in Figures 5.9, 5.10 and 5.11 are generated using our polynomial models based on the three key uncertain parameters. We have not yet introduced the SME reduction; the simulations used to fit our polynomial model are performed on the full, unreduced system model. We present the results of the Monte Carlo simulations based on these three key uncertain parameters as a basis of comparison. As we see here, the probabilistic collocation method performs extraordinarily well if we are primarily interested in moments of the system. For much of the transient, the computed means and standard deviations are nearly identical. The same set of ten simulations are used to fit the polynomial models for every point of all three outputs.

Our fourth set of plots in Figures 5.12, 5.13 and 5.14 present the results of our complete method. We again use PCM polynomial models based on the three key parameter uncertainties, but we fit our models using simulations of our reduced-order SME model of the power system. Other than the speed variable, the correspondence between these results and those obtained by PCM with a model fit using the full system is quite good.

The discrepancy in the speed variables is more than we would normally tolerate, especially since the expected value is obviously inaccurate. We address this issue in the next paragraph, but first we re-present our final results in Figures 5.15, 5.16 and 5.16. In these new figures, we compare the results obtained with our method to those obtained with our benchmark analysis, instead of to a partial implementation of our method. These figures demonstrate the ability of our method to reasonably approximate the actual statistics of the outputs of the system while radically decreasing the amount of computation needed to obtain the results. If we could ignore the speed variable, we would assert that our results are "good enough" for practical purposes, but we only require ten simulations of a reduced-order model to obtain all of these results.

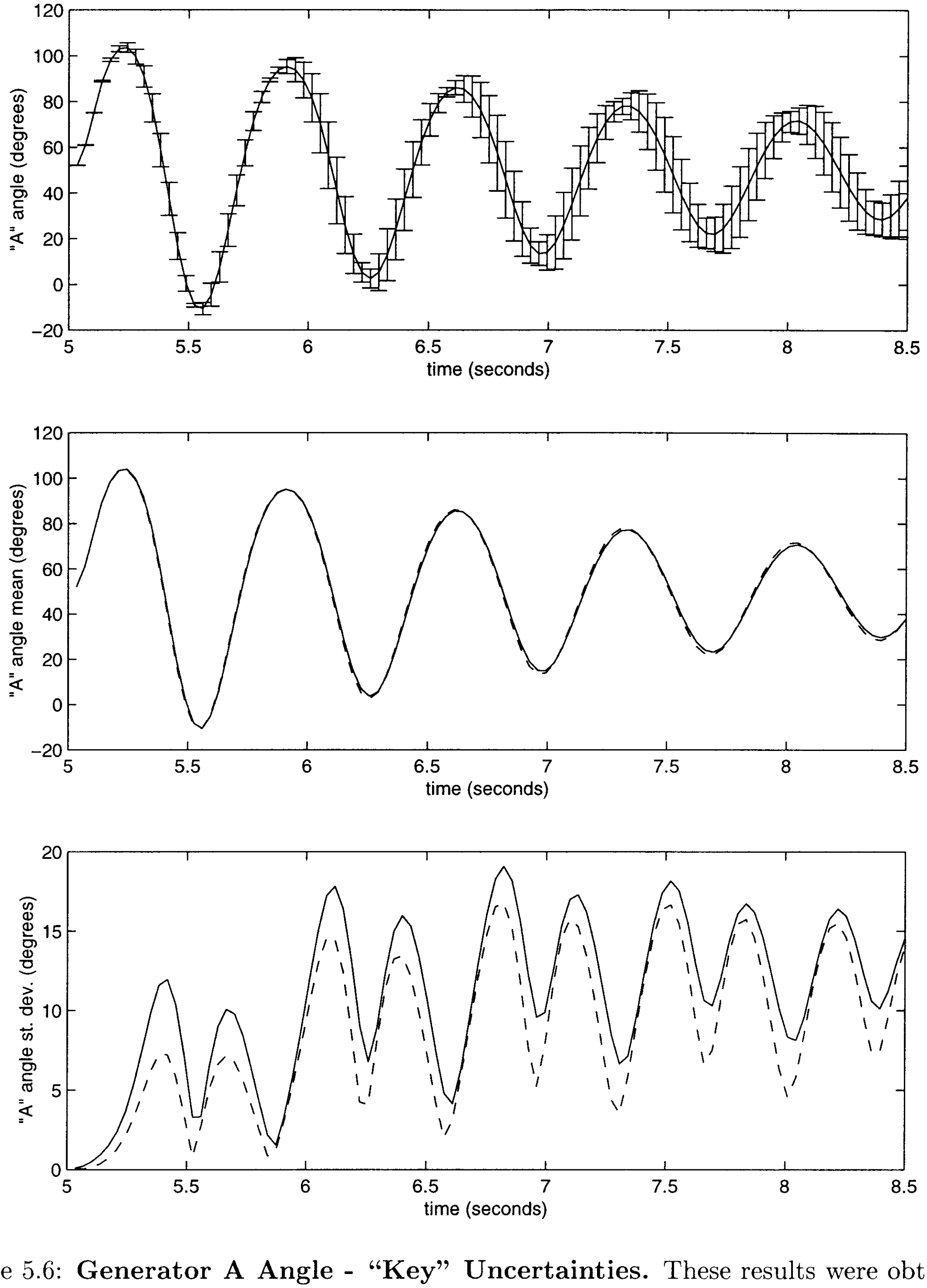


Figure 5.6: **Generator A Angle - "Key" Uncertainties.** These results were obtained by simulating the full system model 1000 times using a brute force Monte Carlo analysis, but only based on three of the uncertain parameters. The figures are similar to those for Figure 5.3. The solid lines on the bottom two figures are from the benchmark analysis, which are re-presented here for comparison purposes, while the dashed lines re-present the mean and standard deviation from the top figure.

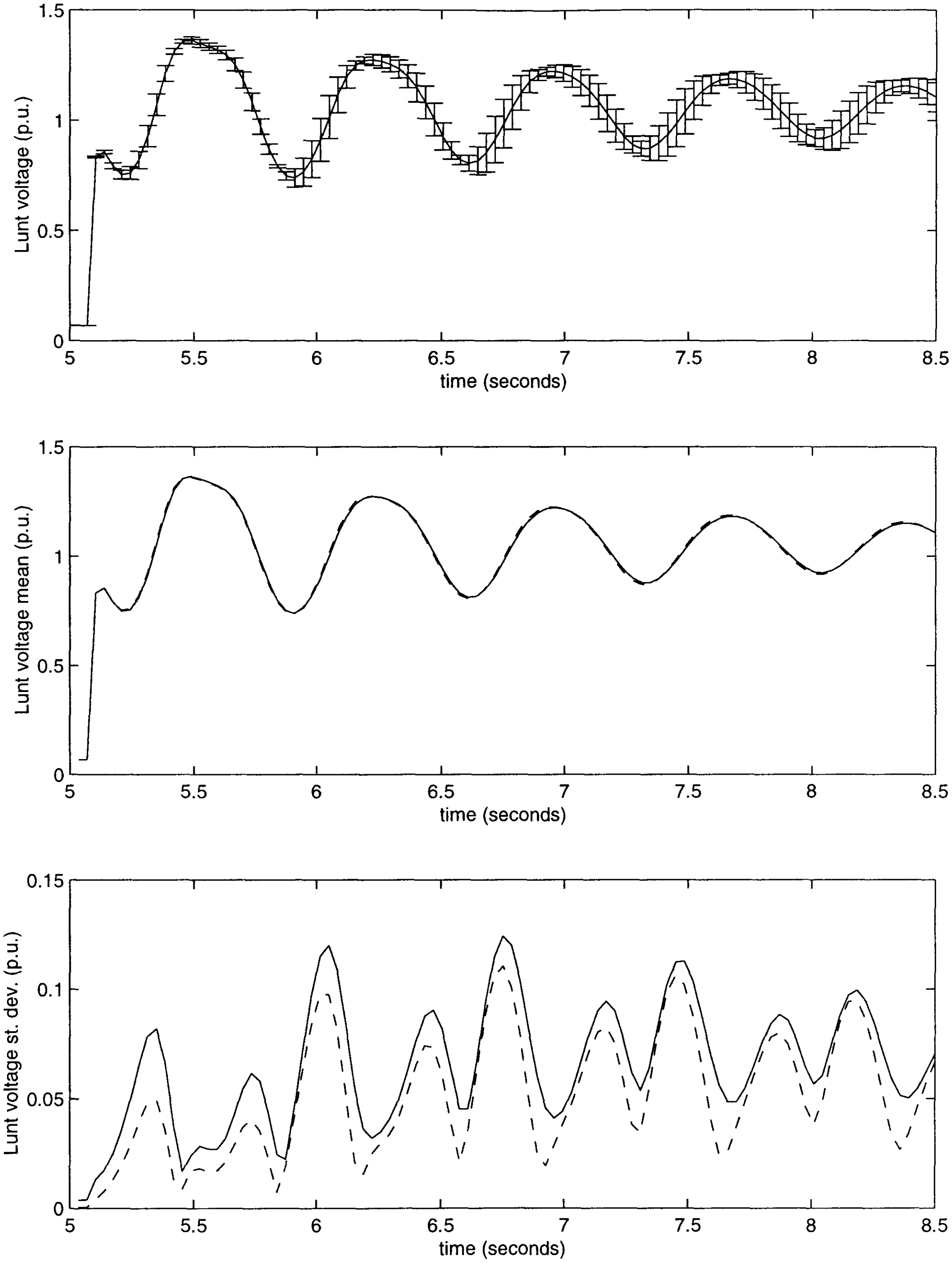


Figure 5.7: **Voltage at Bus Lunt - "Key" Uncertainties.** These results were obtained by simulating the full system model 1000 times using a brute force Monte Carlo analysis, but only based on three of the uncertain parameters. The figures are similar to those for Figure 5.4. The solid lines on the bottom two figures are from the benchmark analysis, which are re-presented here for comparison purposes, while the dashed lines re-present the mean and standard deviation from the top figure.

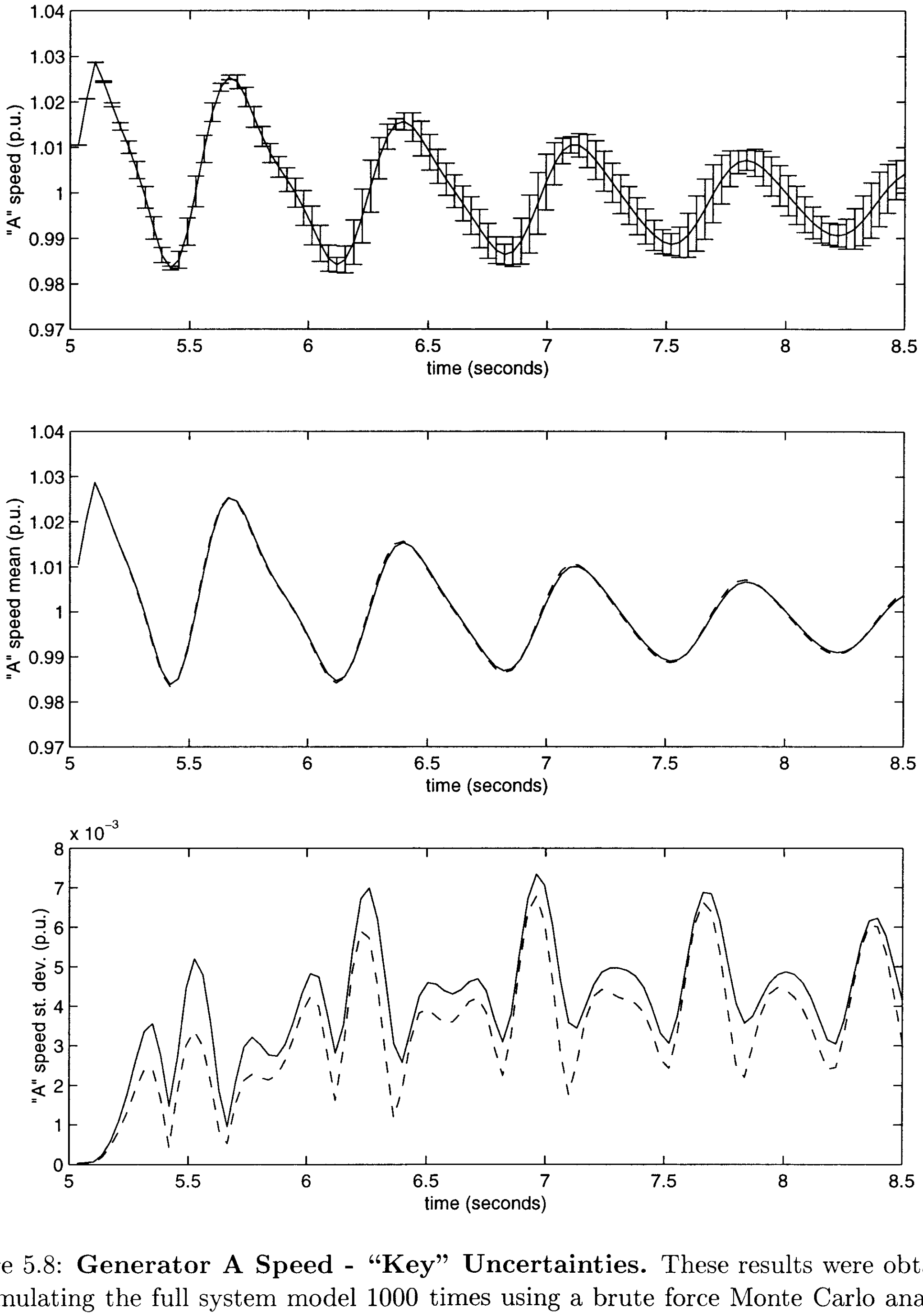


Figure 5.8: **Generator A Speed - "Key" Uncertainties.** These results were obtained by simulating the full system model 1000 times using a brute force Monte Carlo analysis, but only based on three of the uncertain parameters. The figures are similar to those for Figure 5.5. The solid lines on the bottom two figures are from the benchmark analysis, which are re-presented here for comparison purposes, while the dashed lines re-present the mean and standard deviation from the top figure.

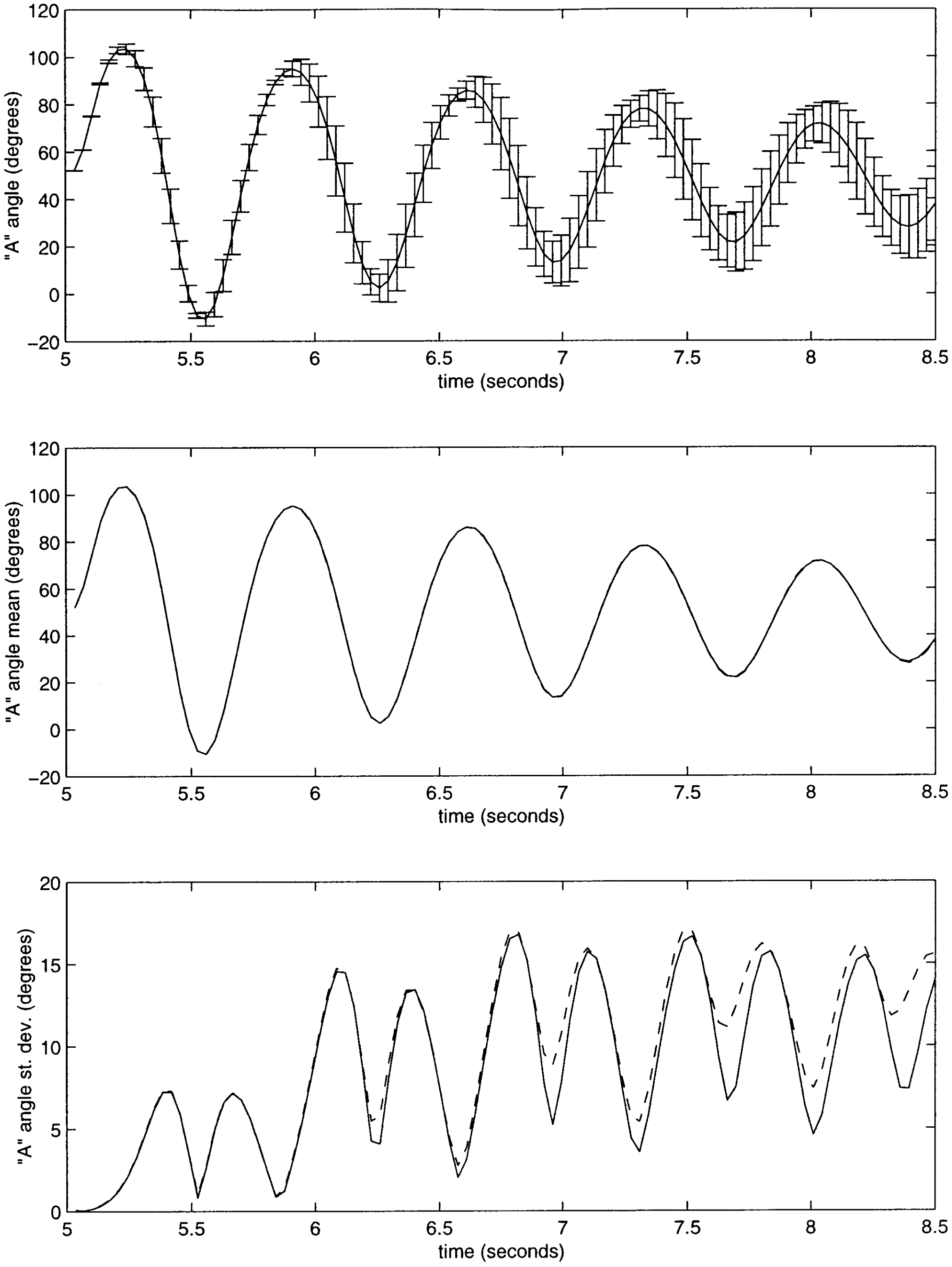


Figure 5.9: **Generator A Angle - Probabilistic Collocation Method.** These results were obtained by simulating the full system model ten times to fit a three-parameter PCM model, which is used to calculate the means and standard deviations. The figures are similar to those for Figure 5.3. The solid lines on the bottom two figures are from Figure 5.6, which are re-presented here for comparison purposes, while the dashed lines re-present the mean and standard deviation from the top figure.

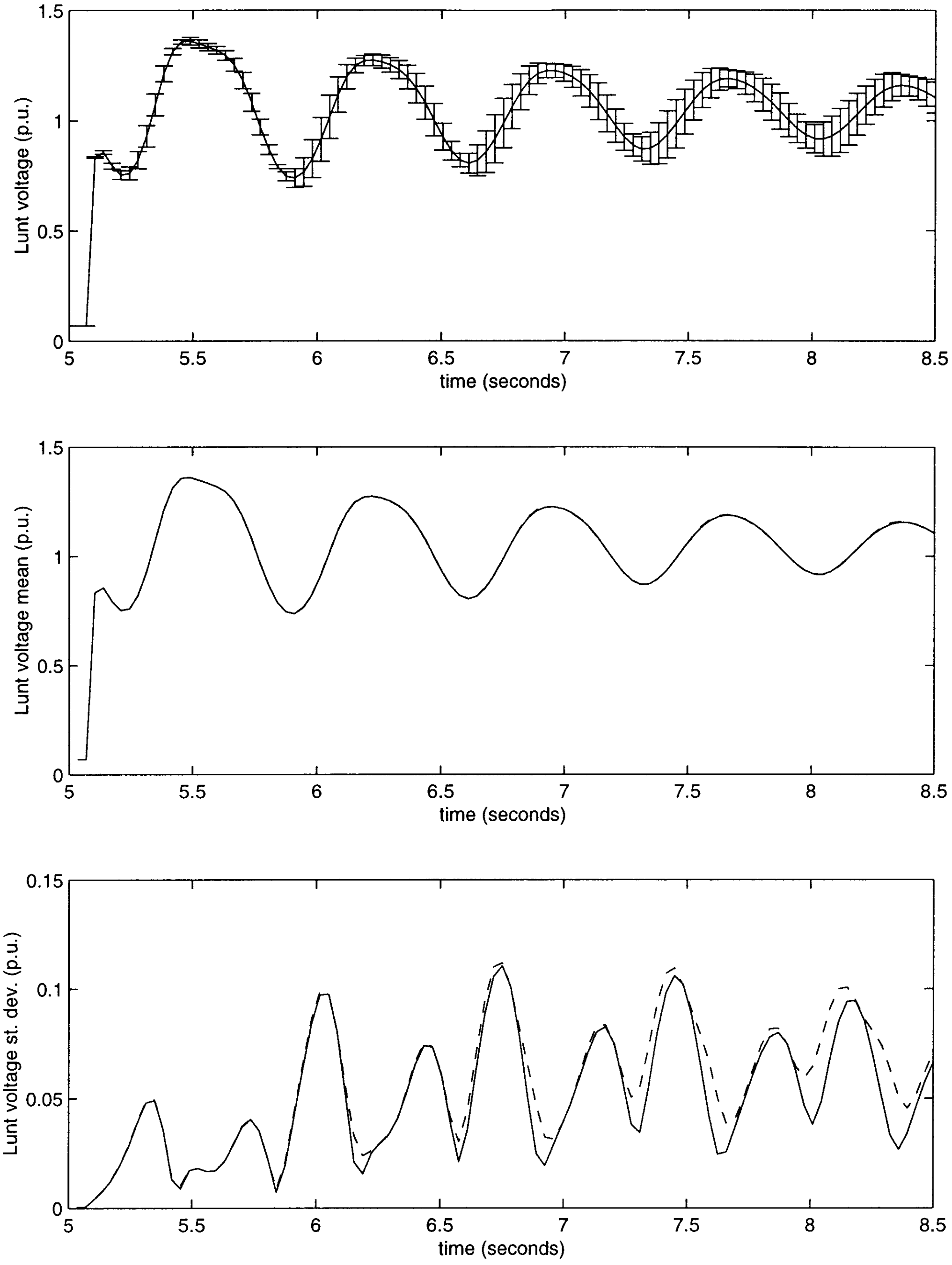


Figure 5.10: **Voltage at Bus Lunt - Probabilistic Collocation Method.** These results were obtained by simulating the full system model ten times to fit a three-parameter PCM model, which is used to calculate the means and standard deviations. The figures are similar to those for Figure 5.4. The solid lines on the bottom two figures are from Figure 5.7, which are re-presented here for comparison purposes, while dashed lines re-present the mean and standard deviation from the top figure.

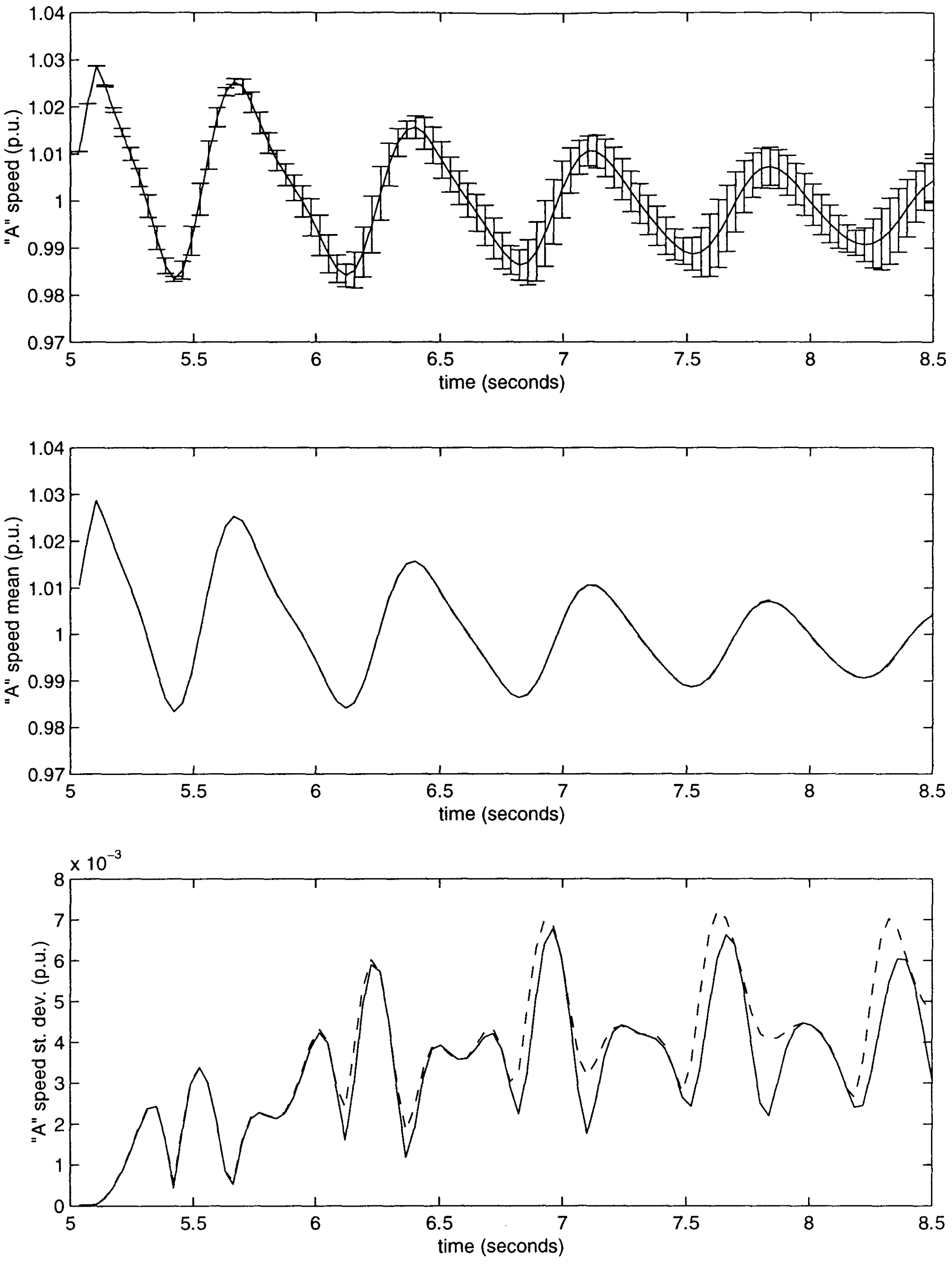


Figure 5.11: **Generator A Speed - Probabilistic Collocation Method.** These results were obtained by simulating the full system model ten times to fit a three-parameter PCM model, which is used to calculate the means and standard deviations. The figures are similar to those for Figure 5.5. The solid lines on the bottom two figures are from Figure 5.8, which are re-presented here for comparison purposes, while the dashed lines re-present the mean and standard deviation from the top figure.

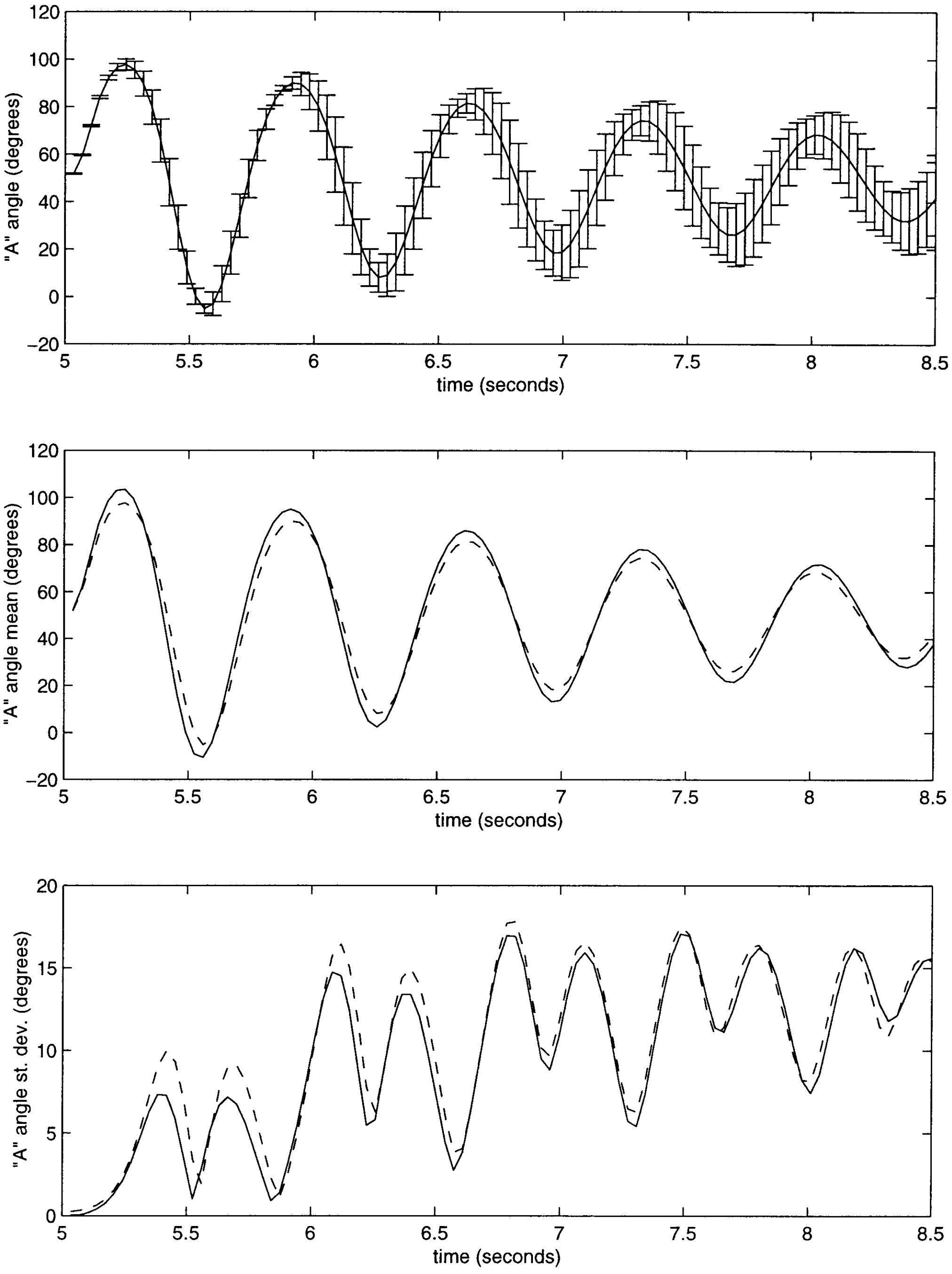


Figure 5.12: **Generator A Angle - PCM & SME.** These figures are conceptually almost identical to those for Figure 5.9. The only difference is that the probabilistic collocation method model is fit using simulations from the SME model rather than the full model. The bases of comparison (solid lines) are now taken from Figure 5.9.

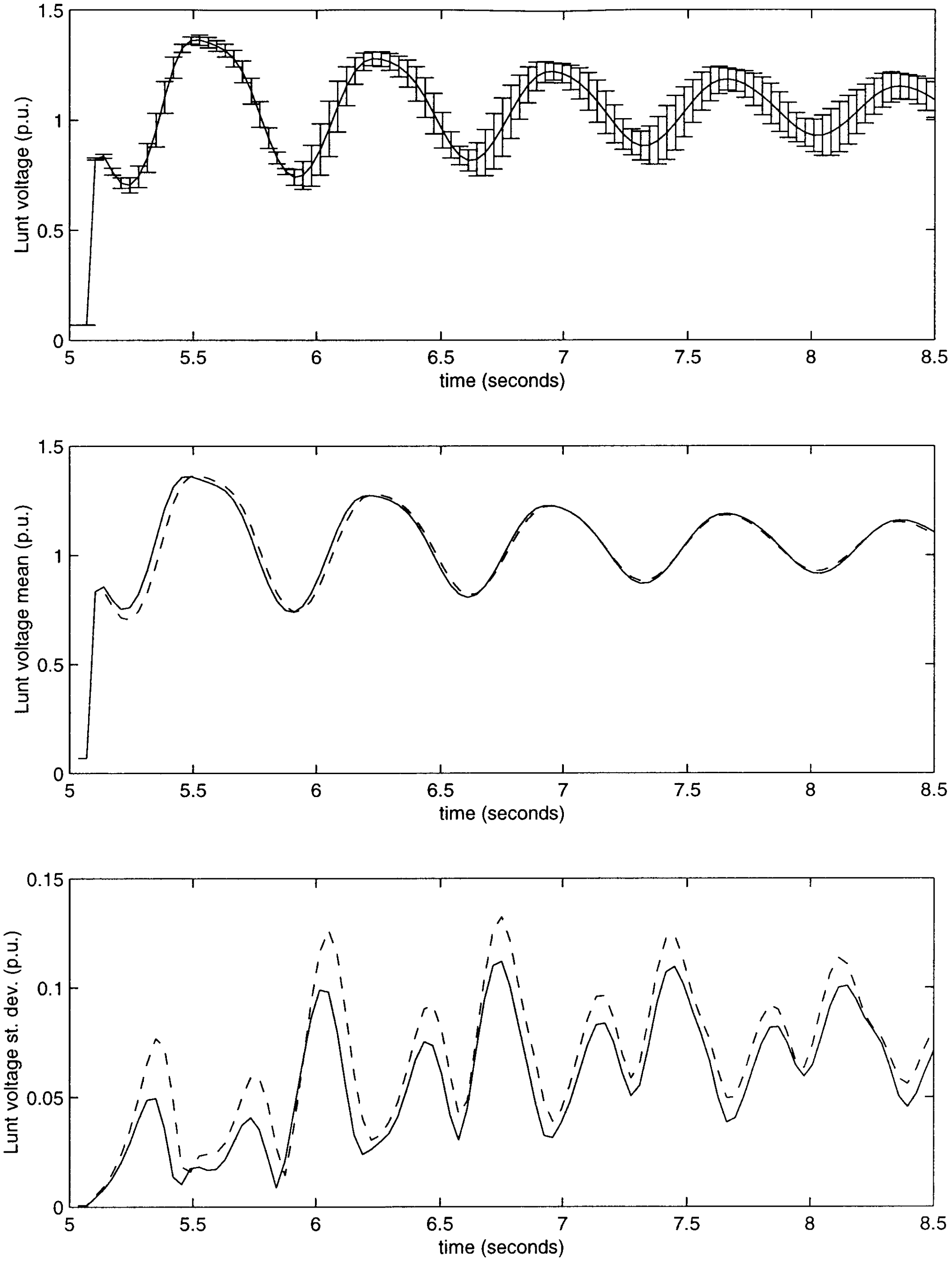


Figure 5.13: **Voltage at Bus Lunt - PCM & SME.** These figures are conceptually almost identical to those for Figure 5.10. The only difference is that the probabilistic collocation method model is fit using simulations from the SME model rather than the full model. The bases of comparison (solid lines) are now taken from Figure 5.10.

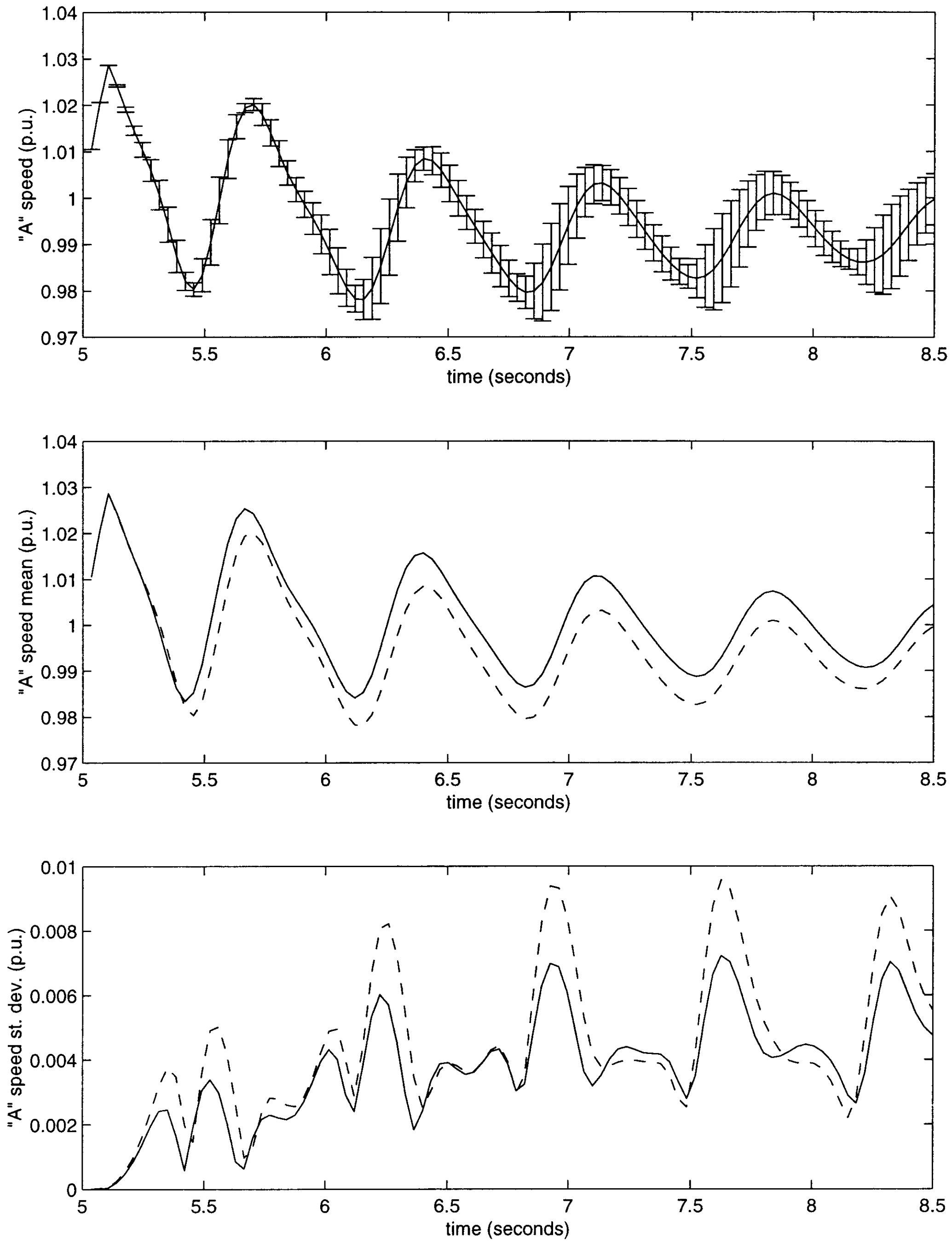


Figure 5.14: **Generator A Speed - PCM & SME.** These figures are conceptually almost identical to those for Figure 5.11. The only difference is that the probabilistic collocation method model is fit using simulations from the SME model rather than the full model. The bases of comparison (solid lines) are now taken from Figure 5.10.

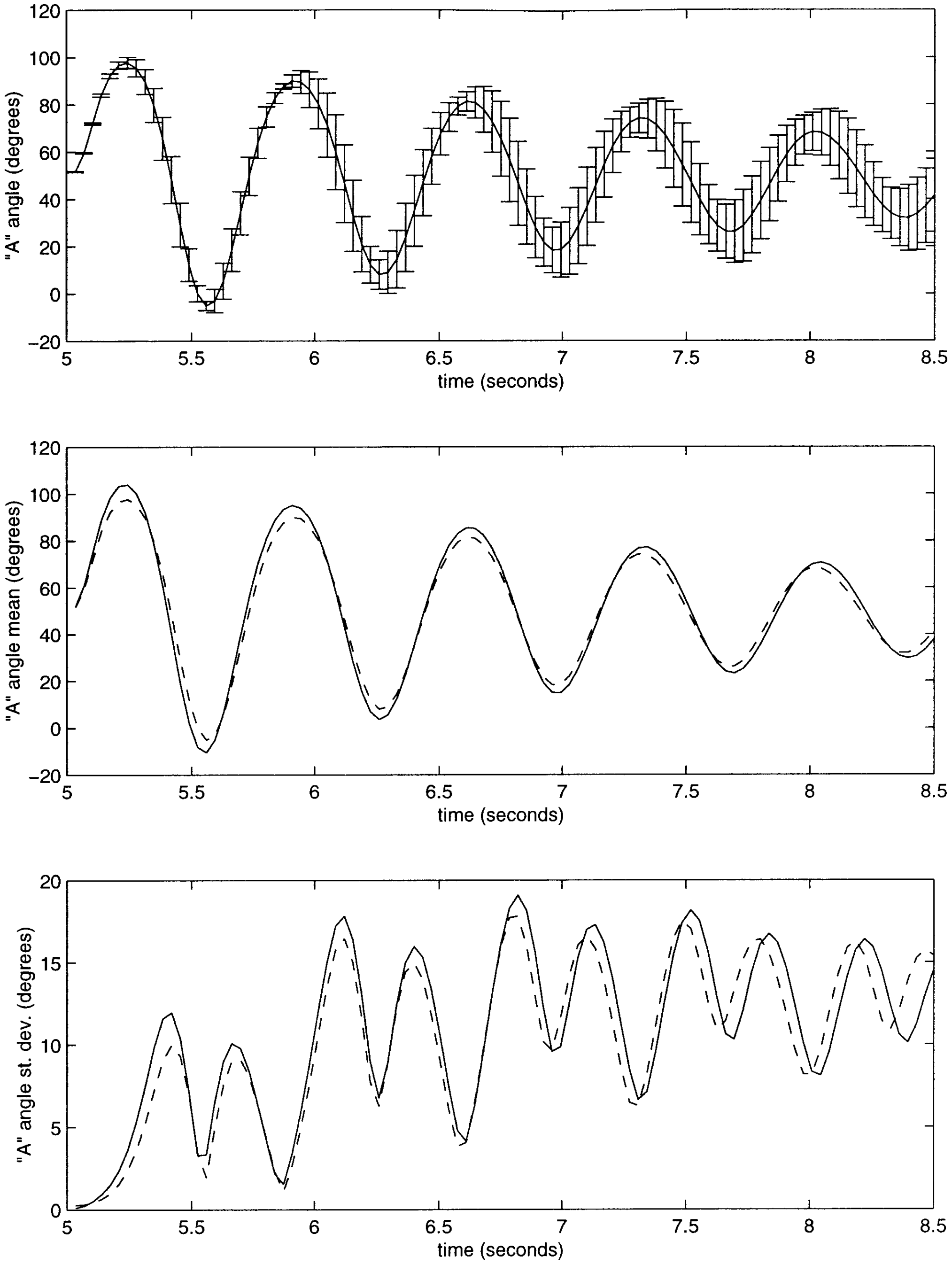


Figure 5.15: **Generator A Angle - PCM & SME (Benchmark).** These figures are almost identical to those for Figure 5.12. The only difference is that the bases for comparison in the bottom two figures are now taken from the benchmark analysis.

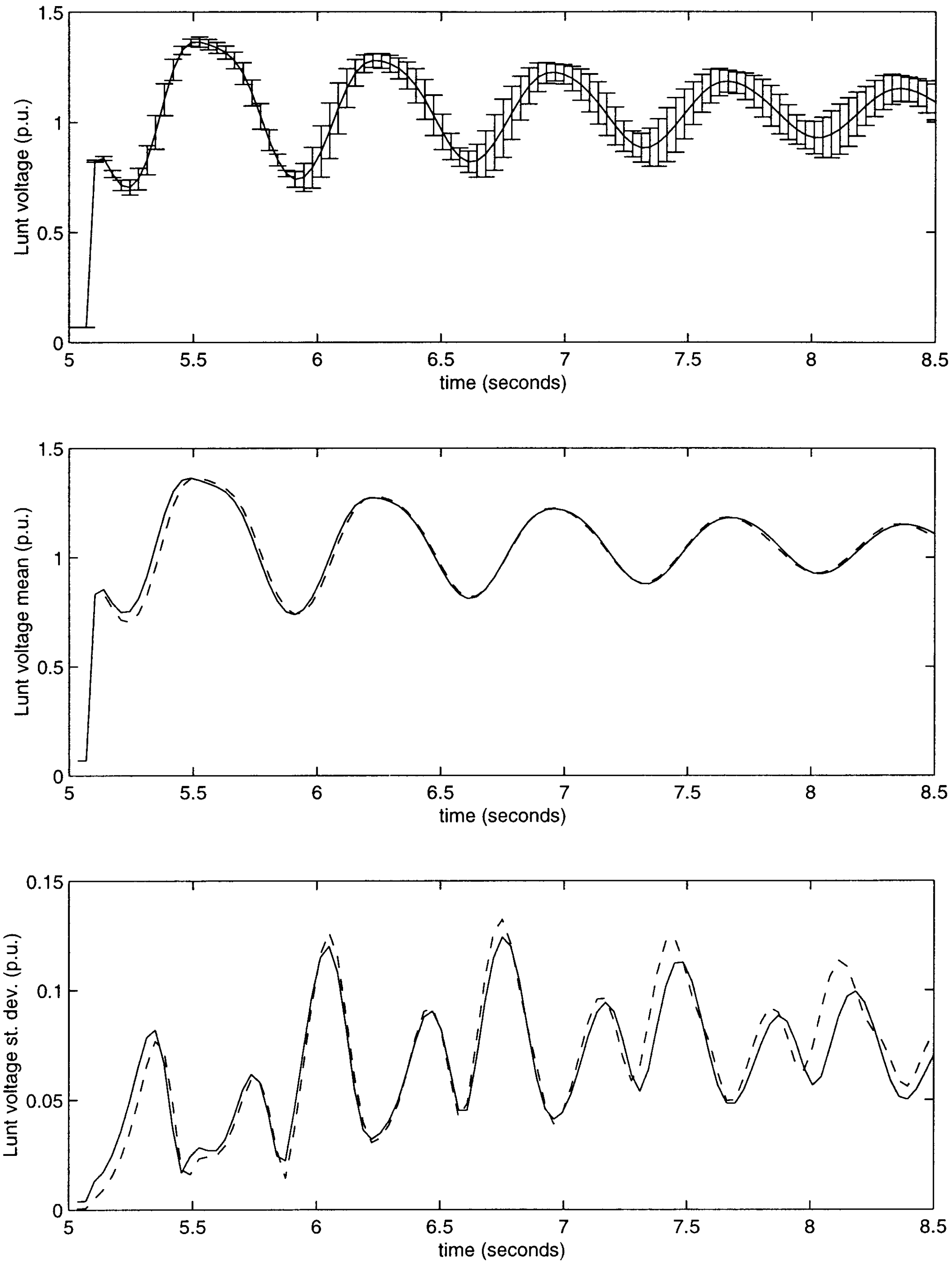


Figure 5.16: **Voltage at Bus Lunt - PCM & SME (Benchmark).** These figures are conceptually almost identical to those for Figure 5.13. The only difference is that the bases for comparison in the bottom two figures are now taken from the benchmark analysis.

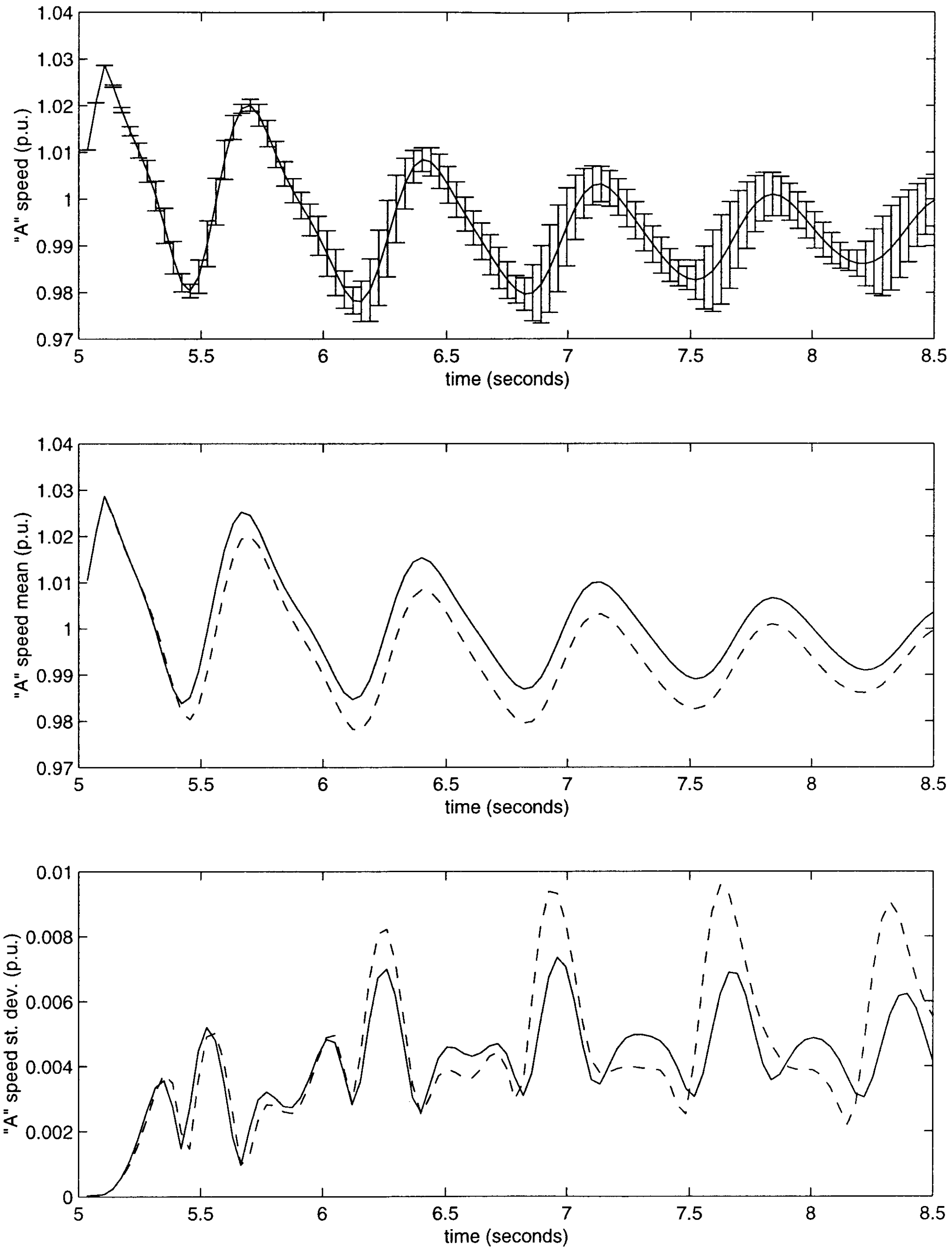
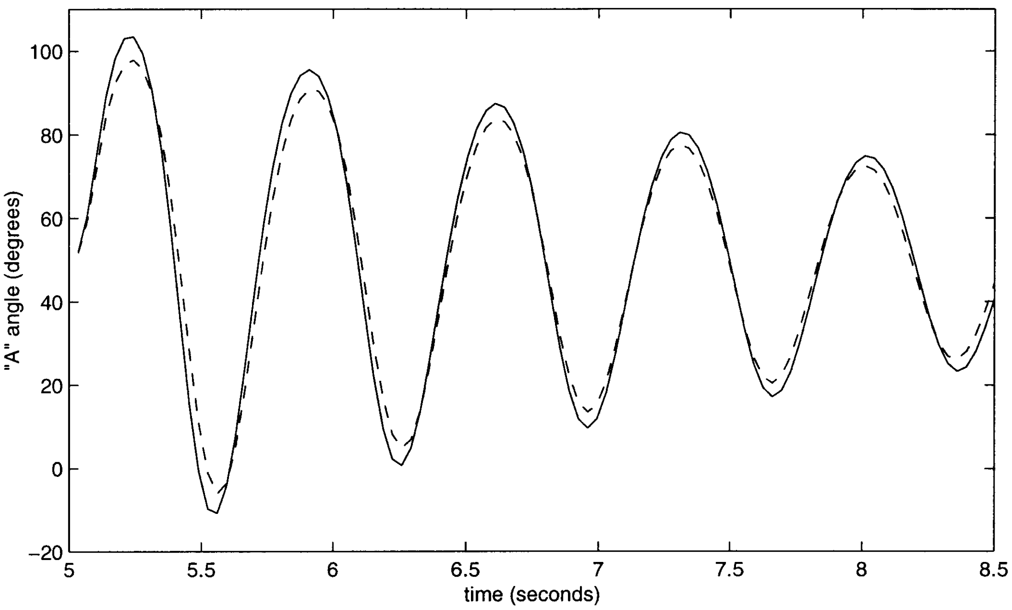


Figure 5.17: **Generator A Speed - PCM & SME (Benchmark).** These figures are conceptually almost identical to those for Figure 5.14. The only difference is that the bases for comparison in the bottom two figures are now taken from the benchmark analysis.



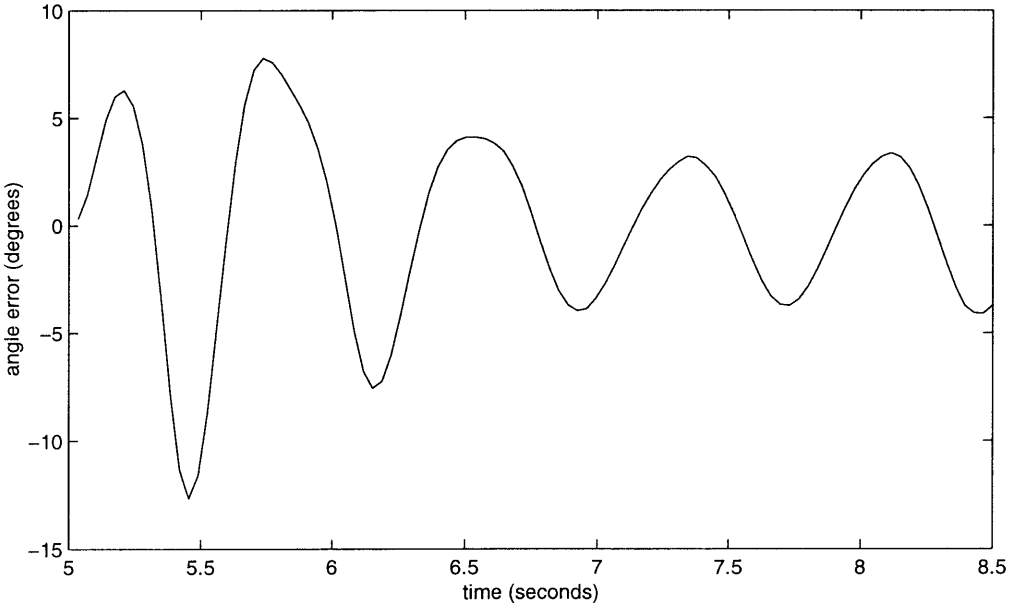


Figure 5.18: **Nominal Generator A Angle - Full Model and SME Model.** The top figure presents the internal angle of generator A in our small system example after a disturbance. The solid line is the result of simulating the full system, while the dashed line was created using our SME model of the system. The bottom figure presents the difference of the two lines in the top figure.

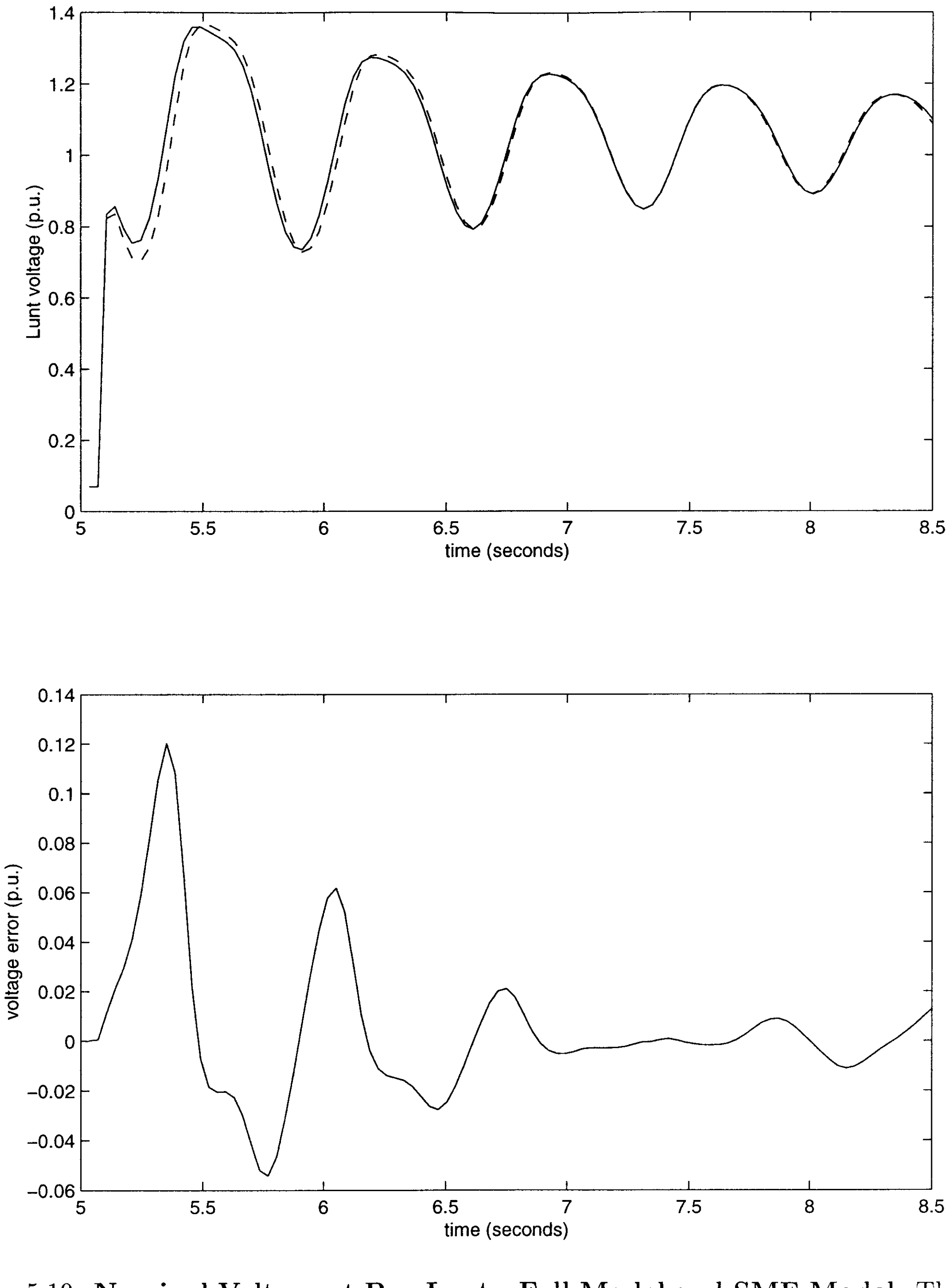


Figure 5.19: **Nominal Voltage at Bus Lunt - Full Model and SME Model.** The top figure presents the voltage at bus Lunt in our small system example after a disturbance. The solid line is the result of simulating the full system, while the dashed line was created using our SME model of the system. The bottom figure presents the difference of the two lines in the top figure.

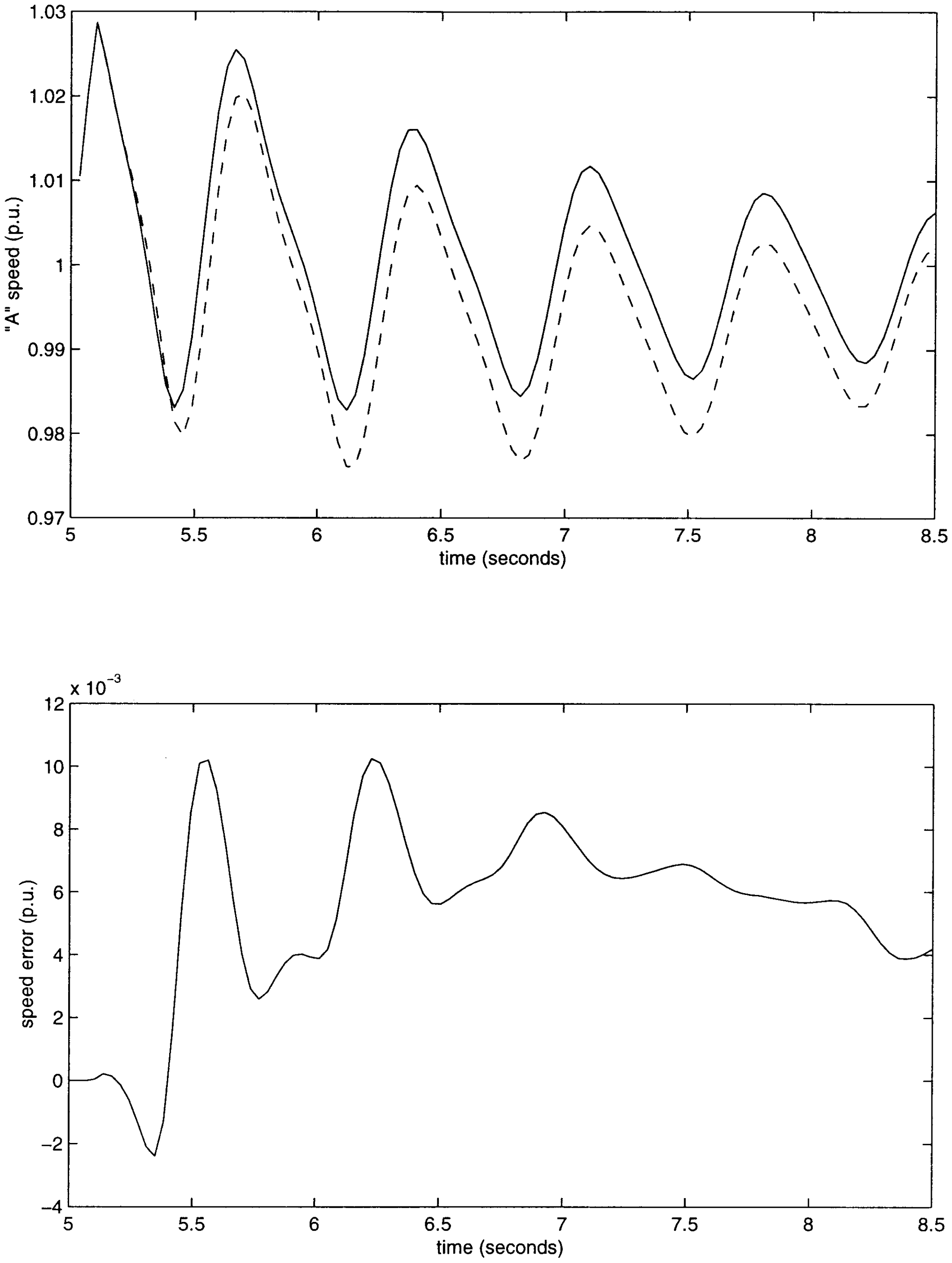


Figure 5.20: **Nominal Generator A Speed - Full Model and SME Model.** The top figure presents the speed of generator A in our small system example after a disturbance. The solid line is the result of simulating the full system, while the dashed line was created using our SME model of the system. The bottom figure presents the difference of the two lines in the top figure.

We return now to the problems presented by the results obtained for our speed variable. The discrepancy in the expected value using our method is largely due to SME, as demonstrated by Figures 5.18, 5.19 and 5.20. Even if we ignore the uncertainty in our model and just examine the nominal results of the full and reduced-order models, the agreement between them for the speed variable is not very good. After extensive testing, we conclude that this problem is inherent to SME when applied to small models. Of course, we could carefully create a small model for which SME would reliably produce accurate results, but analyzing such a system using SME is not really "fair". The problem is in essence that a typical small system violates underlying assumptions of the SME method.

Synchronic modal equivalencing assumes that portions of the system which are distant from the study area are well characterized by a linearization and the details of such distant portions are not crucial to simulations within the study area. These assumptions are made explicitly but also implicitly by our assumption that the system modes can be separated into local and extensive modes. In a system as small as the one in this section, these assumptions are not really valid. No portion of the system could truly be considered to be far from the study area. As SME has been developed, these assumptions have not been stressed because we typically do not test SME on such small models. The reasons for this are practical; if the system model is already small, there is little reason to sacrifice any accuracy to gain an incremental improvement in simulation time. Therefore, we feel that these results do not seriously undermine the validity of our approach.

5.2 "Large" Example

We have obtained a dynamic model which describes a large portion of the western United States. Unfortunately, the data is in a different file format than that used by EUROSTAG and because of differing modeling assumptions the conversion to EUROSTAG format is nontrivial. Therefore, we have been studying the application of synchronic modal equivalencing and the methods described in this thesis to a portion of this system. However, it is important to note that we have substantially modified the data files and system models during the conversion process; the system we are studying is not necessarily an accurate representation of the original system. This large network has 262 generators, 932 buses, 1355 transmission lines (including transformers) and is described by 3060 state variables

	Unreduced	Reduced
# of Gens./Loads	262/670	16/63
# of Buses Removed	-	800
# of Lines Removed/Created	-	1209/282
# of Injectors	-	49
Simulation Time (seconds)	224	34

Table 5.3: **Large Network SME Reduction.** This table presents a numerical summary of some of the salient features of the SME reduction of our large example network.

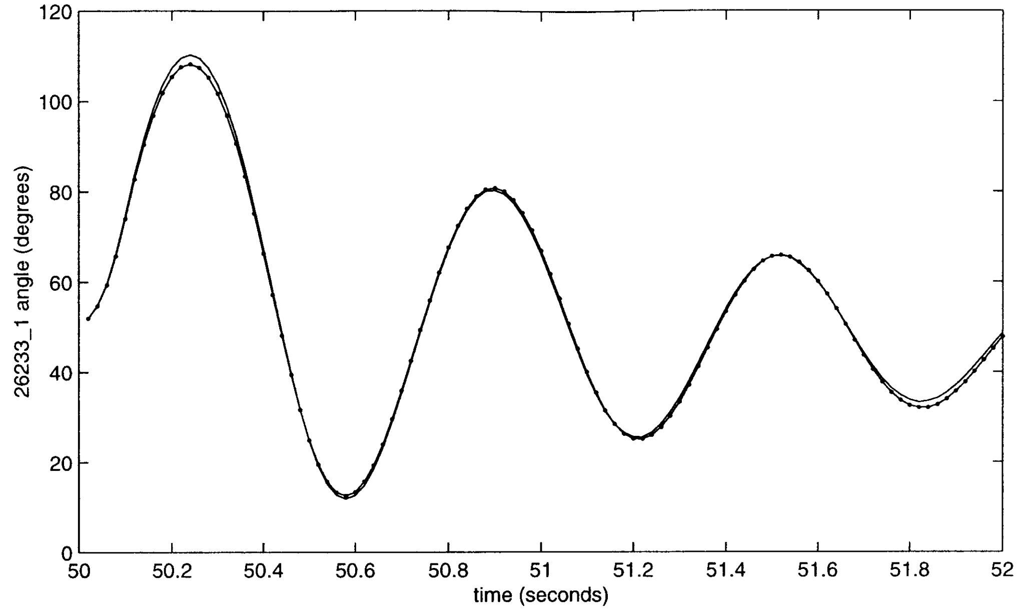
and 2933 algebraic variables.

5.2.1 Synchronic Modal Equivalencing

For our study, we choose a seven-area partition of the system. Unfortunately, we do not have a detailed map of the system, so it is difficult to visualize the partition. We do present a numerical summary of the equivalencing process in Table 5.3. Our event is a short-circuit between nodes 26234NQI - 26448NQI after 50 seconds which we clear 0.1 seconds later; this line is the only direct connection between two groups of generators, and, consequently, faulting the line has a substantial effect on the system. We select the study area to be the group which contains these two nodes. For this event and study area, the reduced-order model simulations are more than six times faster than the full model simulations.

The agreement between our full model and the reduced-order model is striking considering the computational savings of the reduced-order model. Figures 5.21, 5.22 and 5.23 show a comparison of several different output variables for the full and reduced-order models. (The generator and bus used are adjacent to the line which is short-circuited).

Naturally, generating the reduced-order model also requires computation and time. In this example, a reduced-order model can be generated in approximately eight minutes (the procedure requires some user input, so exact times are not appropriate here). The most time-consuming portion of the process is the initialization of the algorithm and extraction of the necessary data from the EUROSTAG output files; this process requires more than five minutes. If a different partition or a different study area is needed for the same system, the five minutes of data preparation do not need to be repeated and this time can be amortized over multiple model reductions.



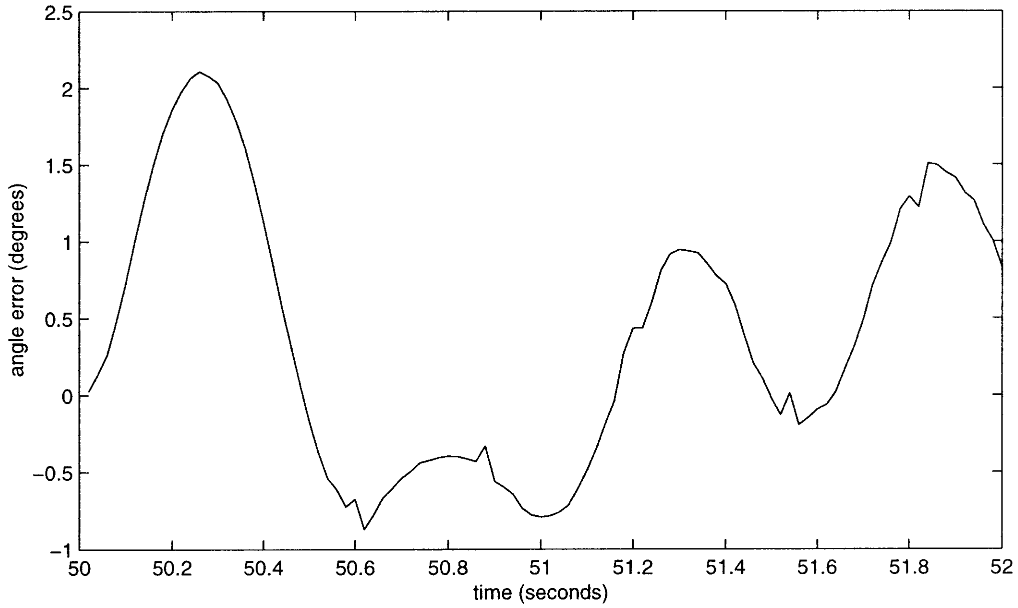
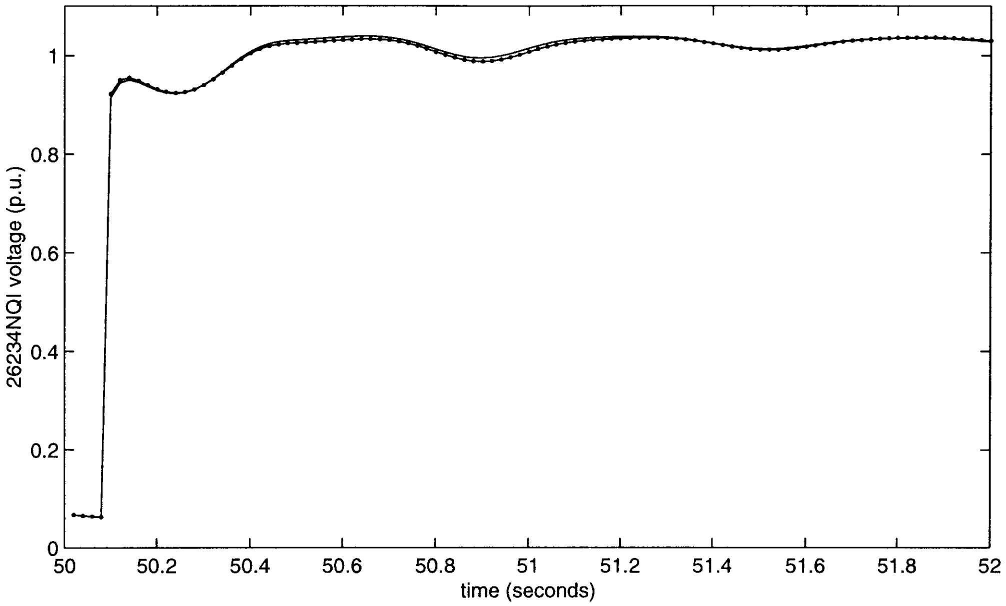


Figure 5.21: **Nominal Generator 26233_1 Angle - Full Model and SME Model.** The top figure presents the internal angle of generator 26233_1 in our large system example after a disturbance. The solid line is the result of simulating the full system, while the line with dots was created using our SME model of the system. The bottom figure presents the difference of the two lines in the top figure.



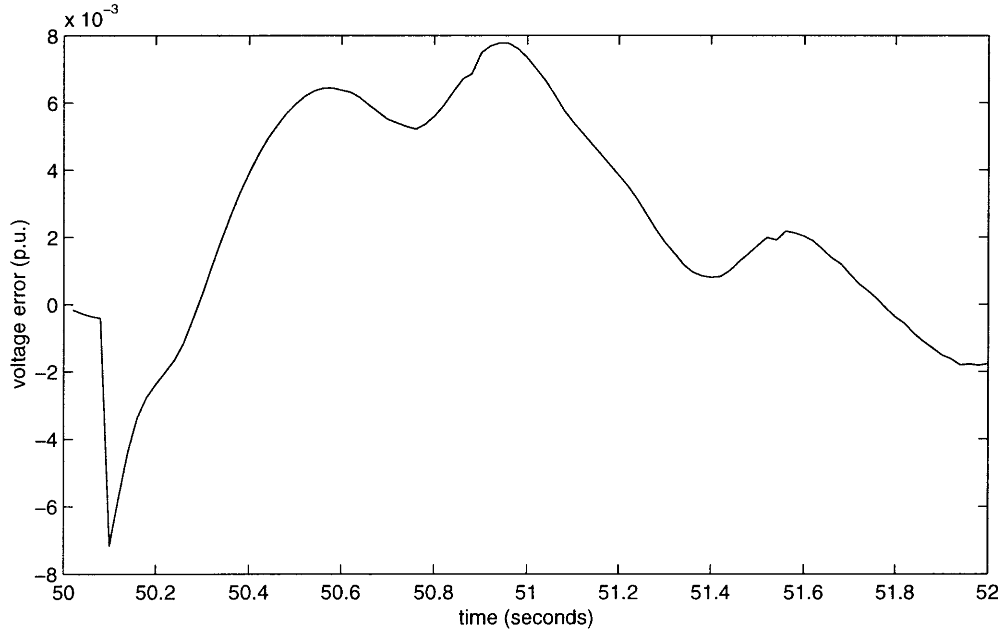


Figure 5.22: **Nominal Voltage at Bus 26234NQI - Full Model and SME Model.** The top figure presents the voltage at bus 26234NQI in our large system example after a disturbance. The solid line is the result of simulating the full system, while the line with dots was created using our SME model of the system. The bottom figure presents the difference of the two lines in the top figure.

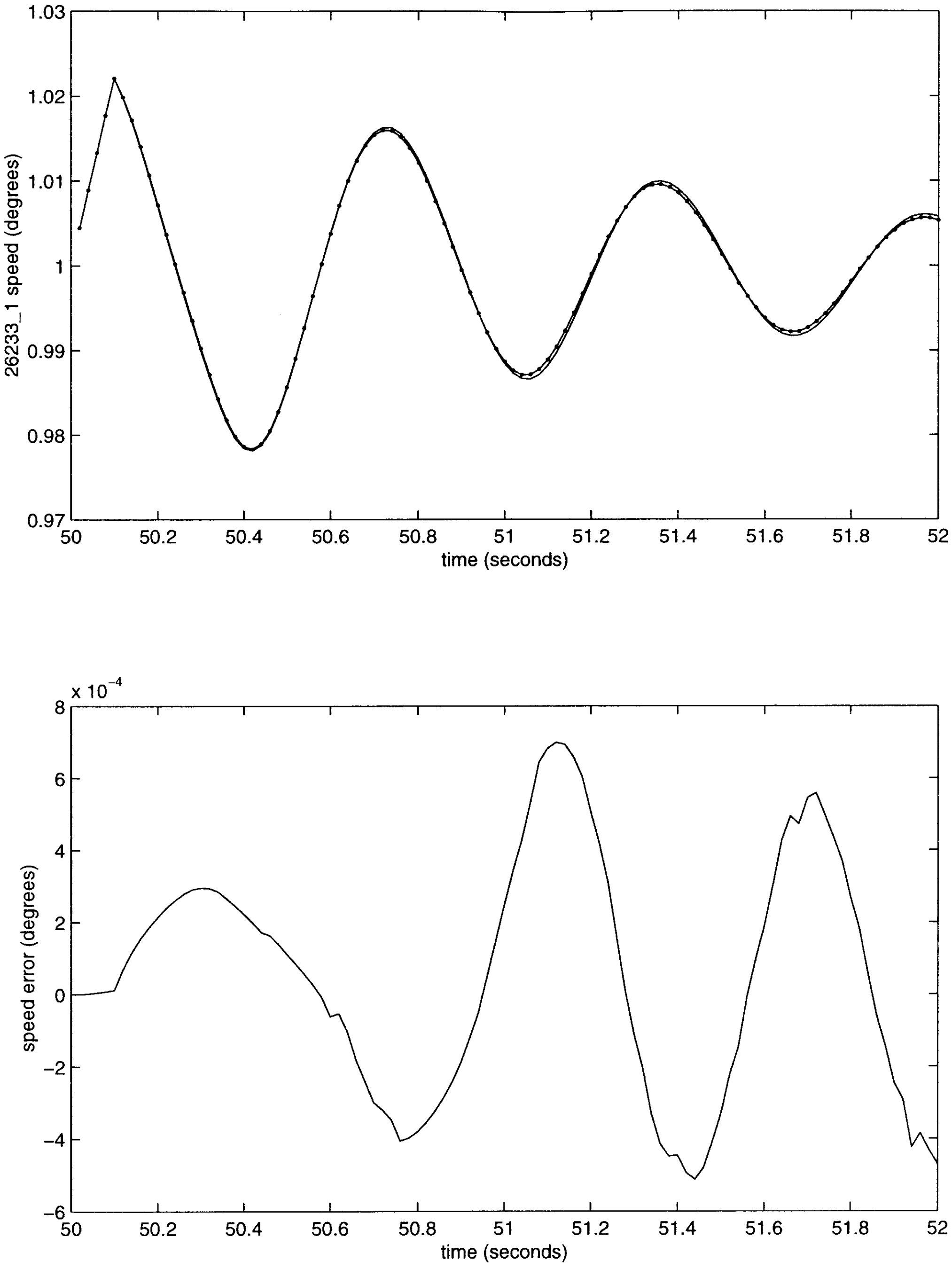


Figure 5.23: **Nominal Generator 26233_1 Speed - Full Model and SME Model.** The top figure presents the speed of generator 26233_1 in our large system example after a disturbance. The solid line is the result of simulating the full system, while the line with dots was created using our SME model of the system. The bottom figure presents the difference of the two lines in the top figure.

5.2.2 Key Uncertainties

Initially, we use similar uncertainty modeling assumptions as we did for our smaller example. We only model the lines and the exponents of the load models as uncertain, with the load model parameters having greater uncertainty than the line parameters. We again use uniform distributions.

Unlike the small model, the large model presents difficulties for our key uncertainty identification procedure. We have just begun to explore the application of SME to a model of this size. Previously, we had only worked with models which were at least an order of magnitude smaller than this large network. The larger model presents many challenges; one of the most important challenges for us is the connectivity of the study area. The study area we selected for the analysis in this section comprises two large unconnected pieces and eight other smaller pieces. An unconnected study area presents at least two problems. First, we believe that SME reduced-order models perform well for events which are firmly situated within the study area; in other words, if the event is near the edge of the study area, the reduced-order model may not perform well. The reasons for this are similar to those which we discussed for the small model. If the study area is fragmented, many of the lines and generators within the study area are necessarily near to nodes and generators external to the study area. Second, our key parameter uncertainty identification procedure is global, in that it is not optimized for any particular event. However, with a poorly-connected study area, the event of interest may be quite distant from some other elements within the study area. These distant elements may be important overall, but they may not substantially affect an event sufficiently distant from them.

Applying our eigenvalue sensitivity approach to this reduced-order model, we see that a transmission line has the greatest weighted sensitivity. One other transmission line and a current injector have weighted sensitivities within 50% of this greatest value. We select these two lines and the original element which has the greatest effect on the current injector (a load) as our key uncertain parameters. Unfortunately, all of these elements are quite distant (topologically) from our event, and they only minimally affect our output variables (as verified by simulations). Reasoning that the lack of connectivity in the study area may be adversely affecting our results, we also select a number of parameters near the event of interest and also study the effect of their uncertainty on the simulations; the extra effect is marginal.

Our system appears to be relatively insensitive to parameter uncertainties. However, we have not yet explored uncertainties in parameters which are intrinsic to the simulation event itself, which is a possibility we briefly mentioned in the conclusion of Chapter 5. A good example of such a parameter in this particular case is the length of time until the fault is cleared. Instead of using a nominal clearing time of 0.1 seconds, we assume that the clearing time is uniformly distributed between 0.0333 and 0.1 seconds. We experimentally verified that this parameter uncertainty dominates in our example (its effect is an order of magnitude greater than the combined effect of our "key" parameter uncertainties); therefore, we use a single parameter PCM polynomial model.

For completeness, we note that the key parameter uncertainty identification requires approximately thirty minutes. The data preparation requires less than one minute, and the analysis of the eigenvalue sensitivities only requires two minutes. The bulk of the time is spent analyzing the external reduction. This time could be substantially reduced if we first rank-ordered the system parameters using unweighted eigenvalue sensitivities. Such an analysis would indicate which of the new elements created by the external reduction are likely to be important and we could limit our analysis of the external reduction to just those elements, which would substantially reduce the amount of computation. We did not implement this approach here since we want to present a worst-case computation.

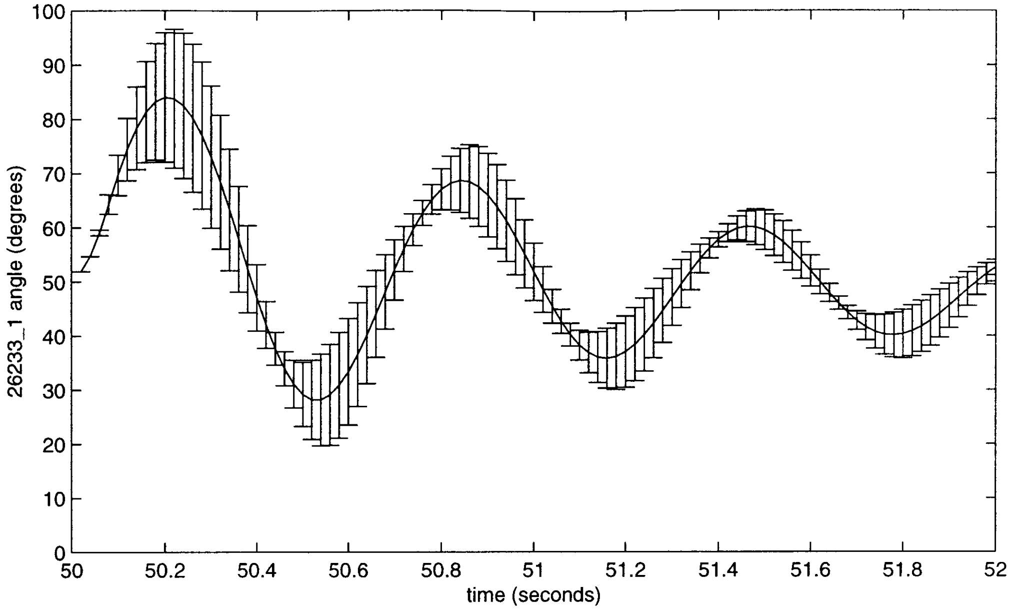
5.2.3 PCM, Simulations and Results

Since we are only using a single parameter PCM polynomial model here, it is straightforward to explore polynomials of increasing order and use one that is suitably accurate. In our case, we begin with a second-order polynomial and explore polynomial models up to sixth-order. (The quadratic model requires three simulations to identify whereas the sixth-order model requires seven simulations). We present the results obtained using the third-order model here. The standard deviation is slightly different at some time points when the third-order model is used as opposed to the quadratic model. The higher-order models yield almost identical results to those obtained using the third-order model (when we examine only the moments). Figures 5.24, 5.25 and 5.26 contain the relevant results. The expected values and standard deviations are obtained from our PCM polynomial models, which were identified using simulations of the SME reduced-order model.

The results for the speed and angle variables are quite promising. The uncertainty introduced by our parameter uncertainty is substantially larger than the error introduced by SME; therefore, we assert that the error introduced by the SME model is small enough to justify using the reduced-order model with its substantially faster simulation time. The situation for the voltage variable is not as clear. The uncertainty in the initial part of the transient (before approximately 50.3 seconds) is clearly larger than the error introduced by SME; however, as the transient continues, the error and the uncertainty often have roughly equal magnitudes. Perhaps it is unrealistic to expect SME to reproduce (with sufficient accuracy) these smaller dynamics after the initial large voltage swing. In this example, the error is only unacceptably large because the uncertainty in parts of the transient is relatively small. Further testing of SME on large models is planned (and warranted); we hope that further study will help us identify output variables which may have unacceptably large error (for whatever reason) for a given simulation.

As a final comparison, we present Figures 5.27, 5.28 and 5.29. We can see from these comparisons of PCM results based on the full model and PCM results based on the reduced-order model that using the reduced-order model does not significantly affect the accuracy of the speed and angle outputs. Although the error for the expected value of the voltage output is apparent, we do note that the standard deviation is not as adversely effected by the reduction.

We stress here that these results are obtained with a minimum of computational effort. The analysis necessary to perform the model reduction and key parameter identification require less than forty minutes, even in the worst-case scenario. The simulations necessary to identify the PCM polynomial model require approximately two minutes. The time required for the identification and analysis of the polynomial model (including finding moments) is on the order of seconds. We summarize the time-savings in Table 5.4.



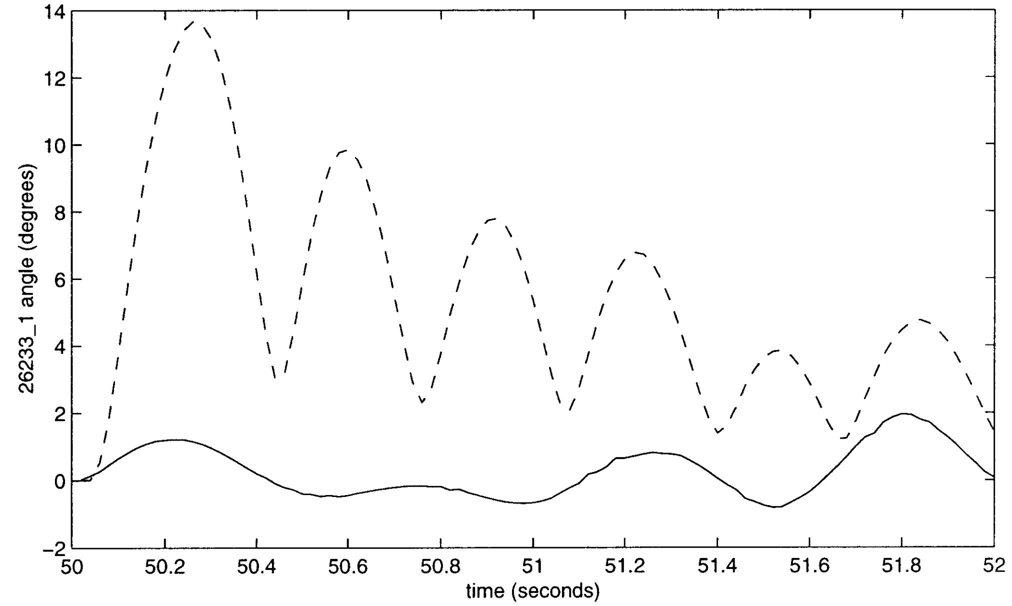


Figure 5.24: **Generator 26233_1 Angle - Uncertainty.** The top figure presents the expected value and standard deviation of the angle of generator 26233_1 in our large system example after a disturbance, where the solid line is the expected value and the error bars are ±1 standard deviation from the mean. The bottom figure presents the standard deviation (dashed line) as well as the difference between a nominal simulation of the full model and a nominal simulation of the reduced-order model (solid line) for comparison.

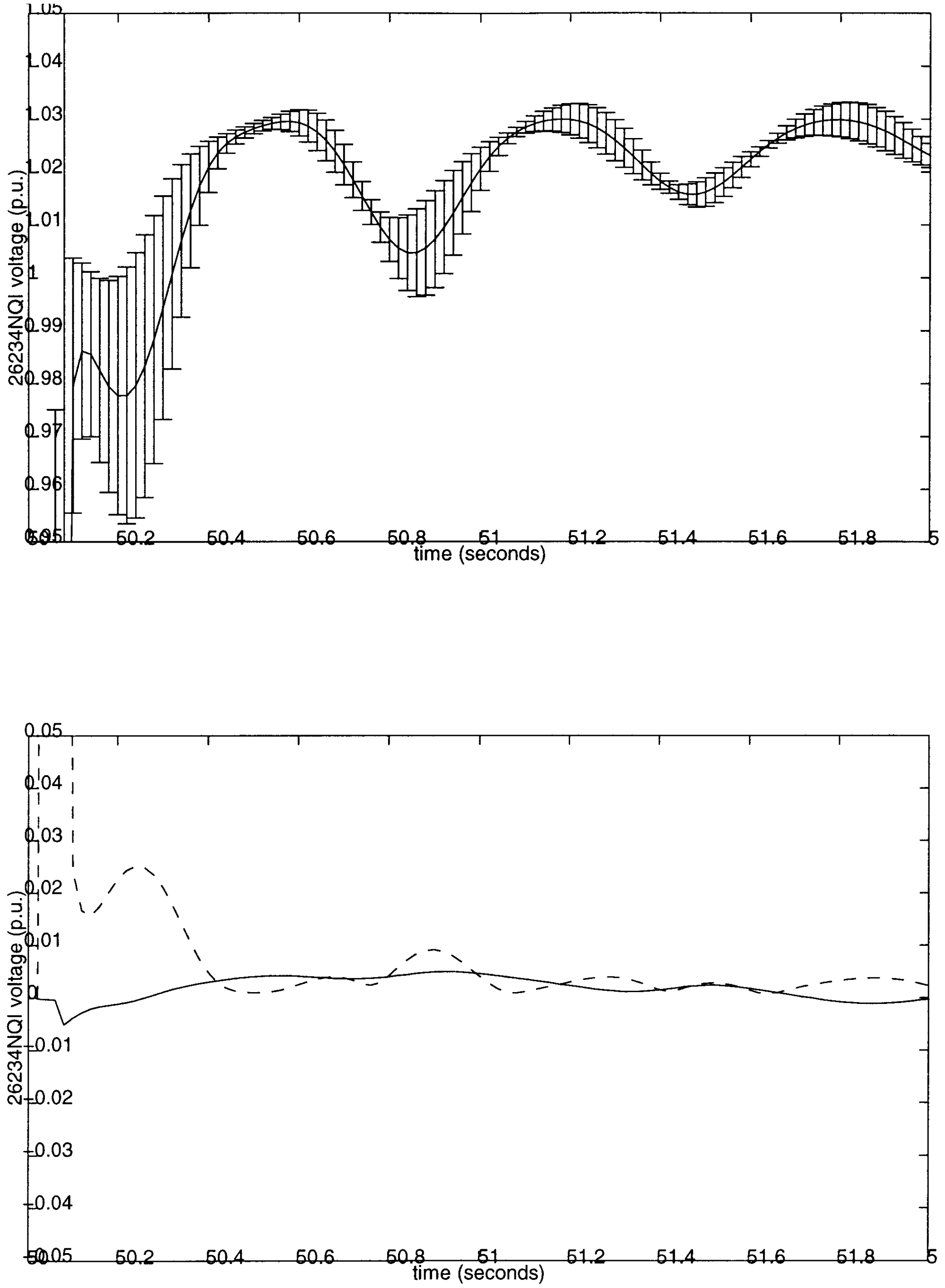


Figure 5.25: **Voltage at Bus 26234NQI - Uncertainty.** The top figure presents the expected value and standard deviation of the voltage at bus 26234NQI in our large system example after a disturbance, where the solid line is the expected value and the error bars are ±1 standard deviation from the mean. The bottom figure presents the standard deviation (dashed line) as well as the difference between a nominal simulation of the full model and a nominal simulation of the reduced-order model (solid line) for comparison.

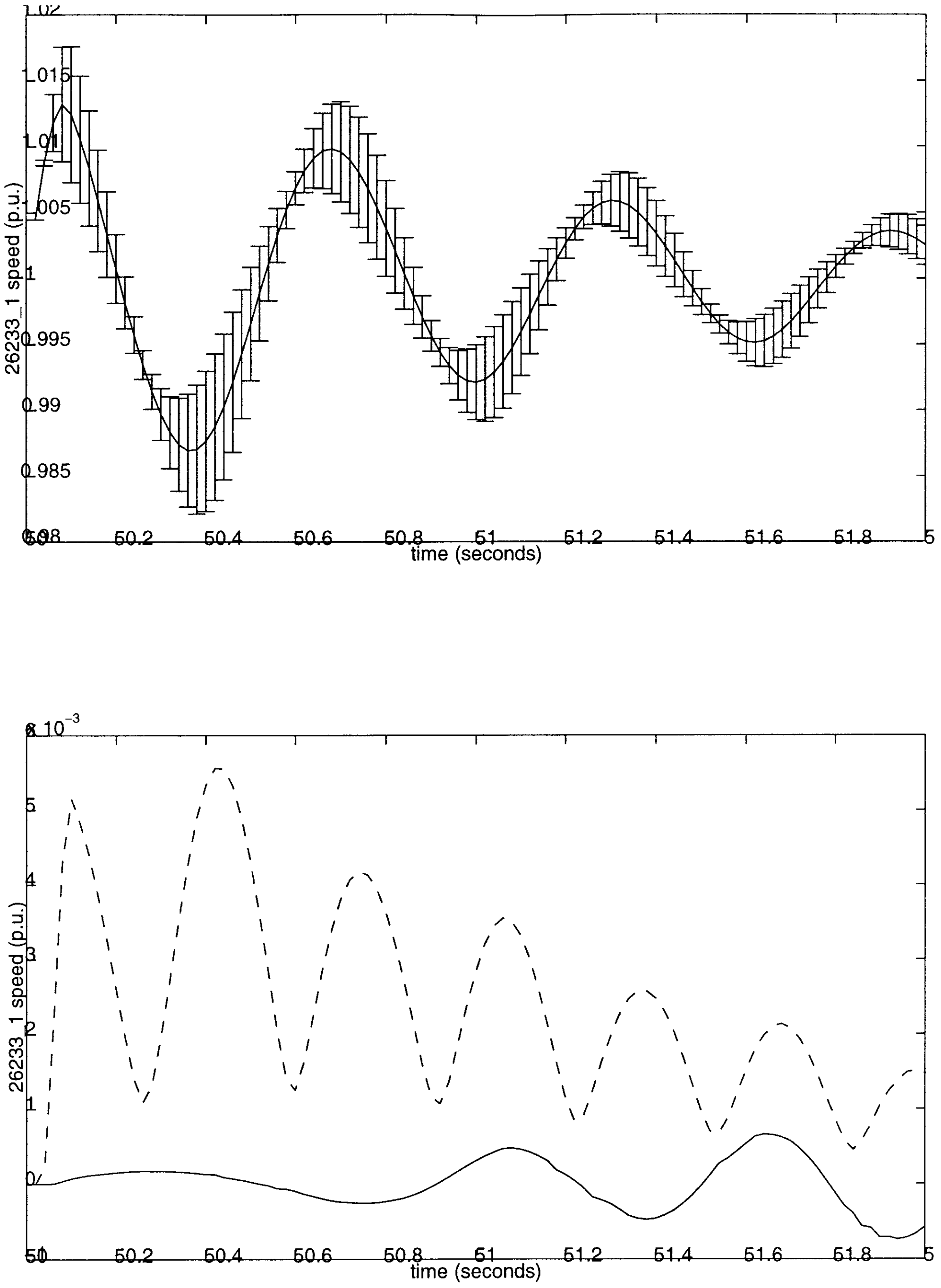


Figure 5.26: **Generator 26233_1 Speed - Uncertainty.** The top figure presents the expected value and standard deviation of the speed of generator 26233_1 in our large system example after a disturbance, where the solid line is the expected value and the error bars are ±1 standard deviation from the mean. The bottom figure presents the standard deviation (dashed) as well as the difference between a nominal simulation of the full model and a nominal simulation of the reduced-order model (solid line) for comparison.

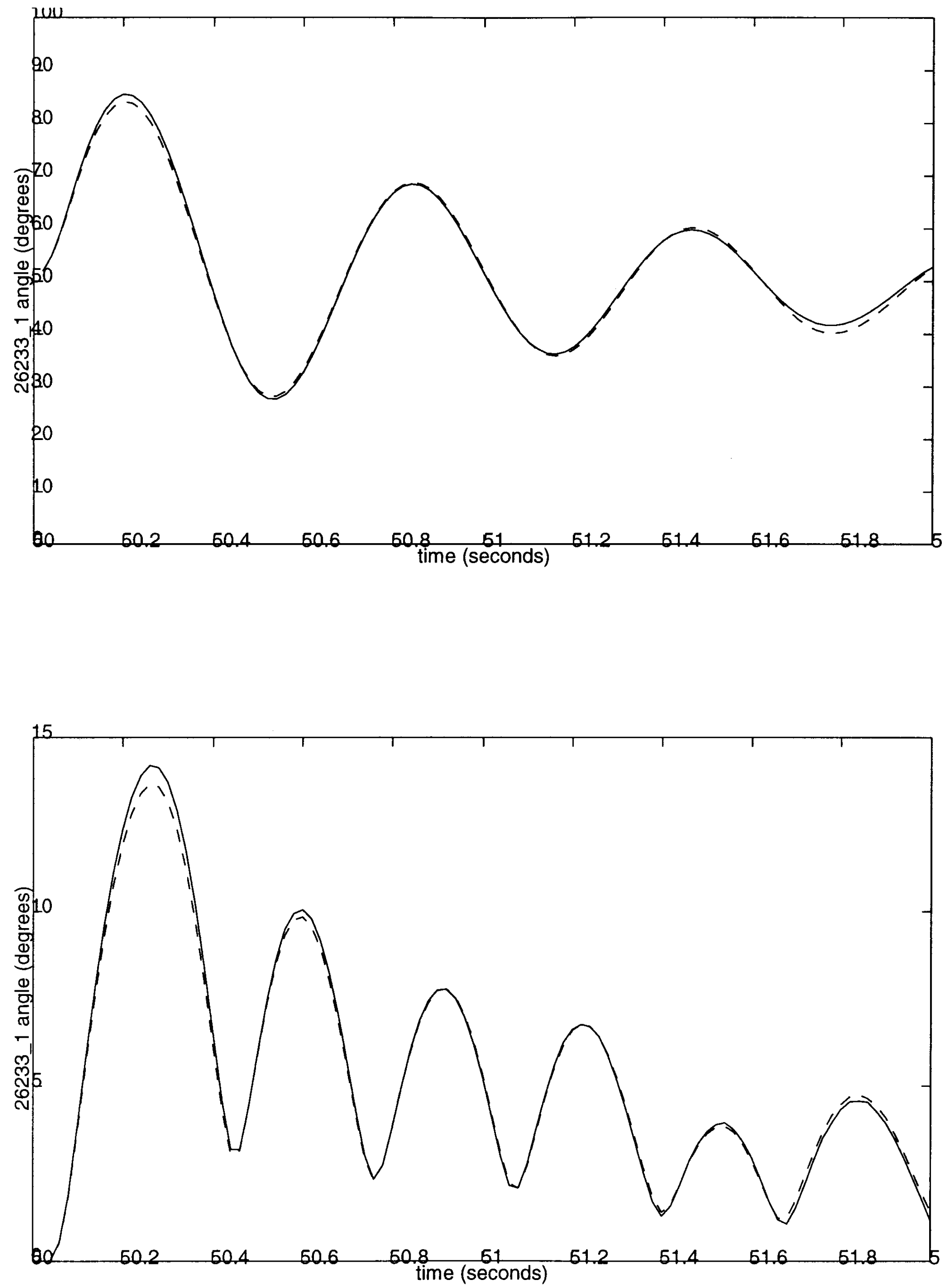
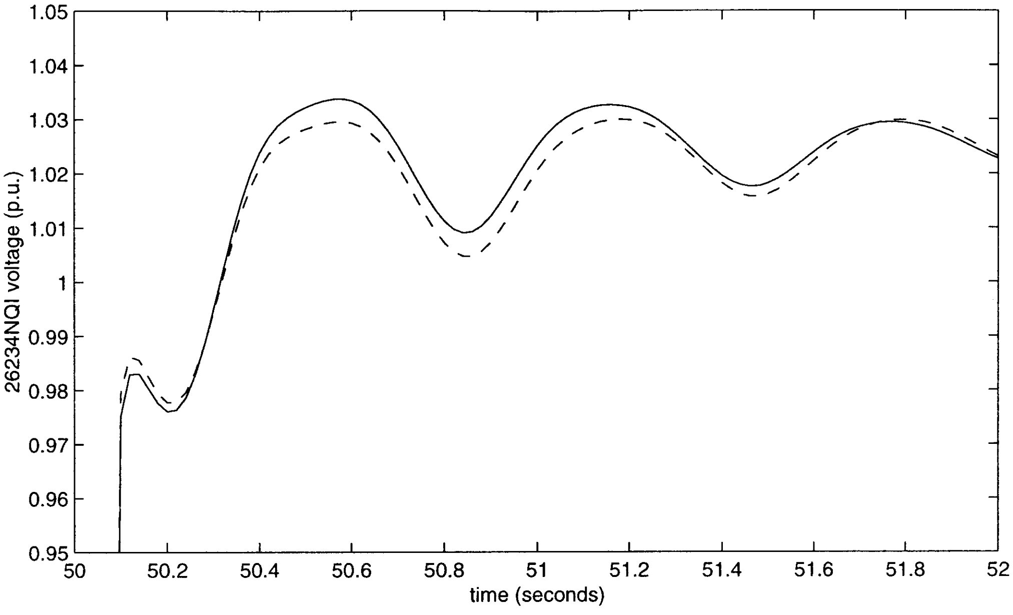


Figure 5.27: **Generator 26233_1 Angle - Comparison.** The top figure presents a comparison of the expected value of the angle of generator 26233_1 computed using a PCM model identified using the original system model (solid line) and identified using the reduced-order system (dashed line). The bottom figure makes the same comparison for the standard deviation.



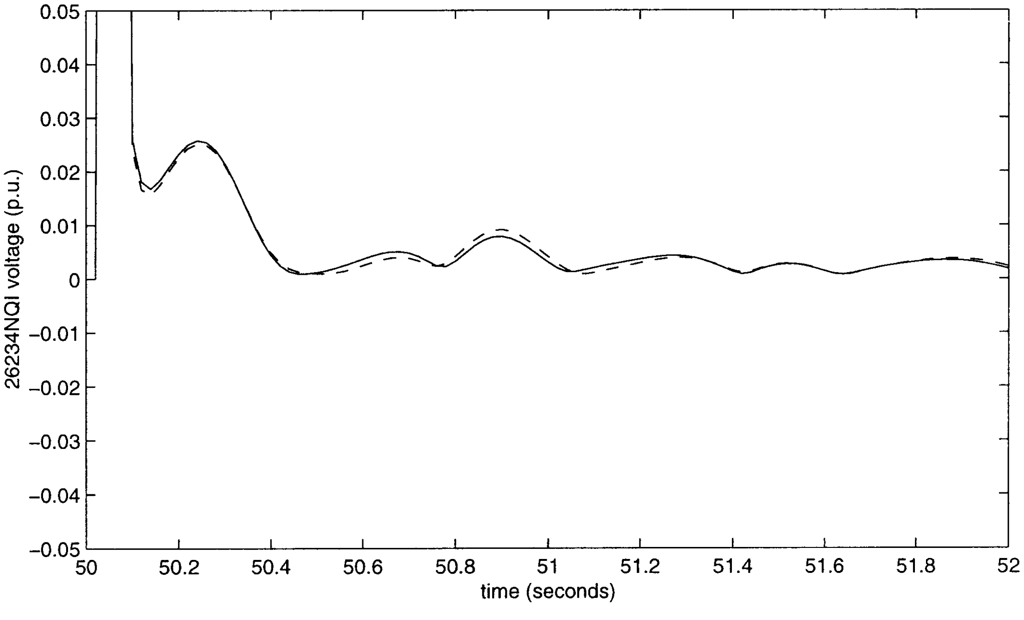


Figure 5.28: **Voltage at Bus 26234NQI - Comparison.** The top figure presents a comparison of the expected value of the voltage at bus 26234NQI computed using a PCM model identified using the original system model (solid line) and identified using the reduced-order system (dashed line). The bottom figure makes the same comparison for the standard deviation.

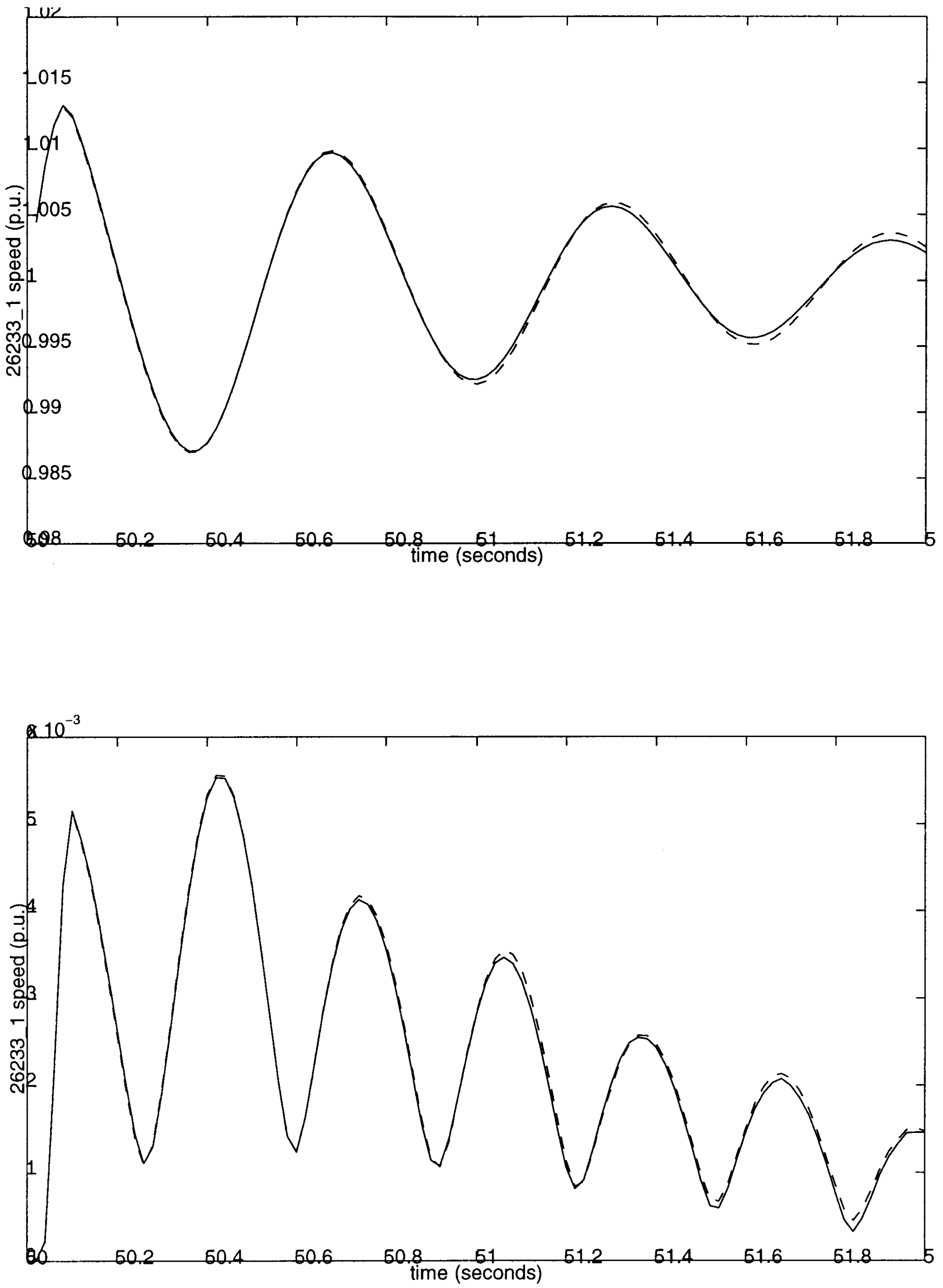


Figure 5.29: **Generator 26233_1 Speed - Comparison.** The top figure presents a comparison of the expected value of the speed of generator 26233_1 computed using a PCM model identified using the original system model (solid line) and identified using the reduced-order system (dashed line). The bottom figure makes the same comparison for the standard deviation.

	Our Method	Full Model PCM	Full Model Monte Carlo
Time to Reduce	8 min	-	-
Key Parameter ID	30 min	10 hr†	-
Number of Simulations	4	4	1000‡
Time per Simulation	30 sec	3 min	3 min
Total Time	**40 min**	**10 hr 12 min**	**2 days**

Table 5.4: **Time-Savings.** We compare the time needed to compute the standard deviation and expected value of any number of output variables using three techniques. (We do not consider variance reduction techniques applied to the Monte Carlo method because they are only applicable to one output variable). The first column is our entire three-step method, the second column is the probabilistic collocation method applied to the full model and the third column is a brute force Monte Carlo method applied to the full model. †This time is based on an estimate of the time required to perform a full eigenanalysis of the original system; we could not empirically verify this because our machines do not have sufficient memory to allow a fair comparison. ‡Our estimate of 1000 simulations is conservative; the true number could be more than ten times as large.

5.3 Summary

This chapter presents the results of tests we conducted of the uncertainty analysis method developed in this thesis. Although the results are not uniformly positive, our approach is shown to be promising.

The tests on the small example system demonstrate in detail each portion of our overall method; this example is designed to serve as a quick review of our approach as well as to test it. Since the example system is of a manageable size, we are able to explore the effects of separate components in our uncertainty analysis technique, in particular, the effects of reduced-order modeling, the probabilistic collocation method and our identification of key uncertain parameters. We study three separate time-varying output variables using the same set of system simulations (which is an advantage of our approach over traditional variance reduction techniques). The results for two of the three outputs (the generator angle and bus voltage) are very encouraging, and they amply demonstrate the promise of our method. The other output variable (generator speed) has a substantial error in its expected value because of problems with creating reduced-order models using SME on small networks. These results show the potential accuracy of our technique but, since the system is unrealistically small, the computation savings introduced by our approach are not apparent.

The results of tests on a large model are also presented. Unlike for the small model, these results cannot by easily compared to those obtained using traditional methods because traditional methods are far too time-consuming; as we have pointed out previously, our approach enables uncertainty analyses where they might not be otherwise possible. The time savings using our approach are apparent from this example. With a reasonable amount of overhead analysis time (which may be amortized over several events of interest), an uncertainty analysis of the system can by performed with approximately five simulations of the reduced-order model, which only requires roughly thirty seconds per simulation. Unfortunately, SME has not yet been fully tested on models of this scale and problems remain. The results for two of our output variables (the generator angle and speed) appear to be quite good, but the other output variable (bus voltage) has an error introduced by SME which is sometimes of the same order as the uncertainty introduced by the parameters. However, the uncertainty is relatively small for much of the transient and as a result the demands on the accuracy of SME are quite stringent here.

This chapter effectively concludes the body of this thesis. For the interested reader, we also include an appendix with the most important of the MATLAB scripts used to obtain the results here.

Chapter 6

Conclusions and Future Work

In this final chapter, we summarize the work performed for this thesis as well as the organization and presentation of the previous five chapters. On a final note, we also discuss possible extensions and applications of our work. Additional work could be done to further refine our techniques and the implications of the enabling aspect of our technique have barely been explored.

6.1 Summary

In Chapter 1, we introduced the topic of uncertainty analyses in large power systems and outlined our approach to this important problem. We proposed to study the uncertainty in dynamic simulations of large power systems by first creating a reduced-order model of the system and then performing a few carefully chosen simulations of the reduced-order model.

Chapter 2 presented the probabilistic collocation method, the uncertainty analysis technique which we apply to the reduced-order model. This chapter is designed to function as a primer. We presented a power system application of PCM, portions of which we had published previously, as well as two elementary examples. We also discussed aspects of the extension of PCM to multiple uncertain parameters. The extension to multiple uncertain parameters had been published by the group which created the method, but the underlying theoretical connections to single parameter PCM as well as limitations of the approach had not been fully explored.

In Chapter 3, we discussed our model reduction technique, synchronic modal equivalencing. This chapter functioned as a very basic primer for the reader who is unfamiliar with SME. To our knowledge, a complete and concise summary of all aspects of SME had not previously been published. In addition, we developed and explored the connections between the models used to perform SME and the assumed models upon which the underlying theory is based.

In Chapter 4, we discussed the connections between the uncertain parameters in the reduced-order model and the uncertain parameters in the full model. Assuming that the uncertainty in the parameters of the full model has already been modeled, we presented a computationally efficient method to identify which of the original uncertain parameters play a significant role in the uncertainty of each of the reduced-order system's parameters. We also adapted a eigensensitivity-based method for identifying the "key" uncertain parameters of the reduced-order model, which is necessitated by our use of PCM. It is important to note that this approach is not specific to any particular simulation, which has advantages and disadvantages. Our discussion was focused on the application of this knowledge to our uncertainty analysis approach, but the material in this chapter may be equally useful for identifying parameters whose uncertainty should be minimized.

Chapter 5 applied the techniques previously presented and discussed to two power system examples: a small example and a large example. Since a small power system is amenable to traditional uncertainty analysis approaches, we used the small example to test our method and compare our results to those obtained through more traditional methods. The results were quite promising. Though we could not compare our results on the large power system example to more traditional means, the analysis of large power systems is our goal and this example illustrated the enabling aspect of our work and the tremendous time savings which our method enjoys over more traditional approaches. The results for the large example were positive overall.

Our contributions have been in the analysis and investigation of previously developed but still new techniques for uncertainty analysis and model reduction. The major contribution is the integration of these techniques and the concrete demonstration of the feasibility of uncertainty analyses on large power systems using our approach.

6.2 Suggestions for Future Work

While we have attempted to address the major points related to our approach to uncertainty analyses of reduced-order power system models, our work could be extended in many ways. We attempt to delineate those possible directions for future research which we feel are either most pressing or have the greatest possibility to be fruitful.

6.2.1 Further Analysis of SME

Our work has barely scratched the surface in terms of fully analyzing synchronic modal equivalencing. Further work is necessary. Since the SME algorithm is currently the focus of active research, such work is made more difficult because one is "shooting at a moving target". However, the effort is well-spent even outside of the context of the current thesis because a good fundamental understanding of the current implementation of SME may open directions for further improvements, which has already been realized to a small extent through our work here. The most pressing need for our work is a better understanding of and an analytical expression for the error introduced by the SME reduction. If such an expression were available, we could better justify our use of SME, and, more importantly, we could refine our approach to include this error in our analysis by modeling the error as an uncertainty. We have not yet explored such a modeling of the error since a non-idealized error expression is not yet available, but our work on this topic is presented in the appendices.

6.2.2 Extension and Refinement of PCM

A few extensions and refinements of PCM would be very helpful for our work. First and foremost, a reliable method for identifying the appropriate order for our polynomial models would greatly improve our methods. We have assumed throughout that the appropriate order for our polynomial models was known *a priori*, but the answer to the question of appropriate order still eludes us. We generally try several model orders and then make a decision based on experience and intuition. This method is problematic because every new polynomial order tried implies extra system simulations, which we would like to avoid.

A secondary concern is extending PCM to accommodate multiple *dependent* uncertain parameters. We have attempted to point out a fruitful direction for exploring this topic, but our uncertainty analysis approach does not require us to consider dependent uncertain parameters. Therefore, the need for this in our application is not pressing.

6.2.3 Modeling of Uncertain Parameters

We have assumed throughout this thesis that the uncertainty in the parameters of the original power system model were modeled (accurately) using probability density functions. At the present moment, this assumption is largely unjustified. We believe that such descriptions are not as available as they should be presently because uncertainty analyses, such as those we perform, are deemed infeasible. However, our research shows that such analyses are feasible and the modeling and measurement of uncertainties in system parameters must advance in order for our technique to be useful.

6.2.4 Further Refinement and Automation of Implementation

We have implemented our ideas using scripts and functions written for MATLAB. While we have made every effort to fully implement our methods in a clear and understandable manner, much work remains to be done. The "key" parameter uncertainty identification is automated, but the probabilistic collocation method is still largely a labor intensive process. Many of these details could also be automated, and an investment in re-examining the code presented here and exploration of efficient ways to automate some of the details would be beneficial after the methods have been tested further. The potential time savings of our approach will remain largely untapped until the need for substantial user input is removed from the process.

6.2.5 Further Testing

Though we have tested our methods, our tests remain largely in the realm of "proof of principle". A more extensive battery of tests is warranted. Hopefully, more testing will further demonstrate the utility of our work, as well as discovering any inherent limitations which we have yet to uncover.

6.2.6 Simulation-specific "Key" Parameters

Our investigation of the simulation-specific key parameters in the context of our large power system example is hardly complete. We concentrated on identifying general critical parameter uncertainties in this thesis, but as our example showed, in some cases simulation-specific parameter uncertainties dominate. For the moment, we identify such critical simulation-specific uncertainties using a combination of intuition and experimentation, but a more formal approach would certainly be useful.

6.2.7 Studying Uncertainty

On a final note, we re-emphasize the enabling aspect of our work. We have not extensively discussed the possible uses for our technique, but the goal of this thesis is to introduce a tool for uncertainty analyses in power systems, which will hopefully be useful in many applications. Uncertainty analyses such as the few examples we have presented in this thesis are not routinely performed on power systems. An interesting further avenue for investigation is to explore the possible useful applications of uncertainty analysis tools on power systems.

Appendix A

Orthogonal Polynomial Generation

Efficient algorithms for the generation of ordered orthogonal polynomials exist. While even the Gram-Schmidt orthogonalization algorithm is not overly burdensome, we do not wish to spend much time generating the polynomials. (Any elementary linear algebra textbook discusses the Gram-Schmidt algorithm; see [58], for example). We present one of several possible algorithms, which is also the one used to create the orthogonal polynomials for this thesis. [32]

Let $\langle f, g\rangle$ denote the inner product of f and g, which are polynomials in z. Let h_i be a polynomial of order i which is orthogonal to all h_j where $j < i$. Let us assume that we already have suitable h_0 and h_1. We can compute h_2 knowing only h_0 and h_1 and subsequently any h_{i+1} by directly using only h_i and h_{i-1}, as follows:

$$a_i = \frac{\langle h_i z, h_i\rangle}{\langle h_i, h_i\rangle} \tag{A.1}$$

$$b_i = \frac{\langle h_i, h_i\rangle}{\langle h_{i-1}, h_{i-1}\rangle} \tag{A.2}$$

$$h_{i+1} = (z - a_i)h_i - b_i h_{i-1} \tag{A.3}$$

We can simplify the proof of this method without loss of generality by assuming that $\langle h_i, h_i\rangle = 1$ for all i.

The proof consists of showing that $\langle h_{i+1}, h_j\rangle = 0$ for all $j < i+1$. We do this for three separate cases. First, the case where $j < i-1$:

$$\begin{aligned} \langle h_{i+i}, h_j\rangle &= \langle (z - \langle h_i z, h_i\rangle)h_i - h_{i-1}, h_j\rangle & \text{(A.4)} \\ &= \langle (z - \langle h_i z, h_i\rangle)h_i, h_j\rangle & \text{(A.5)} \\ &= \langle z h_i, h_j\rangle = \langle h_i, h_j z\rangle = 0 & \text{(A.6)} \end{aligned}$$

The last equality holds since $h_j z$ is a polynomial of order less than $i-1$ and h_i is orthogonal to all such polynomials by construction.

Next, we investigate $j = i - 1$:

$$\begin{aligned}
\langle h_{i+i}, h_{i-1}\rangle &= \langle (z - \langle h_i z, h_i\rangle) h_i - h_{i-1}, h_{i-1}\rangle && \text{(A.7)}\\
&= \langle z h_i, h_{i-1}\rangle - \langle h_{i-1}, h_{i-1}\rangle && \text{(A.8)}\\
&= \langle h_i, z h_{i-1}\rangle - \langle h_i, h_i\rangle && \text{(A.9)}\\
&= \langle h_i, z h_{i-1} - h_i\rangle && \text{(A.10)}\\
&= 0 && \text{(A.11)}
\end{aligned}$$

The last equality holds since $z h_{i-1} - h_i$ is a constant by construction and h_i is orthogonal to h_0.

The last case is $j = i$:

$$\begin{aligned}
\langle h_{i+i}, h_i\rangle &= \langle (z - \langle h_i z, h_i\rangle) h_i - h_{i-1}, h_i\rangle && \text{(A.12)}\\
&= \langle (z - \langle h_i z, h_i\rangle) h_i, h_i\rangle && \text{(A.13)}\\
&= \langle z h_i, h_i\rangle - \langle \langle h_i z, h_i\rangle h_i, h_i\rangle && \text{(A.14)}\\
&= \langle h_i z, h_i\rangle (1 - \langle h_i, h_i\rangle) && \text{(A.15)}\\
&= 0 && \text{(A.16)}
\end{aligned}$$

The last equality holds by our assumption that h_i has unit length. This completes the proof that the algorithm is correct.

Appendix B

SME - Error and Robustness

In Chapter 3, we presented a basic introduction to SME; our concern in this appendix is to extend this work and provide a more detailed analysis of portions of SME. In some sense, the present appendix is necessary to justify our choice of SME as our model reduction technique. Our final goal is to perform uncertainty analyses of power systems using model reduction to enable the process. If the results of such uncertainty analyses are to be useful, the model reduction technique (in our case SME) must be relatively insensitive to parameter changes and also must not introduce too much additional error (uncertainty) into the model. The present appendix addresses these two issues for SME.

As noted in Chapter 3, SME presents a challenge for analysis because of its heuristic nature. However, many of the properties of SME can be predicted by analyzing ideal systems with properties that our systems only possess (at best) approximately. We continue in that spirit in this appendix. We derive and discuss results relevant to SME in idealized settings to indicate positive properties with respect to error and sensitivity. Of course, such results do not conclusively prove that SME always behaves as desired in practice. However, such results in conjunction with positive empirical findings do increase our confidence when applying SME.

B.1 Error Analysis

When analyzing the error in SME, we only deal with the ability of the reduced linear model to match the full linearized model of our power system. We limit ourselves in this way mainly because the SME procedure is wholly carried out in the context of a linear system and the extension to nonlinear systems, while yielding promising results, is not yet well-justified theoretically. In a similar spirit, we assume that the system exhibits exact synchrony and that we have appropriately identified basis generators and partitioned the system in a way which brings out this structure. Our goal is to understand the behavior of a system which is ideally suited to SME and note any inherent limitations of our methods. Because the analysis is strictly limited to the linearized model, we do not treat the network reduction and only discuss the generator equivalencing, since the network reduction on the linear system is exact.

This section comprises two main sections, corresponding to two progressively less idealized approaches to the generator equivalencing. The first section examines the case where the system exhibits synchrony with respect to the chord and we attempt to preserve all of the eigenvalues of the chord as well as their right eigenvectors in the reduced model. The work presented here builds directly upon previously published analyses of selective modal analysis (SMA) in [43], [44], [59] and especially [60]. We will present the relevant results in the first subsection and build from there. Our analysis shows that there are fundamental limitations in the present SME algorithm when one attempts to retain the entire chord. We refer to this approach as a "full-mode correction". The second section more closely parallels the actual SME algorithm, in that we only attempt to retain the zero eigenvalue and its right eigenvector. Such an approach allows greater flexibility in the choice of the correction matrix, with which one can attempt to improve other behaviors of interest; in particular we explore improving the steady-state behavior of the reduced system. We refer to this approach as a "zero-mode correction".

B.1.1 Full-Mode Correction

We assume that the synchrony in the system has already been identified and the variables partitioned into relevant (to be retained) and less-relevant (to be equivalenced) variables. We retain the same number of variables as modes in the chord, though the method is not limited to only this case. For the purposes of the analysis here, we assume that the system exhibits synchrony with respect to the chord, as a best case scenario. Usually, the synchrony is only approximate which would further complicate the analysis and also degrade the performance of the reduced-order model, in general.

Let us assume a fairly general linear state-space model for the system:

$$\dot{x} = \mathbf{A}x + bu \tag{B.1}$$

$$y = \mathbf{C}x \tag{B.2}$$

where x is the vector of state variables, b is a vector, u is a scalar input function, and y is a vector of output variables. We allow for an input to the system here though it is not necessarily clear whether the events on our actual nonlinear power system models are best reflected in our linearized analysis as changes to initial conditions or inputs. We partition the system variables into relevant variables, r, and less-relevant variables, z, to perform the reduction, as follows:

$$\begin{bmatrix} \dot{r} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \mathbf{A_{rr}} & \mathbf{A_{rz}} \\ \mathbf{A_{zr}} & \mathbf{A_{zz}} \end{bmatrix} \begin{bmatrix} r \\ z \end{bmatrix} + \begin{bmatrix} b_r \\ b_z \end{bmatrix} u \tag{B.3}$$

$$y = \begin{bmatrix} \mathbf{C_r} & \mathbf{C_z} \end{bmatrix} \begin{bmatrix} r \\ z \end{bmatrix} \tag{B.4}$$

Unless otherwise noted, we partition matrices using similar notation throughout this subsection.

B.1.1.1 Change of Variables Approach

If we were willing to sacrifice our physically meaningful variables, we could in general improve the performance of our linear reduced-order model. Although we do not allow for this possibility in SME, an analysis of such an approach is helpful as a point of contrast for an approach more in the spirit of SME. We use the change of variables suggested in [60]:

$$\begin{bmatrix} \rho \\ \zeta \end{bmatrix} = \begin{bmatrix} \mathbf{I} + \mathbf{LK} & -\mathbf{L} \\ -\mathbf{K} & \mathbf{I} \end{bmatrix} \begin{bmatrix} r \\ z \end{bmatrix} \tag{B.5}$$

where $\mathbf{K}$ is chosen such that:

$$\mathbf{K}\mathbf{A_{rr}} + \mathbf{K}\mathbf{A_{rz}}\mathbf{K} - \mathbf{A_{zz}}\mathbf{K} - \mathbf{A_{zr}} = 0 \tag{B.6}$$

and $\mathbf{L}$ is chosen such that:

$$(\mathbf{A_{rr}} + \mathbf{A_{rz}}\mathbf{K})\mathbf{L} - \mathbf{L}(\mathbf{A_{zz}} - \mathbf{K}\mathbf{A_{rz}}) + \mathbf{A_{rz}} = 0 \tag{B.7}$$

Such a $\mathbf{K}$ and $\mathbf{L}$ are guaranteed to exist if the system exhibits synchrony in the chosen chord.

The resulting system has decoupled state dynamics:

$$\begin{aligned} \begin{bmatrix} \dot{\rho} \\ \dot{\zeta} \end{bmatrix} &= \begin{bmatrix} \mathbf{A_{rr}} + \mathbf{A_{rz}}\mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{A_{zz}} - \mathbf{K}\mathbf{A_{rz}} \end{bmatrix} \begin{bmatrix} \rho \\ \zeta \end{bmatrix} + \begin{bmatrix} \mathbf{I} + \mathbf{LK} & -\mathbf{L} \\ -\mathbf{K} & \mathbf{I} \end{bmatrix} \begin{bmatrix} b_r \\ b_z \end{bmatrix} u & \text{(B.8)} \\ y &= \begin{bmatrix} \mathbf{C_r} & \mathbf{C_z} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{L} \\ \mathbf{K} & \mathbf{I} + \mathbf{KL} \end{bmatrix} \begin{bmatrix} \rho \\ \zeta \end{bmatrix} & \text{(B.9)} \end{aligned}$$

The obvious reduction is to ignore ζ and examine only the contribution of ρ to the output:

$$y_\rho = \begin{bmatrix} \mathbf{C_r} & \mathbf{C_z} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{K} \end{bmatrix} \rho \tag{B.10}$$

where ρ is described by the following equation:

$$\dot{\rho} = (\mathbf{A_{rr}} + \mathbf{A_{rz}}\mathbf{K})\rho + \begin{bmatrix} \mathbf{I} + \mathbf{LK} & -\mathbf{L} \end{bmatrix} \begin{bmatrix} b_r \\ b_z \end{bmatrix} u \tag{B.11}$$

The error, the difference between the output of the full system and the output of our reduced model, is then given by:

$$y_\zeta = \begin{bmatrix} \mathbf{C_r} & \mathbf{C_z} \end{bmatrix} \begin{bmatrix} \mathbf{L} \\ \mathbf{I} + \mathbf{KL} \end{bmatrix} \zeta \tag{B.12}$$

where ζ is given by the following equation:

$$\dot{\zeta} = (\mathbf{A_{zz}} - \mathbf{K}\mathbf{A_{rz}})\zeta + \begin{bmatrix} -\mathbf{K} & \mathbf{I} \end{bmatrix} \begin{bmatrix} b_r \\ b_z \end{bmatrix} u \tag{B.13}$$

The error can be written compactly by using the Laplace transform and solving for y_ζ in the frequency domain:

$$Y_\zeta(s) = \begin{bmatrix} \mathbf{C_r} & \mathbf{C_z} \end{bmatrix} \begin{bmatrix} \mathbf{L} \\ \mathbf{I} + \mathbf{KL} \end{bmatrix} (s\mathbf{I} - \mathbf{A_{zz}} + \mathbf{KA_{rz}})^{-1} \begin{bmatrix} -\mathbf{K} & \mathbf{I} \end{bmatrix} \begin{bmatrix} b_r \\ b_z \end{bmatrix} U(s) \qquad \text{(B.14)}$$

An even more convenient form is derived using a modal decomposition of the original output in the frequency domain:

$$Y(s) = \begin{bmatrix} \mathbf{C_r} & \mathbf{C_z} \end{bmatrix} (\mathbf{V_z}(s\mathbf{I} - \mathbf{\Lambda}_z)^{-1}\mathbf{W_z}^{\mathrm{T}} + \mathbf{V_r}(s\mathbf{I} - \mathbf{\Lambda}_r)^{-1}\mathbf{W_r}^{\mathrm{T}}) \begin{bmatrix} b_r \\ b_z \end{bmatrix} U(s) \qquad \text{(B.15)}$$

where $\mathbf{V_r}$ ($\mathbf{V_z}$) comprises the columns of the right eigenvector matrix corresponding to the eigenvalues that were (not) retained, $\mathbf{\Lambda}_r$ ($\mathbf{\Lambda}_z$) is a diagonal matrix with those eigenvalues on the diagonal and $\mathbf{W_r}$ ($\mathbf{W_z}$) comprises the columns of the left eigenvector matrix corresponding to those eigenvalues (note that $\mathbf{W}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$). Since we retain the structure associated with r, the error is given by

$$Y_\zeta(s) = \begin{bmatrix} \mathbf{C_r} & \mathbf{C_z} \end{bmatrix} \mathbf{V_z}(s\mathbf{I} - \mathbf{\Lambda}_z)^{-1}\mathbf{W_z}^{\mathrm{T}} \begin{bmatrix} b_r \\ b_z \end{bmatrix} U(s) \qquad \text{(B.16)}$$

(The same result can be derived directly from (B.14) by invoking (B.6) and (B.7)).

The dynamics of the error are entirely determined by the dynamics of the portion of the system which is not retained, in the sense that only the neglected modes appear in this formulation. The error is zero in the case where b is a right eigenvector corresponding to one of the retained eigenvalues (or a linear combination of any such eigenvectors) because $\mathbf{W_z}^{\mathrm{T}}b$ is identically zero in this case. Similarly, if the removed variables are decoupled from the state variables that directly appear in y, the error is also zero. We refer to the analytical approach described above as the "change of variables approach".

As we shall soon see, the analysis that follows in the next subsection differs from that given above and is closer in spirit to SME. The major difference is the new model reduction is not formulated as a change of variables. In practical terms, the reduction is nearly identical, except b is not premultiplied by a matrix in the reduced-order model (except to reduce its dimension). In other words, our new reduced-order model is of the form

$$\dot{\pi} = (\mathbf{A_{rr}} + \mathbf{A_{rz}}\mathbf{K})\pi + \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} b_r \\ b_z \end{bmatrix} u \tag{B.17}$$

$$y_\pi = \begin{bmatrix} \mathbf{C}_r & \mathbf{C}_z \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{K} \end{bmatrix} \pi \tag{B.18}$$

This form can be derived by assuming that $r = \pi$ and $z = \mathbf{K}\pi$ in the original model. We refer to this approach as the "direct substitution approach".

Though we do not pursue the details here, a system event represented by a nonzero initial condition also gives rise to different responses under the two approaches. In both cases, an initial condition which satisfies the following relation yields identical responses (without error):

$$z_0 = \mathbf{K}r_0 \tag{B.19}$$

However, they will otherwise differ due to the lack of a matrix $\mathbf{L}$ in the direct substitution approach.

B.1.1.2 Direct Substitution Approach

Instead of performing a change of variables, we instead apply a set of substitutions ($r = \pi$ and $z = \mathbf{K}\pi$) only to the right side of (B.3), (B.4). The system matrix is transformed as in the previous subsection, but the input matrix b remains the same. In this case, if we assume that a relationship of the form $z = \mathbf{K}r$ exists and is exact when only the modes in the chord are excited, the error can be written in a reasonably compact form in the frequency domain

$$E(s) = \mathbf{C}\left[\mathbf{V}\,(s\mathbf{I} - \Lambda)^{-1}\,\mathbf{W}^{\mathrm{T}} - \begin{bmatrix} \mathbf{I} \\ \mathbf{K} \end{bmatrix} \mathbf{V_{rr}}\,(s\mathbf{I} - \Lambda_{rr})^{-1}\,\mathbf{V_{rr}}^{-1} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}\right] bU(s) \tag{B.20}$$

where $\mathbf{V}_{rr}$ and Λ_{rr} are defined consistently with our earlier subscripting convention. (Note that $\mathbf{V}_{rr}$ must be invertible for the synchrony to be exact in the chord, assuming that the original right eigenvectors are distinct). This form arises because the reduced-order system retains the eigenvalues in the chord as well as the portions of the corresponding right eigenvectors which relate to the retained variables. In general, the left eigenvectors are not retained, which is reflected by the formula.

When SME is applied, the study area is generally chosen to include the state(s) that contribute directly to the output, as the local dynamics of these state(s) are likely to be important. Therefore, it is fruitful to look at a **C** that reflects this observation. We assume **C** to be of the following form:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_\mathbf{r} & \mathbf{0} \end{bmatrix} \tag{B.21}$$

This leads to the following simplification of the error formula:

$$\begin{aligned} E(s) \;&=\; \begin{bmatrix} \mathbf{C}_\mathbf{r}\mathbf{V}_\mathbf{rr} & \mathbf{C}_\mathbf{r}\mathbf{V}_\mathbf{rz} \end{bmatrix} (s\mathbf{I} - \mathbf{\Lambda})^{-1} \mathbf{W}^\mathrm{T} b U(s) - \\ &\mathbf{C}_\mathbf{r}\mathbf{V}_\mathbf{rr} \left(s\mathbf{I} - \mathbf{\Lambda}_{rr}\right)^{-1} {\mathbf{V}_\mathbf{rr}}^{-1} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} b U(s) \end{aligned} \tag{B.22}$$

Similarly, if we are to disturb a state variable of the system, we also generally want to retain that state, as its local dynamics are important. We assume an appropriate form for b:

$$b = \begin{pmatrix} b_r \\ 0 \end{pmatrix} \tag{B.23}$$

Simplifying based on this assumption without applying the simplification based on the form of **C**:

$$E(s) = \mathbf{C} \left[\mathbf{V} \left(s\mathbf{I} - \mathbf{\Lambda}\right)^{-1} \begin{bmatrix} {\mathbf{W}_\mathbf{rr}}^\mathrm{T} b_r \\ {\mathbf{W}_\mathbf{zr}}^\mathrm{T} b_r \end{bmatrix} - \begin{bmatrix} \mathbf{I} \\ \mathbf{K} \end{bmatrix} \mathbf{V}_\mathbf{rr} \left(s\mathbf{I} - \mathbf{\Lambda}_{rr}\right)^{-1} {\mathbf{V}_\mathbf{rr}}^{-1} b_r \right] U(s) \tag{B.24}$$

If both assumptions are applied simultaneously, the error is in a form which can be more easily interpreted:

$$\begin{aligned} E(s) \;&=\; \mathbf{C}_\mathbf{r}\mathbf{V}_\mathbf{rr} \left(s\mathbf{I} - \mathbf{\Lambda}_{rr}\right)^{-1} \left({\mathbf{W}_\mathbf{rr}}^\mathrm{T} - {\mathbf{V}_\mathbf{rr}}^{-1}\right) b_r U(s) + \\ &\mathbf{C}_\mathbf{r}\mathbf{V}_\mathbf{rz} \left(s\mathbf{I} - \mathbf{\Lambda}_{zz}\right)^{-1} {\mathbf{W}_\mathbf{zr}}^\mathrm{T} b_r U(s) \end{aligned} \tag{B.25}$$

Under these special conditions on the input and the output, the second term is exactly the error of the change of variables approach, which can be confirmed by comparing this term to (B.16). Unlike in (B.16), we have another term (the first term above) which depends on the modes which we retained. In other words, we have increased the error over the change of variables approach. Perhaps more importantly, our error depends on all of the modes of the system. In a power system application, the system has a zero-mode, which could create a steady-state error in our reduced-order model. The change of variables approach should have no steady-state error.

We can algebraically manipulate (B.25) to further clarify the error. First, we observe that:

$$Y_\varsigma(s) = Y(s) - Y_\rho(s) \tag{B.26}$$

$$E(s) = Y(s) - Y_\pi(s) \tag{B.27}$$

From these equations, we derive the following relationship:

$$E(s) - Y_\varsigma(s) = Y_\rho(s) - Y_\pi(s) \tag{B.28}$$

Using the superposition principle on the right-hand side and our derived formulas for $E(s)$ and $Y_\varsigma(s)$ gives us:

$$\mathbf{C_r V_{rr}}\left(s\mathbf{I} - \mathbf{\Lambda}_{rr}\right)^{-1}\left(\mathbf{W_{rr}}^{\mathrm{T}} - \mathbf{V_{rr}}^{-1}\right) b_r U(s) = \mathbf{C_r V_{rr}}\left(s\mathbf{I} - \mathbf{\Lambda}_{rr}\right)^{-1}\mathbf{V_{rr}}^{-1}\mathbf{LK} b_r U(s) \tag{B.29}$$

We also note that, as defined in [60], $\mathbf{L} = -\mathbf{V_{rr}W_{rz}}^{\mathrm{T}}$. With these results, we are able to express the error in our direct substitution approach as:

$$E(s) = \mathbf{C_r V_{rz}}\left(s\mathbf{I} - \mathbf{\Lambda}_{zz}\right)^{-1}\mathbf{W_{zr}}^{\mathrm{T}} b_r U(s) - \mathbf{C_r V_{rr}}\left(s\mathbf{I} - \mathbf{\Lambda}_{rr}\right)^{-1}\mathbf{W_{rz}K} b_r U(s) \tag{B.30}$$

Looking at the term with the eigenvalues which are not retained, we note that $\mathbf{V_{rz}}$ and $\mathbf{W_{zr}}^{\mathrm{T}}$ are the portions of the right and left eigenvector matrices corresponding to the relevant variables and the modes which are not retained. One could interpret this as the error resulting from the contribution of the "less-relevant" modes to the behavior of the retained variables. As long as the input is such that it does not exclusively excite the retained modes, this error cannot be avoided. Therefore, it appears in both of the methods described above.

Both approaches retain the eigenvalues of interest as well as the associated right eigenvectors in the reduced model. However, the change of variables approach also excites the retained modes in the reduced model exactly as in the full model; this is accomplished by changing the input weighting vector b to reflect the change of variables performed. The direct substitution approach does not modify b, and the corresponding term in (B.30) is a result of not exciting the *retained* modes "correctly". The form of this term is not as easy to explain as for the less-relevant eigenvalue error term. Both $\mathbf{V_{rr}}$ and $\mathbf{W_{rz}}^{\mathrm{T}}$ correspond to the relevant modes, as expected, but the first of these is related to the retained variables while the other is related to the removed variables. We do note that $\mathbf{K}$ satisfies $\mathbf{V_{zr}} = \mathbf{K}\mathbf{V_{rr}}$, but this is not much help since matrix multiplication does not commute.

B.1.1.3 Special Case: Reduction to a Scalar System

One avenue for clarifying the arguments in the previous section is to look at the special case where our reduced order model consists of only one state variable. This, of course, allows us to commute certain terms in our error because they are then scalar quantities. Recalling our earlier definition of the participation of the ith mode in the kth variable, p_{ik}, the error can be written in the following compact form:

$$E(s) = \mathbf{C_r}\left(\sum_{k=2}^{n} \frac{p_{1k}}{s-\lambda_k} - \frac{\sum_{j=2}^{n} p_{1j}}{s-\lambda_1}\right) b_r U(s) \tag{B.31}$$

Since the participation factors sum to unity when added over either of their indices, we could also write this in two other forms:

$$E(s) = \mathbf{C_r}\left(\sum_{k=2}^{n} \frac{p_{1k}}{s-\lambda_k} - \frac{1-p_{11}}{s-\lambda_1}\right) b_r U(s) \tag{B.32}$$

or

$$E(s) = \mathbf{C_r}\left(\sum_{k=2}^{n} \frac{p_{1k}}{s-\lambda_k} - \frac{\sum_{i=2}^{n} p_{i1}}{s-\lambda_1}\right) b_r U(s) \tag{B.33}$$

In either event, the error term corresponding to the less-relevant eigenvalues is expressed in terms of the participation of the retained variable in the less-relevant modes, as noted in the previous section. The other term is easier to interpret in this case, but caution must be exercised because, though the participation factors sum to unity, they are not restricted to being nonnegative or even real. By not changing the input to ensure that the retained modes are excited correctly, the reduced order model has an error term related to these modes. In other words, some of our input (which is only applied to the retained variables) needs to be "spread" over the other variables to excite the mode correctly. The amount of change to the input is related to either the participation of these other variables in the retained mode or the participation of the retained variable in the less-relevant modes.

B.1.1.4 Summary

If the full chord is to be preserved, the optimum thing to do would be to perform a change of variables that block diagonalizes the system. However, such a change of variables necessitates appropriately modifying the input to the state-space model. In terms of the nonlinear model on which our state-space model is based and the disturbances we typically wish to model (e.g., line outages), it is difficult to ascertain how exactly to model these disturbances as inputs to our linear state-space model, or what it would mean to modify these inputs. If we model our disturbance as a non-zero initial condition instead of an input, we also experience similar problems in terms of modeling the system after the change of variables, since the resulting variables can no longer be modeled as power system components.

So, the prospects are not bright if we wish to retain all of the eigenvalues in the chord exactly, as well as retain our original physically-meaningful variables. However, if we only concentrate on retaining some portion of the chord exactly, we should be able to retain those modes as well as have additional degrees of freedom with which to optimize other relevant criteria. The next section will outline how to proceed if we only retain a portion of the eigenvalues (specifically, only the zero eigenvalue and its right eigenvector) and try to intelligently use these extra degrees of freedom. We only retain the zero-mode when we do the SME reduction, so the next section more closely resembles the approach used in SME than the current section.

B.1.2 Zero-Mode Correction

In the previous section, **K** was completely determined by the chord selected and the variables we wanted to retain. We break from our earlier approach and allow ourselves to pick any

K which satisfies a simple set of linear equations that do not fully determine **K**. The model reduction based on this new **K** is only guaranteed to retain the zero eigenvalue and its right eigenvector from the original model. In this section, we primarily concentrate on using the extra freedom to match the steady-state behavior of the full system with our reduced system because we have experienced steady-state problems with some of our reduced-order models; other possible uses for these degrees of freedom in choosing **K** exist and we use steady-state matching as a representative example.

Later in this section, we constrain our **K** in a way consistent with the present implementation of SME. Namely, the removed variables are represented as linear combinations of only a subset of the retained variables (the angle and speed variables of the basis generators), as previously discussed. However, as with full-mode correction, an analysis of a more general structure for **K** is a fruitful preface.

B.1.2.1 Full K

We again deal with a general state-space model, but we assume additionally that the system has a single zero-mode and the other modes of the system are strictly stable. Later, we further restrict the form to more closely represent a power system, but for the time being we remain with a fairly general model of the form:

$$\dot{x} = \mathbf{A}x + bu \tag{B.34}$$

We can perform a modal decomposition to find the steady-state response of this system to a finite length (in time) input. [N.B., the system has a zero mode, so a step input does not have a steady-state response]:

$$X(s) = \sum_{i=1}^{n} \frac{v_i w_i^{\mathrm{T}}}{s - \lambda_i} bu(s) \tag{B.35}$$

where the system has n state variables, the λ_i are the eigenvalues ($\lambda_1 = 0$), and the v_i and w_i^{T} are the corresponding left and right eigenvectors. If we apply the final-value theorem to this expression, we can derive a straightforward expression for the steady-state behavior:

$$\begin{aligned} x(\infty) &= \lim_{s\to 0} s \sum_{i=1}^{n} \frac{v_i w_i^{\mathrm{T}}}{s-\lambda_i} b u(s) & \text{(B.36)} \\ &= v_1 w_1^{\mathrm{T}} b \lim_{s\to 0} u(s) & \text{(B.37)} \\ &= v_1 w_1^{\mathrm{T}} b \int_0^{\infty} u(t) dt & \text{(B.38)} \end{aligned}$$

The expression (B.38) shows that the outer product of the left and right eigenvectors corresponding to the zero-mode is a key quantity. We already retain the shape of the right eigenvector with the present reduction; keeping the shape of the *left* eigenvector as well as the scaling of the product $v_1 w_1^{\mathrm{T}}$ are our goals. (Note, eigenvectors are only specified up to a scaling factor, but the inner and outer products of the left and right eigenvectors are completely determined by the system with no scaling ambiguities). Now, we need to reexamine how the reduction is performed, in order to achieve these goals.

As in the previous subsection, we need to partition the system into relevant (r) and less-relevant (z) variables. We assume that the input is only applied to the relevant variables and that there are m of these:

$$\begin{bmatrix} \dot{r} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \mathbf{A_{rr}} & \mathbf{A_{rz}} \\ \mathbf{A_{zr}} & \mathbf{A_{zz}} \end{bmatrix} \begin{bmatrix} r \\ z \end{bmatrix} + \begin{bmatrix} b_r \\ 0 \end{bmatrix} u \tag{B.39}$$

As before, the reduced system is obtained by the substitution ($r = \pi$ and $z = \mathbf{K}\pi$):

$$\dot{\pi} = (\mathbf{A_{rr}} + \mathbf{A_{rz}}\mathbf{K})\pi + b_r u, \tag{B.40}$$

but, now, instead of requiring $\mathbf{K}$ to satisfy a relationship involving the right eigenvectors of all of the modes in the chord, $\mathbf{K}$ is constrained simply by

$$v_z = \mathbf{K} v_r, \tag{B.41}$$

where v_z and v_r are the portions of the zero-mode right eigenvector corresponding to the less-relevant and relevant variables, respectively. (As we see later, we can predict the form of the right eigenvector for our power system models without actually having to do an eigenanalysis, which is an advantage of this approach). In practical terms, we are constraining a weighted sum of each of the rows of $\mathbf{K}$. However, this leaves us with a substantial amount of freedom in our choice of $\mathbf{K}$ with which we can attempt to also retain the left eigenvector of the zero-mode. At present, a pseudo-inverse procedure is used to solve for $\mathbf{K}$, which, though it yields a legitimate solution, is not optimal when the steady-state behavior is important. (The pseudo-inverse is optimal in the sense that it yields the smallest possible $\mathbf{K}$ (under the Frobenius norm) that also satisfies the constraint, but it is not clear why that is a particularly relevant optimality criterion).

Let us define some notation to avoid confusion. Let v_r and w_r be the portions of the right and left eigenvectors of the zero-mode corresponding to the relevant variables but computed for the original (full) system. (We drop the subscript 1 from our earlier notation for simplicity since we are only interested in the modal structure corresponding to the zero-mode here). Let $\tilde{v}_r$ and $\tilde{w}_r$ be the right and left eigenvectors of the zero-mode in the reduced system. For the steady-state behavior to be identical for all possible inputs that have a steady-state response, the outer product of $\tilde{v}_r$ and $\tilde{w}_r$ must be identical to the outer product of v_r and w_r. We already require that $\tilde{v}_r$ and v_r have the same shape. For the outer products to be identical, $\tilde{w}_r$ and w_r must also have the same shape. Of course, that will only guarantee that the resulting outer products are equal to within a scale factor, and we return to this issue shortly, but for now we attempt to simply match w_r with $\tilde{w}_r$. In the case of a simple swing-equation model, we can predict the shape of the left eigenvector without doing a full eigenanalysis. For now, we approach the problem from a general perspective and return to the specifics of power system examples in a later section.

The shape of $\tilde{w}_r$ can be determined using the following relationship.

$$\tilde{w}_r^{\mathrm{T}} \left(\mathbf{A_{rr}} + \mathbf{A_{rz}}\mathbf{K}\right) = 0 \tag{B.42}$$

$$\tilde{w}_r^{\mathrm{T}} \mathbf{A_{rz}}\mathbf{K} = -\tilde{w}_r^{\mathrm{T}} \mathbf{A_{rr}} \tag{B.43}$$

We thus need to select a $\mathbf{K}$ matrix that satisfies (B.41) exactly and as closely as possible satisfies the following constraint:

$$w_r^{\mathrm{T}} \mathbf{A}_{rz}\mathbf{K} = -w_r^{\mathrm{T}} \mathbf{A}_{rr} \tag{B.44}$$

One way to proceed is to split $\mathbf{K}$ into two parts $\mathbf{K_0}$ and $\tilde{\mathbf{K}}$ such that

$$\begin{aligned} \mathbf{K} &= \mathbf{K_0} + \tilde{\mathbf{K}} && \text{(B.45)} \\ v_z &= \mathbf{K_0} v_r && \text{(B.46)} \\ \tilde{\mathbf{K}} v_r &= 0 && \text{(B.47)} \end{aligned}$$

For convenience, we can find a suitable $\mathbf{K_0}$ using the pseudo-inverse. As long as the appropriate weighted row sum (v_r provides the weights) of each row of $\tilde{\mathbf{K}}$ is zero, the resulting $\mathbf{K}$ in (B.45) is guaranteed to satisfy (B.41). The following approach attempts to create a $\mathbf{K}$ that satisfies (B.44) as well.

We are now in a position to substitute into (B.44) and derive an equation which $\tilde{\mathbf{K}}$ must satisfy:

$$\begin{aligned} w_r^{\mathrm{T}} \mathbf{A_{rz}} \mathbf{K} &= -w_r^{\mathrm{T}} \mathbf{A_{rr}} && \text{(B.48)} \\ w_r^{\mathrm{T}} \mathbf{A_{rz}} (\mathbf{K_0} + \tilde{\mathbf{K}}) &= -w_r^{\mathrm{T}} \mathbf{A_{rr}} && \text{(B.49)} \\ w_r^{\mathrm{T}} \mathbf{A_{rz}} \tilde{\mathbf{K}} &= -w_r^{\mathrm{T}} \mathbf{A_{rr}} - w_r^{\mathrm{T}} \mathbf{A_{rz}} \mathbf{K_0} && \text{(B.50)} \end{aligned}$$

It should be clear that we need $\tilde{\mathbf{K}}$ to satisfy both (B.47) and (B.50). A suitable $\tilde{\mathbf{K}}$ can be easily found using standard linear algebra techniques (treating the matrix as if it was a vector). In most practical situations (i.e., as long as $m > 2$ and $(n - m) > 2$), the problem is underconstrained, which means that multiple solutions exist. We outline one algorithm for finding a solution.

First, find a matrix $\mathbf{U}$, which satisfies the following properties:

$$\begin{aligned} \mathbf{U}^{\mathrm{T}} \mathbf{U} &= \mathbf{I} && \text{(B.51)} \\ \mathbf{U} v_r &= \begin{bmatrix} \|v_r\| \\ 0 \\ \vdots \\ 0 \end{bmatrix} && \text{(B.52)} \end{aligned}$$

(A suitable matrix (which is essentially a rotation) can be calculated using the "qr" function in MATLAB). Then, we make the following useful definition:

$$\breve{\mathbf{K}} = \tilde{\mathbf{K}}\mathbf{U}^{\mathrm{T}} \tag{B.53}$$

Using this new matrix, we revisit the constraints on $\tilde{\mathbf{K}}$:

$$\begin{aligned} \tilde{\mathbf{K}}v_r &= 0 && \text{(B.54)} \\ \tilde{\mathbf{K}}\mathbf{U}^{\mathrm{T}}\mathbf{U}v_r &= 0 && \text{(B.55)} \\ \breve{\mathbf{K}}\begin{bmatrix} \|v_r\| \\ 0 \\ \vdots \\ 0 \end{bmatrix} &= 0 && \text{(B.56)} \end{aligned}$$

This result suggests a straightforward way to incorporate this constraint into the structure of $\breve{\mathbf{K}}$:

$$\breve{\mathbf{K}} = \begin{bmatrix} 0 & \\ \vdots & \breve{\mathbf{K}}_{\mathbf{1}} \\ 0 & \end{bmatrix} \tag{B.57}$$

The other constraint equation is easily adapted to $\breve{\mathbf{K}}$ as well:

$$w_r^{\mathrm{T}}\mathbf{A_{rz}}\breve{\mathbf{K}} = -w_r^{\mathrm{T}}\mathbf{A_{rr}}\mathbf{U}^{\mathrm{T}} - w_r^{\mathrm{T}}\mathbf{A_{rz}}\mathbf{K_0}\mathbf{U}^{\mathrm{T}} \tag{B.58}$$

Using (B.57), we can substitute into (B.58) and directly solve for $\breve{\mathbf{K}}_{\mathbf{1}}$ using (for example) the pseudo-inverse. We then obtain $\tilde{\mathbf{K}}$ by a simple transformation:

$$\tilde{\mathbf{K}} = \breve{\mathbf{K}}\mathbf{U} \tag{B.59}$$

With the selection of the $\mathbf{K}$ matrix as outlined above, the modal structure for the zero mode in the reduced model is retained up to a scaling factor. The scaling factor is not unity and it is impossible to eliminate, as we show below.

Looking at the eigenvectors of the zero-mode for the full system, v and w, we see that their inner product must be one, since $\mathbf{W}^{\mathrm{T}} = \mathbf{V}^{-1}$. To explore the scaling factor, we will assume a particular scaling of w, call it $\hat{w}$, and allow v to be $c\hat{v}$, where c is our scaling factor and $\hat{v}$ is a particular scaling of v. Then we can derive the following relationship defining c:

$$1 = \hat{w}^{\mathrm{T}} v \tag{B.60}$$

$$1 = \hat{w}^{\mathrm{T}} c\hat{v} \tag{B.61}$$

$$1 = c\langle \hat{v}, \hat{w} \rangle \tag{B.62}$$

$$1 = c \sum_{i=1}^{n} \hat{w}_i \hat{v}_i \tag{B.63}$$

$$c = \frac{1}{\sum_{i=1}^{n} \hat{w}_i \hat{v}_i} \tag{B.64}$$

We can find a similar constant $\tilde{c}$ for the reduced system:

$$1 = \hat{w}_r^{\mathrm{T}} v_r \tag{B.65}$$

$$1 = \tilde{c}\langle \hat{v}_r, \hat{w}_r \rangle \tag{B.66}$$

$$\tilde{c} = \frac{1}{\sum_{i=1}^{m} \hat{w}_i \hat{v}_i} \tag{B.67}$$

The two constants must be identical for the outer products to be identical. (Note that the reduced system uses truncated versions of the "hatted" vectors above). Therefore, the steady-state of the reduced system is the steady-state of the full system multiplied by $\frac{\tilde{c}}{c}$ or $\frac{\sum_{i=1}^{n} \hat{w}_i \hat{v}_i}{\sum_{i=1}^{m} \hat{w}_i \hat{v}_i}$. So, the steady-state vector has the right shape, but the magnitude is not retained.

The only way to completely retain the steady-state behavior of the full system while retaining the zero mode and its modal shape is to also modify b_r. One relatively simple approach is to scale b_r by $\frac{\sum_{i=1}^{m} \hat{w}_i \hat{v}_i}{\sum_{i=1}^{n} \hat{w}_i \hat{v}_i}$. If the $\mathbf{K}$ is selected as suggested above and the b_r is scaled, the steady-state behavior is identically matched.

B.1.2.2 Linear, Swing-Equation Example

We treat the linear, *damped* swing-equation model. While we generally concern ourselves only with *undamped* swing-equation models, our numerical example here is a damped, swing-equation model because we would like the system response to settle to a steady-state. Our real models do have damping in them, but the damping terms are non-trivial to calculate.

In practice, SME has other further restrictions which we discuss later; the present example is only designed to illustrate our results so far and is not designed to necessarily reflect SME in a literal sense.

A damped swing-equation model can be represented by the following equation:

$$\begin{bmatrix} \dot{\delta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{M}^{-1}\mathbf{F} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix} \begin{bmatrix} \delta \\ \omega \end{bmatrix} \tag{B.68}$$

where δ is a vector of generator internal angles, ω is a vector of speeds, $\mathbf{M}$ is a diagonal matrix of machine inertias, $\mathbf{F}$ is an admittance matrix and $\mathbf{D}$ is a diagonal matrix of damping factors. (Note that $\mathbf{M}^{-1}\mathbf{F}$ is essentially our core matrix from Chapter 3). We can now partition the system and retain a subset of the state variables, say δ_r and ω_r.

As noted in Chapter 3, a full eigenanalysis of this system is not necessary to determine the structure of the zero-mode. (This system has a zero-mode because the rows of $\mathbf{F}$ sum to zero). The left and right eigenvectors are as follows:

$$v^{\mathrm{T}} = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \end{bmatrix} \tag{B.69}$$

$$w^{\mathrm{T}} = \begin{bmatrix} D_1 & \cdots & D_n & M_1 & \cdots & M_n \end{bmatrix} \tag{B.70}$$

Therefore, the scaling factor by which b_r needs to be scaled is given by the following expression:

$$\frac{\tilde{c}}{c} = \frac{\sum_{i=1}^{n} D_i}{\sum_{i=1}^{m} D_i} \tag{B.71}$$

As an example, we take a tenth-order system (five generators), reduce it to a sixth-order system (three generators), and compare the reductions based on using different $\mathbf{K}$. We use $\mathbf{K_0}$, $\mathbf{K} = \mathbf{K_0} + \tilde{\mathbf{K}}$ and that same $\mathbf{K}$ but with the additional scaling of b_r as well. As promised, the steady-state of the last correction exactly matches the steady-state of the full model. We present a representative simulation in Figure B.1. In this example, the "new" $\mathbf{K}$ (our second case) performs marginally better (in terms of steady-state) than the $\mathbf{K_0}$ correction, but that is not true generally.

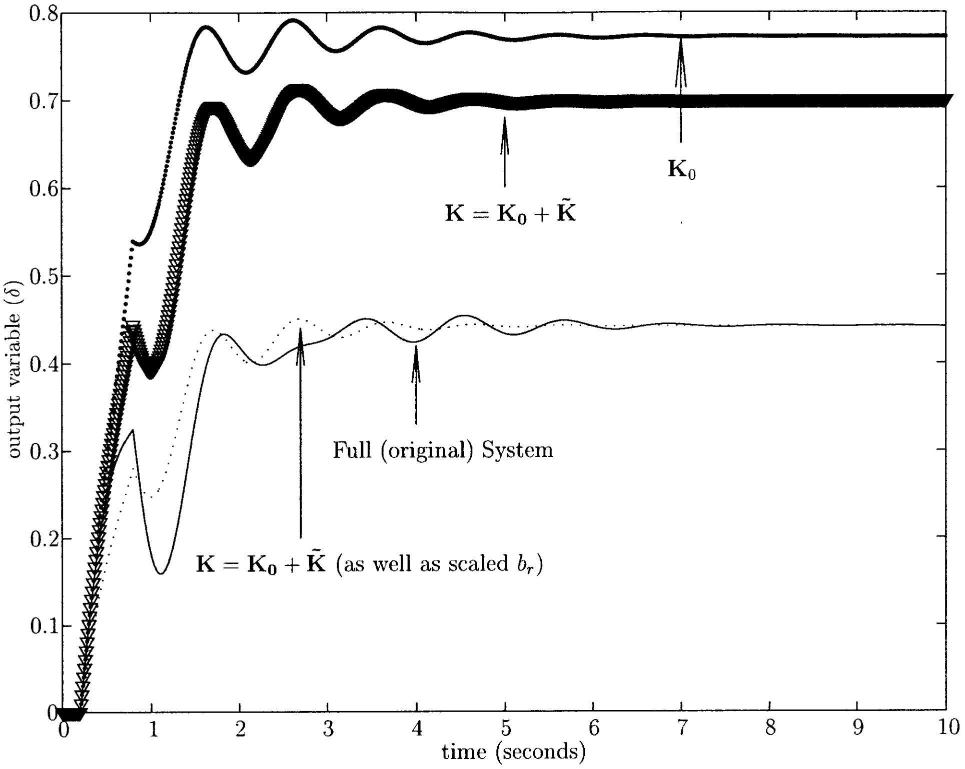


Figure B.1: **Steady-State Matching for Damped Swing-Equation Model.** Plot of the response of one state variable (the angle of one of the generators) of a simple tenth-order state-space model (five generators) to a pulse input, as well as the response of the same variable in three related reduced-order models (sixth-order). We show the reduction currently implemented ($\mathbf{K_0}$), a modification to the correction matrix ($\mathbf{K}$), and this modification along with a rescaling of the input disturbance.

B.1.2.3 Basis Generator Correction $\mathbf{K_b}$

In reality, we do not allow ourselves to represent our less-relevant variables as a linear combination of all of the retained variables. Rather, some subset of the retained variables are designated as a basis and our linear combination is only allowed to involve these variables. The other variables are designated "study area variables" and they are retained but not used in the equivalencing of the less-relevant variables. Therefore, we need to designate our partitioning of the system variables differently.

We use the notation x_s for the vector of study area variables, x_b for the vector of basis variables and x_z for the vector of less-relevant variables. We use $\mathbf{K_b}$, as opposed to $\mathbf{K}$, to emphasize that the equivalencing uses only the basis generators. Thus our original system can be written as follows:

$$\begin{bmatrix} \dot{x}_s \\ \dot{x}_b \\ \dot{x}_z \end{bmatrix} = \begin{bmatrix} \mathbf{A_{ss}} & \mathbf{A_{sb}} & \mathbf{A_{sz}} \\ \mathbf{A_{bs}} & \mathbf{A_{bb}} & \mathbf{A_{bz}} \\ \mathbf{A_{zs}} & \mathbf{A_{zb}} & \mathbf{A_{zz}} \end{bmatrix} \begin{bmatrix} x_s \\ x_b \\ x_z \end{bmatrix} + \begin{bmatrix} b_r \\ 0 \\ 0 \end{bmatrix} u \tag{B.72}$$

The reduced system is then given by:

$$\begin{bmatrix} \dot{s} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \mathbf{A_{ss}} & \mathbf{A_{sb}} + \mathbf{A_{sz}K_b} \\ \mathbf{A_{bs}} & \mathbf{A_{bb}} + \mathbf{A_{bz}K_b} \end{bmatrix} \begin{bmatrix} s \\ z \end{bmatrix} + \begin{bmatrix} b_r \\ 0 \end{bmatrix}, \tag{B.73}$$

where $\mathbf{K_b}$ is constrained to satisfy

$$v_z = \mathbf{K_b} v_b. \tag{B.74}$$

We can now proceed analogously to the method used for the "full **K**". However, there is an extra complication in this case. It is not, in general, possible to match the shape of the left eigenvector by manipulating $\mathbf{K_b}$ the way we did previously. The following equations demonstrate why this is so:

$$\begin{bmatrix} w_s^{\mathrm{T}} & w_b^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{A_{ss}} & \mathbf{A_{sb}} + \mathbf{A_{sz}K_b} \\ \mathbf{A_{bs}} & \mathbf{A_{bb}} + \mathbf{A_{bz}K_b} \end{bmatrix} = 0 \tag{B.75}$$

$$\begin{bmatrix} w_s^{\mathrm{T}}\mathbf{A_{ss}} + w_b^{\mathrm{T}}\mathbf{A_{bs}} & w_s^{\mathrm{T}}\mathbf{A_{sb}} + w_b^{\mathrm{T}}\mathbf{A_{bb}} + w_s^{\mathrm{T}}\mathbf{A_{sz}K_b} + w_b^{\mathrm{T}}\mathbf{A_{bz}} \end{bmatrix} = 0 \tag{B.76}$$

The first entry in the vector in (B.76) is either equal to zero (unlikely) or it is not. We cannot influence it with our choice of $\mathbf{K_b}$. However, one can attempt to minimize the second entry by choosing $\mathbf{K_b}$ in a manner analogous to that used to create **K** above. Of course, such a choice of $\mathbf{K_b}$ does not guarantee anything about the structure of the actual left eigenvector in the reduced model.

We have tested this scheme on real power system data (not just a swing-equation model) within the context of the SME reduction with some additional constraints. The results have been mixed, and it is not clear whether the extremely limited modification to the correction matrix is effective overall or not. As clarification, we note that our dynamic equivalent matrix ($\mathbf{K_b}$) is created defining x_b to be only the angle and speed variables of the basis generators. Since the right eigenvector of the zero-mode is zero everywhere except for the angle variable of each generator, the $\mathbf{K_b}$ calculated using a pseudo-inverse with no left eigenvector correction is extremely sparse. Specifically, it has equal entries for each of the terms relating a less-relevant generator angle to a basis generator angle and is zero elsewhere. When we attempt to match the left eigenvector, we restrict ourselves in this spirit to only allow nonzero terms which relate angles and speeds of generators, though we do not necessarily have to apply such a restriction theoretically. However, practically such a restriction is necessary because of limitations on the number of inputs to a current injector in EUROSTAG. In addition, we did some testing on small models (where such a correction is feasible), but many of the resulting reduced-order models were unstable.

B.1.2.4 Summary

We restrict ourselves to only using information about the zero-mode of our system to do the dynamic equivalencing in SME. The reasons for this are largely practical; such equivalents involve a small subset of the dynamic variables in the generators (i.e., we do not have problems with limitations on current injectors in EUROSTAG) and the right eigenvector corresponding to the zero eigenvalue is known *a priori* from physical principles which obviates an eigenanalysis of the full system. As the previous subsection shows, preserving the full chord may not be desirable in SME anyway.

We have tried to show how information about the left eigenvector of the zero-mode could be incorporated into an equivalent formed using only the right eigenvector of the zero-mode. We have done this from the perspective of a state space model with an input and trying to match the steady-state response of such a system. The actual simulations are typically done on the nonlinear system and involve topological changes to the system (e.g., opening a line), and it is not clear how best (or at all) to represent such events in terms of the linearized model. Unfortunately, our knowledge of the nonlinear system is limited, as are the tools available to perform a realistic analysis on it.

When we apply SME, we further restrict ourselves by only building our equivalent based on the basis generator dynamics and not on the dynamics of the full study area. As noted earlier, very good reasons for this exist; however, that seriously limits our ability to match the steady-state behavior of the system, even for the linear model with scaling of the input.

B.1.3 Error Analysis Conclusions

A detailed analysis of the discrepancies between the simulation results produced by the reduced-order SME model and the original full model is still beyond us. In practice, we obtain very promising results, but the heuristic nature of the procedure makes a theoretical analysis difficult. However, in this section, we attempt to clarify at least some of the situation in an idealized setting. We limit ourselves by only treating "exact" synchrony (for the full mode correction case); our analysis is also only valid for the linearized model.

If we attempt to retain all of the modes in the chord without performing a change of variables, we can describe the error induced by the reduction and unfortunately this error will contain all of the modes of the system, those in the chord as well as those not in the chord. This result is important in particular for the steady-state response of the system, since it guarantees that some steady-state error will exist. On the other hand, we can correct for only a portion of the modes in the full chord and the resulting degrees of freedom can be used to optimize particular parts of the response, such as the steady-state values. However, because we use only basis generators to create our dynamic equivalents, we cannot even achieve a guaranteed steady-state response matching.

The results here are not completely negative. We have explored this topic seriously for the first time, and some avenues of inquiry have been opened. In particular, modifying the matrix $\mathbf{K_b}$ to optimize the reduction in certain ways, let alone using the left eigenvector of the zero-mode, is a new development, which opens avenues for further exploration. In addition, the result in (B.76) has been useful in research into determining the optimal number of groupings for any particular system under the SME algorithm. If a particular number of groupings leads to a smaller first term in (B.76), we consider that grouping to be better, based on our desire for this term to be zero in order to match the left eigenvector of the zero-mode. This term is only exactly zero if the system is completely decoupled into at least two separate areas, which also coincides nicely with our use in the context of optimum groupings.

B.2 "Robustness" of the SME partition

Our analysis of the error in SME was concerned with a single, nominal (original) system model. However, we are primarily concerned in this thesis with parameter uncertainties, which give rise to many different original system models. In other words, though the topology and form of the original model are static, changes to the parameters give rise to linearized models with differing values in the system matrices. This section attempts to address this issue, at least in part.

While we expect (and in fact need) differing sets of original parameters to yield differing reduced-order models, we would like the changes to the reduced-order models to be strictly limited to parameter changes within the reduced-order model. This property holds if and only if the same basis generators and study area are selected in all cases. We would like to find results that suggest that the SME partitioning algorithm is relatively insensitive to parameter changes in the original model. We refer to this insensitivity as "robustness". Of course, if the original parameters are changed sufficiently, a new partition is induced. Therefore, most of our results are asymptotic.

We approach this problem in two ways: empirically and theoretically. Our empirical studies involve changing system parameters in a model and observing whether such changes affect the partition created by SME. Our theoretical studies focus on the robustness of our algorithms in the context of exact synchrony within the system. In other words, we wish to avoid losing ourselves in the details of how to identify approximate synchrony and focus on the underlying issue of whether the concept of synchrony itself has good properties, and by doing so we also do not limit the applicability of our analysis by limiting ourselves to one particular chord selection method. In addition, we do not discuss the details of the algorithms used to select the basis generators or to partition the system, but rather our work is an analysis of how these algorithms should perform based on their stated objectives. In this way, we avoid a potentially distracting tangent to the main focus of this thesis as well as broadening the applicability of our results, since many of these algorithms are being modified by present research investigations.

B.2.1 Eigenvector Sensitivity

Since the chord selection, basis selection and system partitioning are all performed based solely on the eigenvectors (left and right) of the core and extended core matrices, these matrices and the corresponding eigenvectors are the focus of our study. For purposes of clarity, we limit ourselves to discussing just the core model. Only the load node partitioning relies on the extended core matrix and our discussion of the generator partitioning generalizes directly to encompass the load node partitioning and the extended core matrix without any complications, and we omit the details solely for simplicity of presentation.

To further simplify the problem, we assume that the core matrix is symmetric. All of our algorithms are based on analyzing participation factors as opposed to right eigenvectors, since participation factors are scale-invariant. However, while participation factors are an extremely useful concept, their definition and relationship to the essential behavior of a system are not nearly as well developed as those for eigenvectors. If the core matrix is symmetric, we sidestep this issue, since the participation factors in this case are the term by term squares of the entries of the right eigenvector matrix. In this case, exact synchrony between two variables with a given chord measured using participation factors exists if and only if the right eigenvectors also exhibit such synchrony. Therefore, we only treat the right eigenvectors.

In summary, we are concerned with a symmetric matrix $\mathbf{F}$ which describes the small-signal internal angle dynamics of the generators:

$$\Delta\ddot{\delta} = \mathbf{F}\Delta\delta \tag{B.77}$$

where we use $\mathbf{F}$ in analogy to (3.13) under the assumption of equal generator inertias. In accord with our standard notation, we will denote the kth left and right eigenvectors corresponding to eigenvalue λ_k by v_k and w_k^{T}. We can also compute the kth right eigenvector of the associated matrix $\hat{\mathbf{F}}$:

$$\hat{\mathbf{F}} = \mathbf{F} + \epsilon\tilde{\mathbf{F}} \tag{B.78}$$

and we shall refer to this new right eigenvector as $v_k(\epsilon)$ and it can be described by [61]:

$$v_k(\epsilon) = v_k + \epsilon \sum_{i=1, i \neq k}^{n} \left\{ \frac{w_i^{\mathrm{T}} \tilde{\mathbf{F}} v_k}{(\lambda_k - \lambda_i) w_i^{\mathrm{T}} v_i} \right\} v_i + \mathcal{O}(\epsilon^2) \tag{B.79}$$

Similar results exist for the generalized eigenvector problem [62].

A few properties in particular of (B.79) are important for us. First, the eigenvector sensitivities are well-behaved for sufficiently small ϵ, but only under the condition that the associated eigenvalue is not repeated. This result is not surprising when we consider that, if a repeated eigenvalue exists, the corresponding eigenvectors form an invariant subspace, but the basis vectors chosen to represent the subspace are arbitrary. In other words, if a repeated eigenvalue exists, the corresponding right eigenvectors are not uniquely defined. If two eigenvalues are arbitrarily close in value, the sensitivity becomes large because the indeterminacy of the repeated eigenvalue case can be "felt".

Second, the kth right eigenvector sensitivity equation (to first order) is a weighted sum of all of the eigenvectors of the system. In addition, this relationship can be written row by row. Of course, the weighting term is also dependent on the eigenvectors, so the sensitivity of the first entry in any eigenvector may depend on all of the entries in all of the eigenvectors and, due to the presence of the left eigenvector, the relationship is not necessarily linear.

B.2.2 Synchrony

We revisit our definition of synchrony from Section 3.3.2 through the use of a simple three generator example. Since our basis generator selection and partitioning attempt to recognize and bring out any inherent synchrony within the system, we need to examine the relationship between our eigenvector sensitivity result and the synchrony within the system.

Let our matrix $\mathbf{F}$ be given as the sum of two matrices, $\mathbf{F} = \mathbf{A} + \mathbf{E}$, where:

$$\mathbf{A} = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix} \tag{B.80}$$

$$\mathbf{E} = \begin{bmatrix} -.01 & .01 & 0 \\ .01 & -.02 & .01 \\ 0 & .01 & -.01 \end{bmatrix} \tag{B.81}$$

If we look at the right eigenvectors of $\mathbf{F}$, the synchrony within the system becomes clear:

$$\mathbf{V} = \begin{bmatrix} .5774 & -.7071 & .4082 \\ .5774 & 0 & -.8165 \\ .5774 & .7071 & .4082 \end{bmatrix} \tag{B.82}$$

If we look at the chord corresponding to the first and third column, we see that the first row (generator) is synchronous with the third row (generator) because one row is a constant multiple of the other. The need to look at rows of the right eigenvector matrix as opposed to columns makes the analysis somewhat trickier because the columns, not the rows, of the matrix are the eigenvectors.

Let us now look at a closely related matrix $\mathbf{F}' = \mathbf{A} - \mathbf{E}$, which is almost numerically identical to $\mathbf{F}$. If we examine its right eigenvector matrix, we discover something startling:

$$\mathbf{V}' = \begin{bmatrix} .5774 & -.7071 & .4082 \\ .5774 & .7071 & .4082 \\ .5774 & 0 & -.8165 \end{bmatrix} \tag{B.83}$$

If we look at either the chord (1,3) or the chord (1,2), the previous synchrony between the first and the third generator is no longer present! No matter what reasonable measure is used for approximate synchrony with respect to any nontrivial chord, it should be apparent that these two rows cannot represent synchronous generators. In addition, using chord (1,3), we see that the first and second generator are now synchronous, which was in no way apparent in $\mathbf{F}$.

The answer to this riddle lies in a full eigenanalysis of $\mathbf{A}$. We see that $\mathbf{A}$ has an eigenvalue at zero and a repeated eigenvalue at -3. We can also calculate the eigenvalues of $\mathbf{F}$ and $\mathbf{F}'$, which are in both cases (0,-3.01,-3.03). Both matrices have a nearly repeated eigenvalue, and if we look at our definitions for these matrices, we see that both matrices satisfy the following relationship:

$$\mathbf{F}(\xi) = \mathbf{A} + \xi\mathbf{E} \tag{B.84}$$

In particular:

$$\begin{aligned} \mathbf{F}(\mathbf{1}) &= \mathbf{F} && \text{(B.85)} \\ \mathbf{F}(-\mathbf{1}) &= \mathbf{F}' && \text{(B.86)} \\ \mathbf{F}(\mathbf{0}) &= \mathbf{A} && \text{(B.87)} \end{aligned}$$

Therefore, the transition from $\mathbf{F}$ to $\mathbf{F}'$ has caused us to pass through the repeated eigenvalue case; the result is of course that the eigenvectors are not "well-behaved".

Unfortunately, this result indicates that SME may not be robust (at all) if we choose a repeated eigenvalue for our chord. More importantly, even if there are no repeated eigenvalues (which is generally true for real systems), the synchrony within the system (and therefore the partition) may change radically due to only relatively small parameter changes even though the synchrony is of course robust for vanishingly small parameter changes. This example makes clear a related property: though the individual eigenvectors may be sensitive, the invariant subspace itself is relatively insensitive to small perturbations as long as the (nearly) repeated eigenvalue is separated from the other eigenvalues in the system. [61]

Of course, one straightforward way around this limitation is to not choose any eigenvalue for the chord which is not sufficiently separated from the other eigenvalues in the system. So far in practice, we have fortunately not encountered a situation where a nearly repeated eigenvalue was desirable for the chord. However, not much can be done if the user is intent on choosing such an eigenvalue for the chord. If only two eigenvalues are near each other, the approximately invariant subspace defined by their eigenvectors may well be insensitive to parameter changes, but that is small comfort. For any particular choice of parameters for which the two eigenvalues are not identical, the eigenvectors are extremely sensitive to parameter changes, but they cannot be arbitrarily chosen to span the subspace; they are uniquely defined. We would counsel that the best solution is to simply avoid such eigenvalues as candidates for the chord, or, minimally, to verify with representative sets of parameter values that the resulting changes to **F** are not great enough to cause radical changes to the associated eigenvector(s).

B.2.3 Non-Asymptotic Results

If the chord does not contain any of the repeated eigenvalues in the system, the grouping of the nodes used for the SME reduction is insensitive to small changes of the original parameters. However, our parameter changes are never vanishingly small. Our last example showed that, though the nominal system **F** had no repeated eigenvalues, a relatively small change to the parameters of the system could radically change the synchrony within the system because **F** has nearly repeated eigenvalues. In this section, we explore the other case where synchrony may be disturbed by relatively small but non-vanishing changes to the original parameters, but unlike the case of nearly identical eigenvalues, our present SME algorithms effectively preclude problems.

Again, we use an example to motivate the discussion. We define two matrices, **A** and **E**:

$$\mathbf{A} = \begin{bmatrix} -1.7498 & 0.2502 & 0.7639 & 0.7356 \\ 0.2502 & -1.7498 & 0.7639 & 0.7356 \\ 0.7639 & 0.7639 & -1.2785 & -0.2494 \\ 0.7356 & 0.7356 & -0.2494 & -1.2218 \end{bmatrix} \tag{B.88}$$

$$\mathbf{E} = \begin{bmatrix} -0.0070 & 0.0000 & 0.0000 & 0.0070 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0070 & 0.0000 & 0.0000 & -0.0070 \end{bmatrix} \tag{B.89}$$

Let **F** = **A**. The eigenvalues of **F** are (0,-1,-2,-3), so there is no issue with nearly identical eigenvalues. If we look at the eigenvectors of the system (represented by matrix **V**), we can observe synchrony within this system:

$$\mathbf{V} = \begin{bmatrix} 0.5000 & 0.0100 & 0.7071 & 0.4999 \\ 0.5000 & 0.0100 & -0.7071 & 0.4999 \\ 0.5000 & 0.6970 & 0 & -0.5140 \\ 0.5000 & -0.7170 & 0 & -0.4858 \end{bmatrix} \tag{B.90}$$

Looking at the chord represented by the first, second and fourth columns of $\mathbf{V}$, we see that the first and second generator are synchronous (actually, coherent). We now examine $\mathbf{F}' = \mathbf{A} + \mathbf{E}$ and its eigenvectors (the eigenvalues are nearly identical):

$$\mathbf{V}' = \begin{bmatrix} 0.5000 & -0.0062 & 0.7047 & 0.5034 \\ 0.5000 & -0.0113 & -0.7095 & 0.4965 \\ 0.5000 & -0.6982 & 0.0050 & -0.5123 \\ 0.5000 & 0.7157 & -0.0002 & -0.4876 \end{bmatrix} \tag{B.91}$$

The synchrony is still present at least approximately, although the second column no longer has the correct structure. The first and second entries of the second eigenvector are small, so they should not have a large effect on a reasonable definition of approximate synchrony. In fact, our measure of synchrony uses the angle between two rows when treated as vectors (taking only the columns corresponding to the chord). In our example, the angle between the following two vectors is used:

$$\begin{bmatrix} 0.5 \\ -0.0062 \\ 0.5034 \end{bmatrix}, \begin{bmatrix} 0.5 \\ -0.0113 \\ 0.4965 \end{bmatrix} \tag{B.92}$$

Our algorithm protects us from some robustness issues by using such a definition because the smaller entries affect the angle less.

The relative sensitivity of small entries in an individual eigenvector is not unexpected based on (B.79). If we ignore the higher-order terms in ϵ and note that $w_i^{\mathrm{T}} v_i = 1$, we obtain the following:

$$v_k(\epsilon) - v_k = \epsilon \sum_{i=1, i \neq k}^{n} \left\{ \frac{w_i^{\mathrm{T}} \tilde{\mathbf{F}} v_k}{(\lambda_k - \lambda_i)} \right\} v_i \tag{B.93}$$

The $w_i^{\mathrm{T}}\tilde{\mathbf{F}}v_k$ term is a scalar and depends on all of the entries in v_k and w_i^{T} as well as the location and size of the changes to the parameters ($\tilde{\mathbf{F}}$). The two vectors are each of length 1, due to our normalization. Every row of each v_i in the sum is multiplied by this same scalar. If a particular entry in v_k is small and the corresponding entries in some of the v_i are large, the relative sensitivity of that entry may be large as long as the scalar terms which multiply those v_i are not small. In other words, an appropriately selected $\tilde{\mathbf{F}}$ that is not vanishingly small may greatly affect the smaller terms in our eigenvector matrix.

SME is protected from this possibility in at least two ways. First, we select a chord which is "extensive". One effect of such a choice is that most of the generators in the system have a nontrivial participation in the modes of the chord. In other words, we hope (and have reason to expect) that very few of the entries in the right eigenvector matrix (in the symmetric case) used for basis generator selection and partitioning have small magnitudes. Second, one of our criteria for selecting basis generators is the amount of participation in the chord. If a certain generator was selected as a basis generator, but many of its right eigenvector entries in the chord have small magnitudes, the partitioning would be highly sensitive to parameter changes for obvious reasons. Other generators are grouped to basis generators based on synchrony considerations and, if the basis generator entries changed radically due to parameter changes, the partition would also be likely to change radically.

B.2.4 "Empirical" Studies and Conclusions

We have performed some limited studies to verify the analytical results above. All of our studies have been on a model which is roughly equivalent to the France-Spain power grid. This model is already highly equivalenced. The system and the nominal partition into five groups is shown in Figure B.2. We concentrate only on one parameter of the real part of the static load model in this system. Nominally, the same load model is used at every bus in the system. The real power part of the load model is given by the following:

$$P = P_0|V|^{\alpha}\omega^{\gamma} \tag{B.94}$$

where P is the real power consumed, $|V|$ is the voltage magnitude at the corresponding bus, ω is the system frequency at that bus and α and γ are modeling parameters. The parameter α is nominally 2.0 and we changed it by as much as 100% (i.e., $\alpha = 1.0$) on each

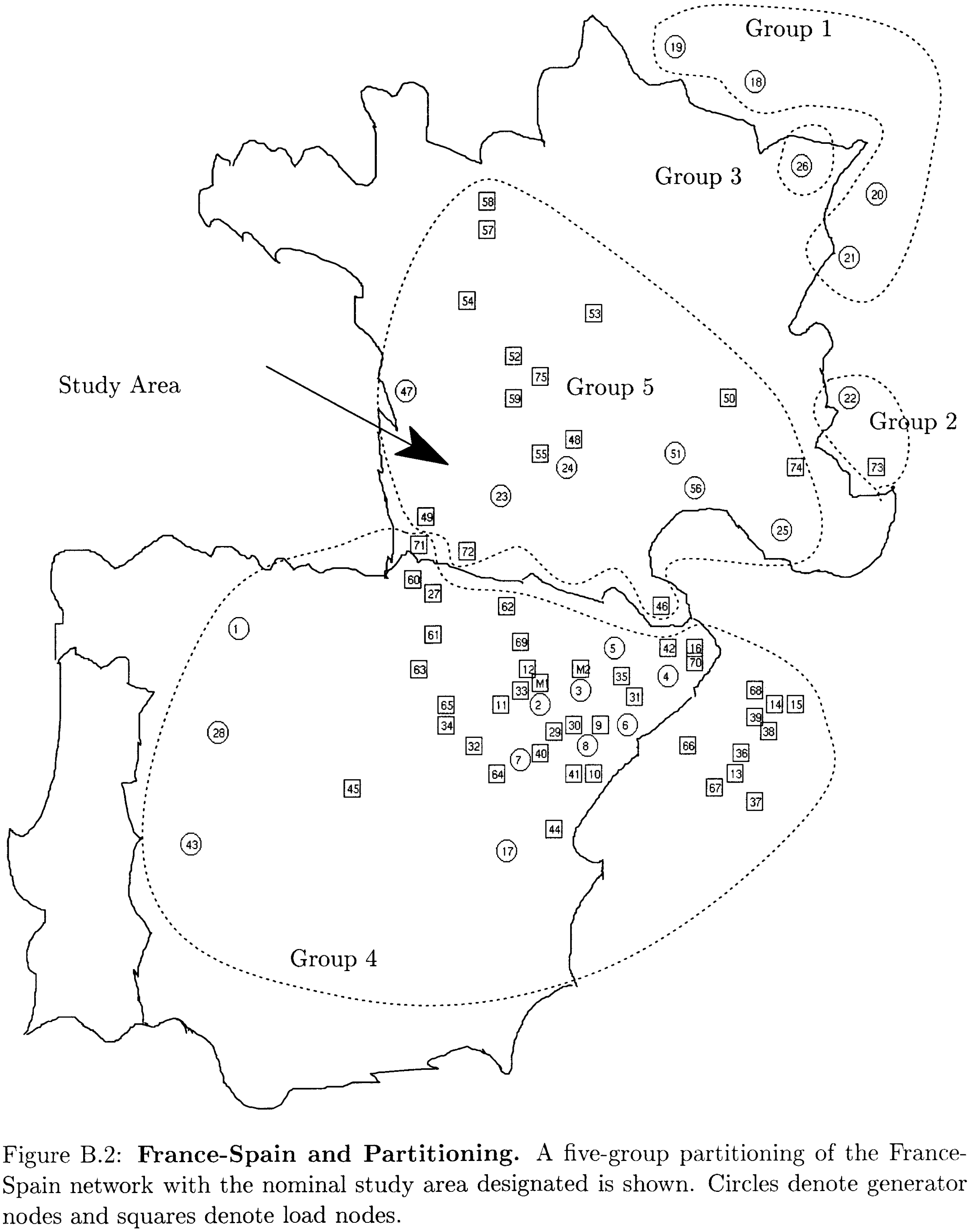


Figure B.2: **France-Spain and Partitioning.** A five-group partitioning of the France-Spain network with the nominal study area designated is shown. Circles denote generator nodes and squares denote load nodes.

Bus Number	Nominal Group	"New" Group
19	1	2
42	4	5
73	2	5

Table B.1: **Partitioning Changes Induced by Load Parameter Variations.** A summary of the changes induced by varying load parameter α on the France-Spain network.

Bus Number	Nominal Group	Next "closest" Group
19	.9858	.9844
42	.8633	.7902
73	.7776	.64753

Table B.2: **Degrees of Synchrony of Affected Nodes.** The degree of synchrony listed under nominal is that between the given node and its basis generator with the nominal α. The third column lists the next largest degree of synchrony between the given node and the rest of the basis generators with the nominal parameters.

bus of the system (while retaining the original model parameters for the other buses) and also on selected groups of buses.

Overall, the results verify our analytical results. We were only able to induce three (separate) changes to the system partitioning, which are summarized in Table B.1. Bus 19 is a generator bus while the other two buses are load nodes. One important thing to note is that out of eighty-three buses (some of the buses in the figure are shown as a single number) only three buses ever changed groups. In addition, these buses were not "strongly grouped", in the sense that these three buses did not exhibit good approximate synchrony with any of the basis generators. We can see that by looking at the degrees of synchrony (a measure of approximate synchrony) of each of these buses with their basis generators. A degree of synchrony of one indicates exact synchrony, and the measure varies between zero and one. Degrees of synchrony between generators in a group and their basis generator are often 0.99 or greater. For load nodes, the degree of synchrony is often greater than 0.9. The appropriate degrees of synchrony for each of these three nodes is given in Table B.2. Therefore, it is not surprising that a parameter change might affect the grouping of these nodes.

In summary, if the chord does not contain any of the repeated or nearly repeated eigenvalues in the system, SME should be fairly robust to parameter changes. We are working under the assumption that the SME chord selection, basis selection and partitioning algorithms are all functioning in a way consistent with their stated goals. We have verified this conclusion both analytically in an ideal setting and also empirically.

B.3 Summary

In this appendix, we touch upon two basic issues that help determine whether a reduction method is appropriate for the purposes of uncertainty analyses: error and robustness. Obviously, if the reduced-order model does not faithfully reproduce the behavior of the full system, its utility is limited. We cannot expect perfect reproduction, but we cannot allow for excessive discrepancies either. Similarly, we are interested in the behavior of the system when its parameters are modified. If the reduced-order model form changes radically when the parameters of the original model are changed, each new set of parameters in the full model necessitates a new reduction, which does not allow us to amortize the computation cost of computing the reduction over the course of our uncertainty analysis. Of course, the parameters of the reduced model do change, and we return to this issue later in the thesis.

Our approach here is to analyze SME under a set of limiting assumptions. Unfortunately, SME is a heuristic reduction method comprising individual heuristic algorithms, which makes analysis difficult. However, the methods used are based on underlying theoretical results for a system with ideal structure, which our models only approximate.

The error analysis shows that SME has inherent limitations. We derive modal formulas for the error introduced by SME under very limiting assumptions, but their utility is limited because the practical implementation of SME is far removed from the underlying theory. In particular, we only seek to retain the zero-mode in the system when we do our dynamic equivalencing, which has yielded surprisingly good results but leaves us with little to work with analytically. One positive development is the realization that, by retaining only the zero-mode of the system, we have degrees of freedom to work with within our reduction technique. We explore one possibility: retention of the steady-state behavior. Though this particular result is not practically useful because we only base our equivalents on the basis generators, the methodology is sound. These explorations have also been useful for studies related to determining the optimal number of groups for a particular system.

The robustness analysis shows that the SME partition (which determines the form of the reduced-order model) is robust to changes to the original model. We identify the one case where this is not true even for small parameter changes, namely, if the chord contains one of the (nearly) repeated eigenvalues of the system. In this case, SME may not be particularly robust, and it appears that not much can be done to remedy this situation. In all other cases, SME is robust because of particular goals of the chord selection and basis selection algorithms. These goals were not formulated specifically to enhance robustness, but our analysis shows that they have this additional benefit; this observation is important for current development of the SME algorithm.

Appendix C

Key Uncertain Parameter Identification Program

This appendix contains the MATLAB scripts necessary to do our key uncertainty identification algorithm. We have attempted to liberally comment these scripts to aid the reader. We start with the main program script which is called to start the process of key parameter uncertainty identification and then list each of the functions called by it in order. The individual files are separated by lines.

```
% uncertainty_working.m - Matlab Script
% Last Modified 13 June 2000 by James Hockenberry
% Author: James Hockenberry

% This Matlab script identifies the key uncertain
% parameters in a power system model which has
% already been reduced using SME.
% This is done in two steps: analysis of the external
% reduction in SME and eigenanalysis of the reduced-order
% model.
%
% This program requires the output files of an SME
% reduction 'name'.data and 'namenew'. It will give
% a file 'name'_uncertainty, which contains important
% data, such as a table giving rank-ordered
% lists of important original parameters for each of
% the new parameters in the external reduction. It
% also produces 'name'_eigentable, which contains a
% rank-ordered list of the uncertain parameters in
% the reduced order model from most critical to least
% critical.

% Other scripts and functions called: dataprep,
% eigensense, createtable, externalreductionprep,
% reducedsystemprep, win_for and writetable.
```

```
%**************************************************

% Clear all of the variables and turn off more, so
% we have memory and the program doesn't unexpectedly
% stop.  Also, clear the screen
clear
more off
clc

% Print a list of all of the Matlab files in this
% directory, so the user will know what 'name' and
% 'namenew' files are available.
what
% We need to find out the name of our case.
disp('Please input the name of the case you would like analysed:');
name = input('> ','s');
disp(' ');

disp('Please input the name of the "newname" file');
disp(' generated by reduction:');
namenew = input('> ','s');
disp(' ');

% Use the dataprep function to create all of
% the necessary matrices.
disp('Has the data for this case already been prepared?');
disp('If yes: reply "Y" or "y".  Otherwise: reply anything else');
answer = input('> ','s');
if answer ~= 'Y' & answer ~= 'y'
    tic
    dataprep(name,namenew);
    toc
end

%*********************************************************
% Find out if the external reduction table has already been
% created or not. We are looking at the sensitivity of new
% elements to old (removed) elements.

disp(' ');
disp('Has an external reduction sensitivity table already been ');
disp('created for this case?');
disp('If yes: reply "Y" or "y".  Otherwise: reply anything else');
answer = input('> ','s');
```

```
% If the table hasn't been created yet, run "createtable" to
% create it.
if answer ~= 'Y' & answer ~= 'y'
  tic
  table = createtable(name,namenew);
  toc
else
  eval(['load ',name,'_uncertainty table;']);
end

% Now that we have the table, we can winnow it out and
% only keep the most 'important' old elements for each
% new element.  win_for accomplishes this.
disp(' ');
disp('Has a percentage table already been created for this case?');
disp('If yes: reply "Y" or "y".  Otherwise: reply anything else');
answer = input('> ','s');

if answer ~= 'Y' & answer ~='y'
  disp('All of the entries in the table for a particular new line');
  disp(' will be kept from greatest to least sensitivity.  You can');
  disp(' specify down to what percentage of the maximum sensitivity');
  disp(' for each new line you would like to keep.');
  disp('Please enter your value as a percentage, i.e., 50% should');
  disp(' be entered as "50".');
  percentage = input('> ','s');
  percentage = str2num(percentage);

  % The new_table is the smaller table and howmany is the number of
  % elements kept for each of the new lines.
  [table_new,howmany] = win_for(percentage,table);
  eval(['save ',name,'_uncertainty table_new percentage howmany -append;']);
end

% table can be dauntingly huge (read, sucks up memory), so
% we ditch it for now.
clear global table

%***********************************************************%
% We also have to determine what the most important elements
% in the entire reduced system are.  We will do this by building
% the DAE description of the reduced system and examining
% the sensitivities of its eigenvalues.  If the system is
% highly sensitive to any of the elements in the (now reduced)
% external area, we can use the information gained about the
% external reduction and integrate that information.

% The first task is to perform an eigenanalysis of the DAE model.
% This will be a "generalized" eigenanalysis problem.  We will
% then proceed to compute eigen-sensitivities.
tic
eigentable = eigensense(name,namenew);
toc
eval(['save ',name,'_eigentable eigentable;']);
clear all
```

```
function [] = dataprep(name,namenew);
% We will create all of the necessary initial data for
% our procedures here and save the results in a file called
% 'name'_uncertainty.mat.  That way we can get whatever we
% need afterward and we'll only have to do this step once if
% we wind up running the program more than once on the same
% example for some reason.

%******************************************************
% First, we need to create the matrices necessary to analyse
% the external reduction.

externalreductionprep(name,namenew);

%******************************************************
% Here we are worried about what's "important" for the entire
% reduced order system.  In other words, the external
% reduction is already completed.

reducedsystemprep(name,namenew);
```

```
function [] = externalreductionprep(name,namenew);

% Load in the 'J' Matrix and the 'CONNEX' Matrix
eval(['load ',name,'.data J CONNEX;']);

% Load in the indices of the various types of nodes.
eval(['load ',namenew,' relout nrelnear nrelfar;'])
```

```
% Get the length of the J matrix
Jlength = length(J);

% We only care if the node is “kept” and external and not which
% particular type it is (nrelnear or relout).
kept = sort([nrelnear,relout]);

% After some further investigations it seemed reasonable to get the
% actual matrices used for the reduction.
Jzfzf = J(sort([nrelfar*2,nrelfar*2-1]),sort([nrelfar*2,nrelfar*2-1]));
Jk_zf = J(sort([kept*2,kept*2-1]),sort([nrelfar*2,nrelfar*2-1]));
Jzf_k = J(sort([nrelfar*2,nrelfar*2-1]),sort([kept*2,kept*2-1]));
iJzfzf = inv(Jzfzf);

% k kept nodes, z far nodes (lengths)
k = length(kept);
z = length(nrelfar);

% create the matrix of new lines
J_new = -Jk_zf*iJzfzf*Jzf_k;

% We'll also have a version that is complex for easier access.
twice_total = length(J_new);
G1 = (J_new(1:2:twice_total,1:2:twice_total)+...
   J_new(2:2:twice_total,2:2:twice_total))/2;
G2 = G1';
Y1 = (J_new(1:2:twice_total,2:2:twice_total)-...
   J_new(2:2:twice_total,1:2:twice_total))/2;
Y2 = Y1';
J_new_complex = (G1 + G2)/2 + i*(Y1+Y2)/2;

% We also need to find the amount on the diagonal of Jzfzf after
% we subtract off the amount contributed by lines as well as the
% same thing for J_new.
Jzfzf_sum_odd = sum(J(sort([nrelfar*2-1,nrelfar*2]),1:2:Jlength),2);
Jzfzf_sum_even = sum(J(sort([nrelfar*2-1,nrelfar*2]),2:2:Jlength),2);
J_new_sum_odd = sum(J_new(:,1:2:length(J_new)),2);
J_new_sum_even = sum(J_new(:,2:2:length(J_new)),2);

% “ff” is the number of old lines between far nodes
% Also, we need to find the zf,zf lines, so we can perturb them.
% (Iff,Jff) is a set of all the nonzero entries in the upper triangular
% part of Jzfzf. We restrict ourselves to this triangle because the
% matrix is symmetric.
```

```
conn_zf = CONNEX(nrelfar,nrelfar);
conn_tri_zf = triu(conn_zf);
ff = sum(sum(CONNEX(nrelfar,nrelfar)))/2;
[Iff,Jff] = find(conn_tri_zf);

% "kf" is the number of lines between kept and zf nodes
% Find the kept,zf lines, so we can perturb them.
% (Ikf,Jkf) is similar to (Iff,Jff) but here we don't need to restrict
% ourselves to upper triangular, since conn_kzf isn't symmetric,
% or even square.
conn_kzf = CONNEX(kept,nrelfar);
kf = sum(sum(conn_kzf));
[Ikf,Jkf] = find(conn_kzf);

% Re-order the connection matrix for later use. Order is nrelfar, kept.
connex = [conn_zf, conn_kzf';conn_kzf,zeros(k,k)];
connex = connex + sparse(diag(ones(size(connex,1),1)));

% This will allow us to easily go from the "connex" matrix to
% the real node numbers.
translate = [nrelfar,kept];

%*********************************************************
% This is also part of the process of analysing the external
% reduction.

% Find all of the new lines
% countnewlines is the number of new lines + 1.  (inew,jnew) are
% the indices of the new lines.  (Again being careful not to
% double count due to symmetry).
countnewlines = 1;
for index1 = 1:k
   for index2 = index1+1:k
      if abs(J_new_complex(index1,index2)) ~= 0
         jnew(countnewlines) = index2;
         inew(countnewlines) = index1;
         countnewlines = countnewlines+1;
      end
   end
end

eval(['save ',name,'_uncertainty countnewlines jnew J_new_complex ']);
eval(['save ',name,'_uncertainty J_new_sum_even Jzfzf_sum_odd -append']);
eval(['save ',name,'_uncertainty kf Ikf Jkf ff Iff Jff -append']);
eval(['save ',name,'_uncertainty Jzfzf_sum_even Jzfzf -append']);
eval(['save ',name,'_uncertainty iJzfzf z J_new_sum_odd -append']);
eval(['save ',name,'_uncertainty Jzf_k Jk_zf kept inew -append']);
```

```
function [] = reducedsystemprep(name,namenew);

% The building of the DAE model for the entire reduced system
% is based on (i.e., stolen from) the SME function simuldae.m
% The model will be of the form
%      .
% (Elr)X = (Alr)X, where Elr is a diagonal matrix with entry 1
% for differential equations and 0 for algebraic equations.

% I have tried my best to comment this code.  The original was not
% commented and it wasn't always quite clear what the author's
% intention was.

% We also need the "speed" variables for the reduced system so
% we can find the current injectors later.

% As a final note, we get rid of any differential variables that
% are defined by an all zero equation.

% Load in the data
eval(['load ',name,'.data nbgen nbmotors nbstates speed;']);
eval(['load ',name,'.data blockstate Ad Bd Cd J Y;']);
eval(['load ',namenew,' keptmach refext relin relout nrelnear;']);
eval(['load ',namenew,' nrelfar refbasis Kb;'])

% study gives all the generators in the study area
study=setdiff(keptmach,[refext nbgen+1:nbgen+nbmotors]);
nbgens=length(study);

% Here we're gathering the differential variables into those
% that are kept and those that aren't.
xr=[];
for i=1:length(keptmach)
   prec=sum(blockstate(1:keptmach(i)-1));
   xr=[xr prec+1:prec+blockstate(keptmach(i))];
end
xz=setdiff(1:nbstates,xr);
% We will also need to know where the speed variables are in
% the reduced network
```

```
speedr = find(ismember(xr,speed));

% Here we're gathering relevant in and out voltage variables.
uri = 2*relin;
uri = sort([uri uri-1]);
uro = 2*relout;
uro = sort([uro uro-1]);

% Here we're gathering not relevant near and far variables
uzc = 2*nrelnear;
uzc = sort([uzc uzc-1]);
uzf = 2*nrelfar;
uzf = sort([uzf uzf-1]);

% We create the matrices defined in the SME reports.
Adrr = Ad(xr,xr);
Bdrri=Bd(xr,uri);
Bdrro=Bd(xr,uro);
Bdrzc=Bd(xr,uzc);
Cdrir=Cd(uri,xr);
Cdror=Cd(uro,xr);
Cdzcr=Cd(uzc,xr);
Cdzfz=Cd(uzf,xz);
Cdzcz=Cd(uzc,xz);
Jriri=J(uri,uri);
Jriro=J(uri,uro);
Jrori=J(uro,uri);
Jroro=J(uro,uro);
Jrizc=J(uri,uzc);
Jrozc=J(uro,uzc);
Jrizf=J(uri,uzf);
Jrozf=J(uro,uzf);
Jzcri=J(uzc,uri);
Jzcro=J(uzc,uro);
Jzfri=J(uzf,uri);
Jzfro=J(uzf,uro);
Jzczc=J(uzc,uzc);
Jzczf=J(uzc,uzf);
Jzfzc=J(uzf,uzc);
Jzfzf=J(uzf,uzf);
iJzfzf=inv(Jzfzf);

% Get the generator small signal admittance terms as well.
GenSmallSignal = J([uri,uro,uzc],[uri,uro,uzc]) + ...
```

```
    Y([uri,uro,uzc],[uri,uro,uzc]);
GenSmallSignalFar = J(uzf,uzf) + Y(uzf,uzf);

% Create the K correction matrix. We already have Kb, so we just
% have to put the entries on the appropriate speed and angle positions.
K=sparse(length(xz),length(xr));
blockstudy=blockstate(keptmach);
TETAS=[];
for i=1:length(refbasis)
  j=find(keptmach==refbasis(i));
  TETAS(i)=sum(blockstudy(1:j));
end
K(:,sort([TETAS TETAS-1]))=Kb;
% We'll also want to know where the speed and angle of the basis
% machines are.
BasisSpeedAndAngle = sort([TETAS TETAS-1]);

% Create the correction terms due to the external reduction
CdrozK=-Jrozf*iJzfzf*Cdzfz*K;
CdzczK=(Cdzcz-Jzczf*iJzfzf*Cdzfz)*K;
Jnroro=-Jrozf*iJzfzf*Jzfro;
Jnrozc=-Jrozf*iJzfzf*Jzfzc;
Jnzcro=-Jzczf*iJzfzf*Jzfro;
Jnzczc=-Jzczf*iJzfzf*Jzfzc;

% Here's the matrix that appears on the right side
Alr= [Adrr          Bdrri  Bdrro         Bdrzc         ;...
      Cdrir         Jriri  Jriro         Jrizc         ;...
      Cdror+CdrozK  Jrori  Jroro+Jnroro  Jrozc+Jnrozc  ;...
      Cdzcr+CdzczK  Jzcri  Jzcro+Jnzcro  Jzczc+Jnzczc  ];

% Make sure that the correct equations are algebraic and
% differential.
Elr = sparse(zeros(size(Alr)));
Elr(1:length(Adrr),1:length(Adrr)) = sparse(eye(size(Adrr)));

% As a final step we get rid of variables that aren't used
% (i.e., that have a row of all zeros defining them). This
% will necessitate modifying Alr,Elr and speedr, since this
% only happens for differential variables.

notUsed = union(find(ismember(Alr,zeros(1,length(Alr)),'rows')),...
    find(ismember(Alr.',zeros(1,length(Alr)),'rows')));
Alr = Alr(setdiff(1:length(Alr),notUsed),setdiff(1:length(Alr),notUsed));
```

```
Elr = Elr(setdiff(1:length(Elr),notUsed),setdiff(1:length(Elr),notUsed));
% We "subtract out" the notUsed variables from speedr, so that these
% indices are still correct.
for index1 = 1:length(notUsed)
    speedr = speedr - (speedr>notUsed(index1));
    notUsed = notUsed - (notUsed>notUsed(index1));
end

eval(['save ',name,'_uncertainty Alr Elr speedr GenSmallSignal -append']);
eval(['save ',name,'_uncertainty BasisSpeedAndAngle Cdzfz K -append']);
eval(['save ',name,'_uncertainty GenSmallSignalFar -append']);
```

```
function [table] = createtable(name,namenew);
% Here we're creating a huge table of all the
% sensitivities in the external reduction.

global table
% Load our already prepared data
eval(['load ',name,'_uncertainty;']);
clear Alr Elr
eval(['load ',namenew,' nrelfar nrelnear;']);

count = 1;
% We store everything in a matrix called 'table'.
% We initialize a huge table now so that we don't run into dynamic
% allocation slowdown.  We'll trim the matrix later.  Actually,
% let's let the user decide how big. The table is of the form
% [oldnode1,oldnode2,sensitivity,newnode1,newnode2], where
% for any old (new) line old (new) nodei is one of the two nodes
% it's connected to. For old loads and new injectors, xnode1 = 0;
disp(' ');
disp('How big of a table do you anticipate?');
disp(' ');
disp('The table will contain the sensitivity of every removed');
disp('element to every created element due to the Ward-reducton');
disp('France-Spain requires 1905 and PG&E requires 683375');
tablesize = input('> ','s');
tablesize = str2num(tablesize);
table = zeros(tablesize,6);

% We should only do this multiplication for the current injectors
% once.  We also need the nominal current injectors.  It's unfortunately
% difficult to do these steps in the "prep" stage since the matrices
% span both reducedsystemprep.m AND externalreductionprep.m
CdzfzK = Cdzfz*K;
NomInj = Jk_zf*iJzfzf*CdzfzK;

% Perturb the small signal load models on zf nodes. We compute the
% derivative based on a change to a zf node diagonal entry. Here we
% assume that the change is a "complex" change, i.e., impedance form,
% but we may change that later.
z
for index1 = 1:z
    % The derivatives for all of the new lines and current injectors.
    % They share terms in common so we do some multiplications first,
    % to avoid duplication.
    del_left = (Jk_zf*iJzfzf(:,[index1*2−1,index1*2]));

    % Using the original matrices with no averaging. However, this
    % being the derivative is based on the assumption that the change
    % can be represented as an "immitance".
    del_Jnew = del_left*(iJzfzf([index1*2−1,index1*2],:)*Jzf_k);
    del_Cnew = del_left*(iJzfzf([index1*2−1,index1*2],:)*CdzfzK);
    % The nominal value of the load at this node.
    old_line = abs(Jzfzf_sum_odd(index1*2−1)+...
        GenSmallSignalFar(index1*2−1,index1*2−1)+...
        Jzfzf_sum_even(index1*2)+...
        GenSmallSignalFar(index1*2,index1*2)+...
        j*Jzfzf_sum_odd(index1*2)+...
        j*GenSmallSignalFar(index1*2−1,index1*2)−...
        j*Jzfzf_sum_even(index1*2−1) − ...
        j*GenSmallSignalFar(index1*2,index1*2−1))/2;
     % We write all of the sensitivities in our table.
    [count] = writetable(count,0,nrelfar(index1),...
        del_Jnew,old_line,countnewlines,inew,jnew,J_new_complex,kept,...
        J_new_sum_odd,J_new_sum_even,del_Cnew,NomInj,BasisSpeedAndAngle,...
        namenew);
         index1
end
count,ff
% Perturb the small signal models on lines between far nodes
% ff is actually sparse, so we need to make it full.
% The structure is almost identical to above.
ff = full(ff);
for index1 = 1:ff
    del_left = (Jk_zf*(iJzfzf(:,[Iff(index1)*2−1,Iff(index1)*2])−...
        iJzfzf(:,[Jff(index1)*2−1,Jff(index1)*2])));
    del_Jnew = del_left*...
        ((iJzfzf([Jff(index1)*2-1,Jff(index1)*2],:)-...
        iJzfzf([Iff(index1)*2-1,Iff(index1)*2],:))*Jzf_k);
    del_Cnew = del_left*...
        ((iJzfzf([Jff(index1)*2-1,Jff(index1)*2],:)-...
        iJzfzf([Iff(index1)*2-1,Iff(index1)*2],:))*CdzfzK);
    % The nominal value of the old line.  The old lines must have
    % the right symmetry since they are certainly impedances.
    old_line = abs(Jzfzf(Iff(index1)*2-1,Jff(index1)*2-1)+...
        j*Jzfzf(Iff(index1)*2-1,Jff(index1)*2));
    % We write all of the sensitivities in our table.
    [count] = writetable(count,nrelfar(Iff(index1)),...
        nrelfar(Jff(index1)),del_Jnew,old_line,countnewlines,inew,...
        jnew,J_new_complex,kept,J_new_sum_odd,J_new_sum_even,...
        del_Cnew,NomInj,BasisSpeedAndAngle,namenew);
    index1
end
count,kf
% Perturb the small signal load models on lines between kept and
% zf nodes.
% kf is actually sparse, so we need to make it full.
% The structure is almost identical to above.
kf = full(kf);
for index1 = 1:kf
    temp = Jk_zf*iJzfzf(:,[Jkf(index1)*2-1,Jkf(index1)*2]);
    temp1 = iJzfzf([Jkf(index1)*2-1,Jkf(index1)*2],:)*Jzf_k;
    temp2 = iJzfzf([Jkf(index1)*2-1,Jkf(index1)*2],:)*CdzfzK;
    del_Jnew = temp*temp1;
    del_Jnew(:,[Ikf(index1)*2-1,Ikf(index1)*2]) = ...
        del_Jnew(:,[Ikf(index1)*2-1,Ikf(index1)*2])+temp;
    del_Jnew([Ikf(index1)*2-1,Ikf(index1)*2],:) = ...
        del_Jnew([Ikf(index1)*2-1,Ikf(index1)*2],:)+temp1;
    del_Jnew([Ikf(index1)*2-1,Ikf(index1)*2],[Ikf(index1)*2-1,...
        Ikf(index1)*2]) = del_Jnew([Ikf(index1)*2-1,Ikf(index1)*2], ...
        [Ikf(index1)*2-1,Ikf(index1)*2]) + speye(2);
    del_Cnew = temp*temp2;
    del_Cnew([Ikf(index1)*2-1,Ikf(index1)*2],:) = ...
        del_Cnew([Ikf(index1)*2-1,Ikf(index1)*2],:) + temp2;
    old_line = abs(Jk_zf(Ikf(index1)*2-1,Jkf(index1)*2-1) +...
        j*Jk_zf(Ikf(index1)*2-1,Jkf(index1)*2));
    % We write all of the sensitivities in our table.
    [count] = writetable(count,kept(Ikf(index1)),...
        nrelfar(Jkf(index1)),del_Jnew,old_line,countnewlines,inew,...
        jnew,J_new_complex,kept,J_new_sum_odd,J_new_sum_even,...
```

```
        del_Cnew,NomInj,BasisSpeedAndAngle,namenew);
end
count
% At this point we can throw out the extra rows in table if it was
% initialized too large.
temp = min(find(table(:,5)==0));
if temp
    table = table(1:temp-1,:);
end

% We'll need to sort the table, before we work with it further.
table = -sortrows(-table,[6,5,4,3]);

% Let's put the table in our file, so we don't have to do it every time.
eval(['save ',name,'_uncertainty table -append']);

% At some point we need to determine a way to read in standard
% deviations for these sensitivities, so we can also weight our
% sensitivities as explained in the thesis.  I do it by hand now.

% On a related note, figuring out the standard deviations for the
% new elements is also tricky.  I haven't implemented it here, but
% I outline how to do it in the thesis.  I do it "ad hoc" at the
% moment on an "as needed" basis.
```

```
function [count] = writetable(count,old1,old2,...
    del_Jnew,old_line,countnewlines,inew,jnew,J_new_complex,kept,...
    J_new_sum_odd, J_new_sum_even,del_Cnew,NomInj,BasisSpeedAndAngle,...
    namenew);
% In this function we'll actually write what we need to the table.

eval(['load ',namenew,' refbasis;']);
global table

% pull-out the change to each of the new lines.
for index2 = 1:countnewlines-1
    % pull out the derivative.
    deriv = abs(del_Jnew(inew(index2)*2-1,jnew(index2)*2-1)+...
        del_Jnew(inew(index2)*2,jnew(index2)*2)+...
        del_Jnew(jnew(index2)*2-1,inew(index2)*2-1)+...
        del_Jnew(jnew(index2)*2,inew(index2)*2)+...
        j*del_Jnew(inew(index2)*2-1,jnew(index2)*2) -...
        j*del_Jnew(inew(index2)*2,jnew(index2)*2-1) +...
        j*del_Jnew(jnew(index2)*2-1,inew(index2)*2) - j*...
        del_Jnew(jnew(index2)*2,inew(index2)*2-1))/4;
    % write the sensitivity to the table.  We haven't incorporated
    % s.d. weighting yet.
    if deriv~=0
        table(count,:) = [old1,old2,...
            deriv*old_line/abs(J_new_complex(inew(index2),jnew(index2))),...
            kept(inew(index2)),kept(jnew(index2)),0];
        count = count + 1;
    end
end

% We can do the same thing for the "impedance" term of the
% current injector at each retained node, which appears on
% the diagonal of del_Jnew.
del_Jnew_sum_odd = sum(del_Jnew(:,1:2:length(del_Jnew)),2);
del_Jnew_sum_even = sum(del_Jnew(:,2:2:length(del_Jnew)),2);
for index2 = 1:length(del_Jnew)/2
    deriv = abs(del_Jnew_sum_odd(index2*2-1)+...
        del_Jnew_sum_even(index2*2)-j*del_Jnew_sum_odd(index2*2) ...
        + j* del_Jnew_sum_even(index2*2-1))/2;
    if deriv ~= 0
        table(count,:) = [old1,old2,...
            deriv*old_line/abs(J_new_sum_odd(index2*2-1) + ...
            J_new_sum_even(index2*2) - j*J_new_sum_odd(index2*2) + ...
            j*J_new_sum_even(index2*2-1))/2,kept(index2),kept(index2),0];
        count = count+1;
    end
end

% We also need to do the same for the current injector "dynamic" parts.
for index1 = 1:length(BasisSpeedAndAngle)/2
    for index2 = 1:length(del_Jnew)/2
        deriv = abs(del_Cnew(index2*2-1,BasisSpeedAndAngle(index1*2-1))+...
            del_Cnew(index2*2,BasisSpeedAndAngle(index1*2)) + ...
            j*del_Cnew(index2*2-1,BasisSpeedAndAngle(index1*2)) - ...
            j*del_Cnew(index2*2,BasisSpeedAndAngle(index1*2-1)))/2;
        if deriv~=0
            table(count,:) = [old1,old2,...
                deriv*old_line/abs(NomInj(index2*2-1,...
                BasisSpeedAndAngle(index1*2-1))+NomInj(index2*2,...
                BasisSpeedAndAngle(index1*2)) + j*NomInj(index2*2-1,...
                BasisSpeedAndAngle(index1*2)) - j*NomInj(index2*2,...
                BasisSpeedAndAngle(index1*2-1)))/2,refbasis(index1),...
            kept(index2),1];
         count = count+ 1;
      end
   end
end
```

```
function [table_new,howmany] = win_for(percentage,table);
% This function will take a table and a percentage and spit
% out a new table that has all sensitivities for each new
% line from the max for that line down to the given
% percentage of the max.

% Find the length of the table and initialize some counters.
len = length(table);
count = 1;
count1 = 1;
index1 = 1;

% Keep on going until count1 is longer than length
% (i.e., the table is finished).
while (count1<=len)
   % The maximum entry for this new line (assuming table is sorted).
   max1 = table(count1,3);
   % Pull out the nodes corresponding to the new line.
   new_line1 = table(count1,4);
   new_line2 = table(count1,5);
   count2 = 0;
   % Go down through the table until you hit something smaller
   % than percentage of max.
   while (table(count1,3)/max1 >= percentage/100)
      table_new(count,:) = table(count1,:);
      count = count+1;
      count1 = count1 + 1;
      count2 = count2 + 1;
   end
   % record how many old elements you wrote into new_table for
   % this new element.
   howmany(index1) = count2;
   index1 = index1 + 1;
   % scroll down through and skip the rest of the old elements
   % for this new element.
   while(table(count1,4) == new_line1 & table(count1,5) == new_line2)
      count1 = count1 + 1;
```

```
    if count1 > len
      break
    end
  end
end
```

```
function [eigentable] = eigensense(name,namenew);

% This function will produce an analysis of which elements
% (loads, lines or current injectors) are most critical to
% the reduced order system being studied. This analysis is
% based on the sensitivities of the (finite)
% eigenvalues of the DAE model described by Alr and Elr.

eval(['load ',name,'_uncertainty Alr Elr speedr GenSmallSignal;']);

%**************************************************************

% We compute the generalized eigenanalysis for our problem. V
% contains the right eigenvectors, W contains the left eigenvectors
% (in both cases as columns) and D contains (on its diagonal) the
% eigenvalues. The eigenvalues for the algebraic equations are at
% infinity, but we aren't primarily concerned with them, so we don't
% bother to generate them. We don't use Matlab's built-in
% generalized eigenanalysis function, because it's slow.
% So, we only have the generalized eigenvectors corresponding to
% finite eigenvalues.

% we need to know which are the differential equations
% numdiff is the number of differential equations.

% diffs and algs denote the differential and algebraic equations
numdiff = length(find(diag(Elr)));
diffs = 1:numdiff;
algs = numdiff+1:length(Alr);

% The corresponding differential system after "Schur decomposition"
A = full(Alr(diffs,diffs)) - full(Alr(diffs,algs)*inv(Alr(algs,algs))*...
   Alr(algs,diffs));

% We do the eigenanalysis
% (This is based on some basic linear algebra. The details are in
% the third EDF report on SME (1999).
```

```
[Vd,D] = eig(A);
Va = -inv(Alr(algs,algs))*Alr(algs,diffs)*Vd;

V = [Vd; Va];

Wd = inv(Vd).';
Wa = -(Wd*Alr(diffs,algs)*inv(Alr(algs,algs))).';

W = [Wd;Wa];

%**************************************************************

% We are mostly concerned with the Alr(algs,:) portion of the system
% since that is where all of the lines, current injectors and loads
% appear in our represenation.

% The first thing we need to do is generate a "connection matrix" for
% this portion of our system. Each "connection" is a 2x2 block
% (in many cases representing an impedance term), and we want to find
% out where the nonzero blocks are so we can find out the
% sensitivity of the eigenvalues to these blocks. We also need to be
% cautious because "lines" actually appear as four sets of blocks in
% the lower right hand portion of the matrix.

J = Alr(algs,algs); % This is the part functioning as an impedance matrix.

% We collapse things so that there's only one entry per bus.
Jpossym = abs(J) + abs(J');
Jhalfr = Jpossym(1:2:length(Jpossym),:) + Jpossym(2:2:length(Jpossym),:);
Jhalf = Jhalfr(:,1:2:length(Jpossym)) + Jhalfr(:,2:2:length(Jpossym));
% We find all of the nonzero terms.
[IJ,KJ,VJ] = find(Jhalf~=0);
connexY = sparse(IJ,KJ,VJ);

% A similar logic holds for the current injector terms. They reside in
% Alr(algs,diffs).
% The injectors only occur on speed/angle differential variables.

CurrInj = Alr(algs,diffs);
CurrInj = CurrInj(:,sort([speedr,speedr+1]));

% We do things similarly as for Y, but CurrInj isn't square so we don't
% impose symmetry.
CurrInjpos = abs(CurrInj);
```

```
CurrInjhalfr = CurrInjpos(1:2:size(CurrInjpos,1),:) + ...
    CurrInjpos(2:2:size(CurrInjpos,1),:);
CurrInjhalf = CurrInjhalfr(:,1:2:size(CurrInjpos,2)) + ...
    CurrInjhalfr(:,2:2:size(CurrInjpos,2));
% We find the nonzero terms.
[IC,KC,VC] = find(CurrInjhalf~=0);
connexCurrInj = sparse(IC,KC,VC);

% One last thing we need to do for preparation is find the
% magnitudes of the terms on the diagonal of Y while subtracting
% all of the other terms from it's row.  These will be part of
% the small signal admittance in the current injectors at those
% points; another portion (the part that was there before the reduction)
% corresponds to loads that don't get reduced out.  We also subtract
% out the generator small signal admittance terms, since we won't be
% changing generators.

Y = J - GenSmallSignal;
Ydiag1 = sum(Y(:,1:2:length(Y)),2);
Ydiag2 = sum(Y(:,2:2:length(Y)),2);
Ydiag = [Ydiag1,Ydiag2];

%****************************************************************

% Now, we need to compute the sensitivities.

% We'll first do the sensitivities to changes in line parameters.

% The first task is to find these lines.
% We find the location of all of the lines and count how many there are.
connexY = connexY - diag(diag(connexY));
connexY_tri = triu(connexY);
numLines = sum(sum(connexY))/2;
[ILines,JLines] = find(connexY_tri);
numLines = full(numLines);

% Now, we do the sensitivity calculation.  We'll assume that the change
% to the line is reactive.  Our computation is w.'*A*v for each eigenvalue.

LineDel = [0,-1;1,0]/sqrt(2);
for index1 = 1:numdiff
    for index2 = 1:numLines
        sense(index1,index2) = ...
            W([numdiff+2*ILines(index2)-1,numdiff+2*ILines(index2)],...
            index1).'*LineDel*(V([numdiff+2*ILines(index2)-1,numdiff+. . .
            2*ILines(index2)],index1)−V([numdiff+2*JLines(index2)−1,. . .
            numdiff+2*JLines(index2)],index1))+W([numdiff+2*. . .
            JLines(index2)−1,numdiff+2*JLines(index2)],index1).'. . .
            *LineDel*(V([numdiff+2*JLines(index2)−1,numdiff+. . .
            2*JLines(index2)],index1)−V([numdiff+2*ILines(index2)−1,. . .
            numdiff+2*ILines(index2)],index1));
    end
end

% We need to turn this derivate into a sensitivity. We are only
% interested in real shifts to the eigenvalues since these are most
% important for stability.

% First we divide each entry by it's eigenvalue.
Dvec = diag(real(D));
Dvecinv = diag(1./Dvec);
sense = Dvecinv*real(sense);

% Now, let's find the reactive part of each line we perturbed.
for i = 1:numLines
    Jreac(i) = abs(J(ILines(i)*2−1,JLines(i)*2) − . . .
        J(ILines(i)*2,JLines(i)*2−1) − J(JLines(i)*2,ILines(i)*2−1) . . .
        + J(JLines(i)*2−1,ILines(i)*2))/4;
end

% Then, we multiply our matrix by these reactive magnitudes.
sense = sense*diag(Jreac);
%*****************************************************************

% Now, we can do the sensitivity to changes in "load parameters".
% We'll consider those to be changes to the diagonal of J, although
% some of these are folded in the current injectors. We can untangle
% that later.

% Here we'll just use an arbitrary impedance change. Let's say it's real.
LoadDel = [1,0;0,1]/sqrt(2);
for index1 = 1:numdiff
    for index2 = 1:length(Y)/2
        senseLoad(index1,index2) = . . .
            W([numdiff+2*index2−1,numdiff+2*index2],index1).'. . .
            *LoadDel*(V([numdiff+2*index2−1,numdiff+2*index2],index1));
    end
end
```

```
% We need to turn this derivate into a sensitivity.  We are only
% interested in real shifts to the eigenvalues since these are most
% important for stability.

% First we divide each entry by its eigenvalue.
senseLoad = Dvecinv*real(senseLoad);

% Now, let's find the real part of each load we perturbed.
for i = 1:length(Y)/2
   YdiagR(i) = (Ydiag(2*i-1,1) + Ydiag(2*i,1))/2;
end

% Then, we multiply our matrix by these resistive magnitudes.
senseLoad = senseLoad*diag(YdiagR);

%**********************************************************

% We also need to do the sensitivity to the current injector
% parameters.  They appear as described above.  However, some of
% these terms aren't part of the current injectors.  The easy way
% to tell whether it's a current injector or not (that is, current
% injector from SME) is to see which differential variables and
% algebraic variables are related.

% Let's do reactive changes, though again, this is pretty arbitrary.
InjDel = [0,1;-1,0]/sqrt(2);
for index1 = 1:numdiff
   for index2 = 1:length(KC)
      senseInj(index1,index2) = ...
         W([numdiff+2*IC(index2)-1,numdiff+2*IC(index2)],index1).' ...
         *LoadDel*(V([2*KC(index2)-1,2*KC(index2)],index1));
   end
end

% We need to turn this derivate into a sensitivity.  We are only
% interested in real shifts to the eigenvalues since these are most
% important for stability.

% First we divide each entry by its eigenvalue.
senseInj = Dvecinv*real(senseInj);

% Now, let's find the reactive part of each injector we perturbed.
for i = 1:length(KC)
```

```
    CurrInjR(i) = (CurrInj(2*IC(i)-1,2*KC(i)) -...
        CurrInj(2*IC(i),2*KC(i)-1))/2;
end

% Then, we multiply our matrix by these resistive magnitudes.
senseInj = senseInj*diag(CurrInjR);

%**************************************************************

% The last thing to do is to sort and sift this information, somehow.

% The first thing to do is to take an average of the changes to the
% eigenvalues for each component changed.

sense = (sum(abs(sense),1)/length(D)).';
senseLoad = (sum(abs(senseLoad),1)/length(D)).';
senseInj = (sum(abs(senseInj),1)/length(D)).';

% We need some vectors so we can represent our different loads and
% lines in terms of SME bus and generator indices.

eval(['load ',namenew,' keptmach relin relout nrelnear;']);
linetranslate = [relin, relout, nrelnear].';

% So, we'll write things in terms of old stuff1, old stuff2, sense.

%First, we'll do the lines, then the loads, then the CurrInj
eigentable = [linetranslate(ILines),linetranslate(JLines),sense,...
    zeros(length(sense),1);...
    linetranslate, zeros(length(linetranslate),1), senseLoad,...
    zeros(length(senseLoad),1);...
    linetranslate(IC),keptmach(KC).',senseInj,...
    ones(length(senseInj),1)];
eigentable = -sortrows(-eigentable,[3,4,1,2]);

% As for the external reduction, we have not yet automated the
% standard deviation weighting described in the thesis.
```

Bibliography

[1] IEEE Task Force on Load Representation for Dynamic Performance, "Load representation for dynamic performance analysis," *IEEE Transactions on Power Systems*, vol. 8, pp. 472–482, May 1993.

[2] IEEE Task Force on Load Representation for Dynamic Performance, "Bibliography of load models for power flow and dynamic performance simulation," *IEEE Transactions on Power Systems*, vol. 10, pp. 523–527, February 1995.

[3] IEEE Task Force on Load Representation for Dynamic Performance, "Standard load models for power flow and dynamic performance simulation," *IEEE Transactions on Power Systems*, vol. 10, pp. 1302–1313, August 1995.

[4] D. J. Hill, "Nonlinear dynamic load models with recovery for voltage stability studies," *IEEE Transactions on Power Systems*, vol. 8, pp. 166–172, February 1993.

[5] M. K. Pal, "Voltage stability: analysis needs, modelling requirement, and modelling adequacy," *IEE Proceedings-C*, vol. 140, pp. 279–286, July 1993.

[6] J. R. Hockenberry, "Power system dynamic load modeling," Master's thesis, Massachusetts Institute of Technology, February 1997.

[7] S. R. Shaw, C. B. Abler, R. F. Lepard, D. Luo, S. B. Leeb, and L. K. Norford, "Instrumentation for high performance nonintrusive electrical load monitoring," *ASME Journal of Solar Energy Engineering*, vol. 120, pp. 224–229, August 1998.

[8] D. N. Kosterev, C. W. Taylor, and W. A. Mittelstadt, "Model validation for the August 10, 1996 WSCC system outage," *IEEE Transactions on Power Systems*, vol. 14, pp. 967–979, August 1999.

[9] K. Walve, "Modelling of power system components at severe disturbances," Tech. Rep. 38-18, CIGRÉ, 1986.

[10] G. N. Ramaswamy, L. Rouco, O. Fillâtre, G. C. Verghese, P. Panciatici, B. C. Lesieutre, and D. Peltier, "Synchronic Modal Equivalencing (SME) for structure-preserving dynamic equivalents," *IEEE Transactions on Power Systems*, vol. 11, pp. 19–29, February 1996.

[11] G. N. Ramaswamy, C. Evrard, G. C. Verghese, O. Fillâtre, and B. C. Lesieutre, "Extensions, simplifications, and tests of Synchronic Modal Equivalencing (SME)," *IEEE Transactions on Power Systems*, vol. 12, pp. 896–905, May 1997.

[12] G. N. Ramaswamy, *Modal Structures and Model Reduction, with Application to Power System Equivalencing.* PhD thesis, Massachusetts Institute of Technology, June 1995.

[13] J. M. Hagee, "Explorations and extensions of Synchronic Modal Equivalencing (SME)," Master's thesis, Massachusetts Institute of Technology, February 1997.

[14] J. M. Hagee, B. Ayazifar, G. C. Verghese, and B. C. Lesieutre, "Load bus partitioning for Synchronic Modal Equivalencing (SME)," in *28th North American Power Symposium*, pp. 57–61, November 1996.

[15] M. A. Tatang, *Direct Incorporation of Uncertainty in Chemical and Environmental Systems.* PhD thesis, Massachusetts Institute of Technology, 1995.

[16] M. Webster, M. A. Tatang, and G. J. McRae, "Application of the probabilistic collocation method for an uncertainty analysis of a simple ocean model," Tech. Rep. 4, Joint Program on the Science and Policy of Global Change, MIT, Cambridge, MA, January 1996.

[17] B. C. Lesieutre and J. R. Hockenberry, "Uncertainty analysis of power system simulations and ATC calculations using the probabilistic collocation method," in *Bulk Power Systems Dynamics and Control IV - Restructuring*, (Santorini, Greece), August 1998.

[18] R. N. Allan and R. Billinton, "Probabilistic methods applied to electric power systems - are they worth it?," *Power Engineering Journal*, vol. 6, pp. 121–129, May 1992.

[19] M. T. Schilling, A. M. L. da Silva, R. Billinton, and M. A. El-Kady, "Bibliography on power system probabilistic analysis (1962-1988)," *IEEE Transactions on Power Systems*, vol. 5, pp. 1–11, February 1990.

[20] R. Billinton, "Bibliography on the application of probability methods in power system reliability evaluation," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-91, pp. 649–660, Mar/Apr 1972.

[21] IEEE Subcommittee on the Application of Probability Methods, "Bibliography on the application of probability methods in power system reliability evaluation 1971-1977," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-97, pp. 2235–2242, 1978.

[22] R. N. Allan, R. Billinton, and S. H. Lee, "Bibliography on the application of probability methods in power system reliability evaluation 1977-1982," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-103, pp. 275–282, February 1984.

[23] R. N. Allan, R. Billinton, S. M. Shahidehpour, and C. Singh, "Bibliography on the application of probability methods in power system reliability evaluation 1982-1987," *IEEE Transactions on Power Systems*, vol. 3, pp. 1555–1564, November 1988.

[24] R. N. Allan, R. Billinton, A. M. Breipohl, and C. H. Grigg, "Bibliography on the application of probabilitiy methods in power system reliability evaluation 1987-1991," *IEEE Transactions on Power Systems*, vol. 9, pp. 41–49, February 1994.

[25] R. N. Allan, R. Billinton, A. M. Breipohl, and C. H. Grigg, "Bibliography on the application of probability methods in power system reliability evaluation 1992-1996," *IEEE Transactions on Power Systems*, vol. 14, pp. 51–57, February 1999.

[26] I. A. Hiskens and M. A. Pai, "Sensitivity analysis of power system trajectories: Recent results," in *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 439–443, 1998.

[27] I. A. Hiskens and M. A. Pai, "Trajectory sensitivity analysis of hybrid systems." To appear in *IEEE Transactions on Circuits and Systems.*

[28] I. A. Hiskens and M. Akke, "Analysis of the Nordel power grid disturbance of January 1, 1997 using trajectory sensitivites," *IEEE Transactions on Power Systems*, vol. 14, pp. 987–994, August 1999.

[29] The MathWorks, Inc., *Using MATLAB Version 5.* Natick, MA, 1998.

[30] *Maple on Athena (AC-72)*, second ed., August 1998. http://web.mit.edu/olh/Maple/Maple.html.

[31] R. Billinton, *Reliability Assessment of Electric Power Systems Using Monte Carlo Methods.* New York: Plenum Press, 1984.

[32] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration.* New York, NY: Academic Press, Inc., 1975.

[33] J. R. Munkres, *Analysis on Manifolds.* Reading, Massachusetts: Addison-Wesley Publishing Company, 1991.

[34] A. W. Drake, *Fundamentals of Applied Probability Theory.* New York: McGraw-Hill, Inc., 1967.

[35] P. Kundur, *Power System Stability and Control.* McGraw-Hill, 1994.

[36] J. H. Chow, R. A. Date, H. Othman, and W. W. Price, "Slow coherency aggregation of large power systems," in *Eigenanalysis and Frequency Domain Methods for System Dynamic Performance*, IEEE 90TH0292-3-PWR, pp. 50–60, 1989.

[37] J. H. Chow, *Time-Scale modeling of dynamic networks with applications to power systems.* New York: Springer-Verlag, 1982.

[38] J. Y. Astic, A. Bihain, and M. Jerosolimski, "The mixed Adams-BDF variable step size algorithm to simulate transient and long term phenomena in power systems," *IEEE Transactions on Power Systems*, vol. 9, pp. 929–935, May 1994.

[39] J. F. Vernotte, P. Panciatici, B. Meyer, J. P. Antoine, J. Deuse, and M. Stubbe, "High fidelity simulation of power system dynamics," *IEEE Computer Applications in Power*, vol. 8, pp. 37–41, January 1995.

[40] J. B. Ward, "Equivalent circuits for power flow studies," *AIEE Transactions on Power Apparatus and Systems*, vol. 68, pp. 373–382, 1949.

[41] F. R. Gantmacher, *The Theory of Matrices*, vol. two. New York, NY: Chelsea Publishing Company, 1959.

[42] S. B. Yusof, G. J. Rogers, and R. T. H. Alden, "Slow coherency based network partitioning including load buses," *IEEE Transactions on Power Systems*, vol. 8, pp. 1375–1382, August 1993.

[43] I. J. Pérez-Arriaga, G. C. Verghese, and F. C. Schweppe, "Selective modal analysis with applications to electric power systems, part I: Heuristic introduction," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-101, pp. 3117–3125, September 1982.

[44] G. C. Verghese, I. J. Pérez-Arriaga, and F. C. Schweppe, "Selective modal analysis with applications to electric power systems, part II: The dynamic stability problem," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-101, pp. 3126–3134, September 1982.

[45] Y. V. Makarov, V. A. Maslennikov, and D. J. Hill, "Revealing loads having the biggest influence on power system small disturbance stability," *IEEE Transactions on Power Systems*, vol. 11, pp. 2018–2023, November 1996.

[46] P. M. Frank, *Introduction to System Sensitivity Theory.* New York, NY: Academic Press, 1978.

[47] D. Bertsekas and R. Gallager, *Data Networks.* Upper Saddle River, NJ: Prentice Hall, second ed., 1992.

[48] N. Narasimhamurthi and M. Ilić-Spong, "A proof of localized steady-state response in large interconnected power system," in *Proc. Allerton Conf.*, pp. 997–1002, 1980.

[49] M. Ilic-Spong, M. W. Spong, and R. Fischl, "The no-gain theorem and localized response for the decoupled P-θ power network with active power losses included," *IEEE Transactions on Circuits and Systems*, vol. CAS-32, pp. 170–177, February 1985.

[50] M. Ilić-Spong, J. S. Thorp, and M. W. Spong, "Localized response performance of the decoupled Q-V network," *IEEE Transactions on Circuits and Systems*, vol. CAS-33, pp. 316–322, March 1986.

[51] F. Zhang, *Matrix Theory: Basic Results and Techniques.* New York, NY: Springer-Verlag, 1999.

[52] J. V. Milanović and I. A. Hiskens, "Effects of load dynamics in power system damping," *IEEE Transactions on Power Systems*, vol. 10, pp. 1022–1028, May 1995.

[53] I. A. Hiskens and J. V. Milanović, "Load modelling in studies of power system damping," *IEEE Transactions on Power Systems*, vol. 10, pp. 1781–1788, November 1995.

[54] I. A. Hiskens and J. V. Milanović, "Locating dynamic loads which significantly influence damping," *IEEE Transactions on Power Systems*, vol. 12, pp. 255–261, February 1997.

[55] J. V. Milanović, I. A. Hiskens, and V. A. Maslennikov, "Ranking loads in power systems - comparison of different approaches," *IEEE Transactions on Power Systems*, vol. 14, pp. 614–619, May 1999.

[56] E. W. Kimbark, *Power System Stability: Elements of Stability Calculations*, vol. 1. NY, NY: IEEE Press, 1995.

[57] M. C. Tavares, J. Pissolato, and C. M. Portela, "Guidelines for transmission line modeling," in *IEEE Power Engineering Society Summer Meeting*, vol. 2, (Edmonton, Alberta, Canada), pp. 786–791, IEEE, 1999.

[58] W. K. Nicholson, *Elementary Linear Algebra with Applications.* Boston: PWS-KENT Publishing Company, second ed., 1990.

[59] I. J. Pérez-Arriaga, G. C. Verghese, F. L. Pagola, J. L. Sancha, and F. C. Schweppe, "Developments in selective modal analysis of small-signal stability in electric power systems," *Automatica*, vol. 26, no. 2, pp. 215–231, 1990.

[60] I. J. Pérez-Arriaga, *Selective Modal Analysis with Applications to Electric Power Systems.* PhD thesis, Massachusetts Institute of Technology, June 1981.

[61] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Baltimore, Maryland: The Johns Hopkins University Press, 1983.

[62] G. W. Stewart, "Perturbation theory for the generalized eigenvalue problem," in *Recent Advances in Numerical Analysis* (C. de Boor and G. H. Golub, eds.), (New York, NY), pp. 193–206, Academic Press, 1978.