Explorations and Extensions of
Synchronic Modal Equivalencing (SME)

by

Jesko M. Hagee
B.S., United States Naval Academy (1995)

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

Master of Science in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1997

© Jesko M. Hagee, MCMXCVII. All rights reserved.

The author hereby grants to MIT permission to reproduce and distribute publicly paper and electronic
copies of this thesis document in whole or in part, and to grant others the right to do so.
Explorations and Extensions of Synchronic Modal Equivalencing (SME)
by Jesko M. Hagee

Submitted to the Department of Electrical Engineering and Computer Science on January 17, 1997, in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering

Abstract

In this thesis we further develop the theory of synchrony and Synchronic Modal Equivalencing (SME), as applied to models of power system dynamics.

First, we extend the SME partitioning algorithm to include the grouping of load buses. The generalized eigenanalysis setup previously used for load bus grouping in slow coherency is adapted to the synchrony framework. The extended SME partitioning algorithm is then applied to two different power system models.

The second part of this thesis deals with synchrony theory. We examine the sensitivity of a swing model and its eigenstructure to order \( \epsilon \) perturbations on a cutset of lines. Both symbolically and numerically, we establish that the changes in the eigenvalues and eigenvectors are of the same order as the changes in the line parameters. We apply our sensitivity derivations to a simple power system model.

Lastly, we examine the emergence of ideal synchrony from an idealized swing model. By varying the mutual couplings in a nine-generator model, we watch the evolution of the eigenvectors and see synchrony in both the slow and fast modes of the system. We also compare our partitioning results to those achieved using slow coherency and demonstrate why synchrony has the potential for better results.

Thesis Supervisor: George C. Verghese
Title: Professor of Electrical Engineering
Dedication

To

my FAMILY

my Parents,
My Sister,
and My Wife.
Acknowledgements

First and foremost, I must thank profusely my advisor, George Verghese. Always willing to spend a few minutes with me, George’s guidance was indispensable and his patience with my writing was indefatigable. A great deal of what I have learned during the entire research and writing process I owe to George.

Two others were essential in their contributions to discussions regarding research: Bernie Lesieutre and Babak Ayazifar. Both were always available whenever I had problems or questions regarding anything. Bernie helped me to understand any power systems issues that arose and Babak was available for vigorous debates on everything from power systems to politics.

James Hockenberry was also a great help. Aside from the many casual discussions we had, he was also always willing to read and comment on the multiple drafts of the thesis. His insight into the English language made me occasionally think he should be in the humanities rather than engineering.

Christophe Evrard of Electricité de France (EDF) was instrumental in getting me started on this work over a year ago. Since then, Christophe has served as my contact with EDF.

For my sanity, I have to thank everyone else in LEES and 10-082. Vivian always knew how to get things done within the department, even when I had no clue what it was I needed to accomplish. Mark trudged through three classes with me and then brought in Duke Nuke'em to keep me from getting work done. Everyone else, well, there are too many to name, but at the risk of missing someone: Vahe, Jama, Ike, Ankur, Brian, Khurram, Julio, and Kwabena.

This thesis is dedicated to my family; my parents, Mike and Silke, my sister, Stephanie, and my wife, Becky. Without their constant support, I could never have achieved all that I have. To them I owe more than I can ever express.
### Contents

1 Introduction .......................... 15
   1.1 Previous Work .......................... 16
   1.2 Contributions and Organization of this Thesis ...................... 16

2 Background Theory .......................... 19
   2.1 Swing Equation Model .......................... 19
   2.2 Coherency .......................... 20
      2.2.1 Slow Coherency .......................... 20
      2.2.2 Generator Partitioning and Dynamic Equivalencing in Slow Coherency ........ 21
   2.3 Synchrony .......................... 21
      2.3.1 Synchronic Modal Equivalencing .......................... 22
      2.3.2 Generator Partitioning in SME .......................... 22

3 Load Bus Partitioning in SME .......................... 25
   3.1 Load Bus Partitioning in Slow Coherency .......................... 25
   3.2 Extending SME Partitioning .......................... 25
      3.2.1 Structure-Preserving Swing Model .......................... 26
      3.2.2 Generalized Eigenanalysis .......................... 28
      3.2.3 Load Bus Partitioning .......................... 29
   3.3 Test Results .......................... 29
      3.3.1 New England Model .......................... 30
      3.3.2 France-Spain Model .......................... 31
   3.4 Degree of Synchrony in Boundary Buses .......................... 34
Contents

4 Modal Sensitivity .............................................. 37
   4.1 Eigenvector Sensitivity .................................. 38
   4.2 Test Results .................................................. 39
       4.2.1 Three Generators .................................. 39
       4.2.2 Two Generators, One Load ............................ 42

5 Modal Patterns .................................................. 45
   5.1 System Setup .................................................. 45
   5.2 The Decoupled System ....................................... 45
   5.3 The Coupled System ......................................... 47

6 Concluding Remarks ............................................. 51
   6.1 Summary ....................................................... 51
   6.2 Future Work ................................................ 52
       6.2.1 Simulations ......................................... 52
       6.2.2 Weighting Factors ................................... 52

A Basis Selection .................................................. 53
   A.1 Background ................................................ 53
   A.2 The Norm Idea .............................................. 54
       A.2.1 Full Combinatorial Search ............................. 54
       A.2.2 QR decomposition ...................................... 54
       A.2.3 Different Norms ...................................... 56
   A.3 Results ....................................................... 56
       A.3.1 Basis results ........................................... 56
   A.4 Matlab Coding ................................................ 58

B Sensitivity Derivations .......................................... 59
B.1 Eigenvalue Sensitivity ........................................ 59
B.2 Eigenvector Sensitivity .......................................... 60

C Matlab Load/Gen Partitioning Program .......................... 63
List of Figures

2.1 Before and After SME-Based Equivalencing [14] On the left is a schematic of a simple power system. On the right we show the same system after equivalencing. 23

3.1 Line Diagram of the New England Model This diagram shows the generator and load bus connections. The load buses are numbered 1 to 29; the generator buses, 30 to 39. The indicated partitioning corresponds to Case I below. 30

3.2 Relative Geographical Locations of the Generator Buses in the Test Model. The generators are numbered 1 to 23. 32

3.3 Schematic Representation of Areas and Their Interconnections. This schematic map shows the number of lines interconnecting the different areas; the letters in the circles correspond to those used in the text to denote the various areas. 35

4.1 Three-Generator Model This figure shows a model of a simple three-generator power system. We assume lossless lines; $b_{ij}$ are the susceptances between buses $i$ and $j$. 39

4.2 Two Generator/One Load Model This figure shows a model of a simple two-generator, one-load power system. We assume lossless lines; $b_{ij}$ are the susceptances between buses $i$ and $j$. 42

5.1 System Model The model, comprising nine generators, may be thought of equivalently as (a) three ‘tiers’ of generators, or (b) three concentric circles of generators in a plane. 46
List of Tables

3.1 New England Model: Eigenvalues ................................ 30

3.2 France-Spain Model: Eigenvalues ............................... 33

3.3 New England Model: Degrees of Synchrony with Different Areas. The entries show, for certain selected buses, the degrees of synchrony with the basis machine for the indicated area. The column heading indicates the case (I,II) and the area. The row label indicates the bus number. ............................... 35

3.4 France-Spain Model: Degrees of Synchrony with Different Areas. The entries show, for certain selected buses, the degrees of synchrony with the basis machine for the indicated area. The column heading indicates the case (I,II,III) and the area. The row label indicates the bus number. ............................... 36

A.1 Basis selection using different norms. Norm indicates the norm used in doing a combinatorial search (1-norm, 2-norm, ∞-norm, Frobenius norm). Pure QR and Ramas denote the use of the QR decomposition without further searching and the algorithm in [14], respectively. Basis indicates the row vectors chosen for $V_b$. ............................... 57

A.2 Norms of different basis selections. This table shows the norms of the different bases found using three different methods: Ramaswamy, pure QR, and combinatorial. ............................... 58
Chapter 1

Introduction

The size and complexity of interconnected power systems are ever increasing. Dynamic equivalent-
ing is often used to reduce the order of the system model. It is difficult to take a large system
model and reduce it to one that can be simulated in real-time but still maintains accurately the
important dynamics of the full system. This is an important challenge in the area of power systems.
If an entire group of generator and load buses could be well represented by an aggregate model,
then we could potentially reduce the order of the system dramatically and still maintain accuracy
in our model. Large power networks are often partitioned into areas to make this reduction or
equivalencing easier.

In practice, the system-wide dynamics of a large power system can be well represented by
a linearized undamped swing-equation model \[8\]. The modes of this model can be easily found
and examined. The model used in the slow coherency theory \[3\] comprises areas that have only
weak incremented couplings to each other (corresponding to heavily loaded tie-lines), but significant
couplings among the generators within each area. In this setting, the dynamics between machines
of different areas, termed inter-area dynamics, constitute slower modes than the dynamics between
machines in the same area, or intra-area dynamics. Thus, the inter-area dynamics correspond to
the slow modes of the system. Slow coherency demonstrates that, if only the slow inter-area modes
are excited, then the generators in each area will move in an identical fashion, or coherently.

For the past decade and a half, slow coherency has been a widely accepted method for
grouping buses. In partitioning, slow coherency looks for buses that move identically or coherently
in the slow modes, and groups them together. Once the buses are partitioned into areas, a study
area can be chosen. The study area is made up of one or more slow-coherent areas. The buses
external to the study area are termed less-relevant and will be the ones represented by aggregate
models. We maintain all the details of the study area model. In this way, we hope to preserve both
the slower inter-group dynamics and the faster dynamics of the intra-group modes in the study
area.

In this thesis, we will explore a new method for performing the model reduction, termed
Synchronic Modal Equivalencing (SME), which was presented in \[14\]. SME is based on the notion
Introduction

of *synchrony*, a more general concept than slow coherency. There are two major differences between slow coherency and synchrony. First, synchrony (and SME) requires only that the motion of each bus in an area be a *linear combination* of the motion of a set of *basis generators* in the area. Second, we look for synchronic motion to exist in *some subset* of modes, not necessarily the slowest. This subset of modes is termed a *chord*. This is more general than slow coherency because we do not require *identical* motion of buses when only the *slowest* modes are excited. The SME partitioning method is tied very closely to the eigenstructure of the system. Buses are grouped according to the *degree of synchrony* [14] they have with a chosen set of *basis machines*.

Though previous studies in synchrony have yielded promising results, there is still more work to be done in this area. In this thesis, we will concentrate on basic synchrony theory and the partitioning algorithm of SME.

### 1.1 Previous Work

The development of SME by Ramaswamy *et al.* has occurred over the past four years. Details of this development can be found in [14, 15, 16, 17]. In [14, 15, 17], SME was applied to a 23-generator, 60-load model of the France-Spain power grid. SME has also been applied to a 10-generator, 38-load model of the New England power grid [1, 14, 16]. Castrillón Candas [1] also explored alternative basis selection algorithms for synchrony partitioning methods. In addition, as preliminary work for this thesis, a norm-based basis selection algorithm was developed as presented in Appendix A.

All previous work, however, only applied SME theory to the partitioning of *generators* in a network. Although several different methods to partition the load buses have been introduced within the slow-coherency framework, Yusof *et al.* [20] and DeMarco and Wassner [6], very little work on load bus partitioning has been done within the framework of synchrony. The SME method proposed in [14] collapses the model in order to find regular eigenvectors. When this occurs, the load bus variables are lost; grouping of the load buses into areas is not possible.

### 1.2 Contributions and Organization of this Thesis

In this thesis, we will explore several areas of synchrony in greater detail, as well as extend Synchronic Modal Equivalencing theory to include load buses.

In Chapter 2, we will lay the groundwork for the thesis. This will include a review of the background theory and the motivation for slow coherency. We will then proceed, using the same
1.2 Contributions and Organization of this Thesis

setup as the slow coherency picture, to give the background for synchrony and SME theory.

Chapter 3 will focus on the extension of the SME partitioning algorithm to include load buses. From the generalized slow-coherency framework seen in Yusof et al. [20] and DeMarco and Wassner [6], we will extend synchronic partitioning using generalized eigenanalysis. Most of the results presented in this chapter have been previously published in [10]. We apply the extended SME partitioning algorithm to both the New England model seen in [1, 14] and the France-Spain model [14, 15, 16, 17].

Since slow coherency is motivated by the weakly-coupled, multi-area swing model, in Chapter 4 we will study the sensitivity of the model and its eigenstructure to small perturbations in the line parameters. We will see on what order the changes of the eigenvectors are and demonstrate that, for small $\epsilon$ perturbations in the lines, the structure of the eigenvectors is preserved in both generator and load bus variables.

In Chapter 5, by constructing a simple, idealized power system model, we will look at the modal patterns which exist in power systems. By incrementally coupling small, decoupled power systems, we will see how the eigenvectors evolve and the modal patterns change.

Chapter 6 presents our conclusions and suggest ideas for further work in this area.

In Appendix A, we present the norm-based basis selection algorithm mentioned in Section 1.1. This was developed as preliminary research for the thesis. Appendix B shows the derivations for eigenvalue and eigenvector sensitivities. These are drawn from [9]. Appendix C lists the Matlab code written to perform the synchrony-based grouping of generators and loads.
2.1 Swing Equation Model

In practice, the generator dynamics of a large power system can be well represented by a linearized swing-equation model [8]. First we consider the simplest description of the generator dynamics:

\[
J \ddot{\delta}_G + D \dot{\delta}_G = -K_G \Delta \delta_G ,
\]

where \( J \) is the diagonal matrix of generator inertias, \( D \) is the diagonal matrix of damping factors, \( \delta_G \) is the vector of generator bus angles, and \( K_G \) represents the matrix of synchronizing coefficients. The matrix \( K_G \) is symmetric, positive semidefinite and determined solely by network admittance parameters and the load flow solution. Further details on the origin of this model are deferred to Chapter 3. In the undamped case, \( D \) is zero, and the linearized, undamped swing equation takes the form

\[
J \Delta \ddot{\delta}_G = -K_G \Delta \delta_G ,
\]

which can be rewritten as

\[
\Delta \ddot{\delta}_G = -J^{-1} K_G \Delta \delta_G = A \Delta \delta_G
\]

Since \( K_G \) is positive semidefinite and \( J \) is a diagonal matrix with positive, non-zero entries, \( A \) is negative semidefinite. If we suppose that our system has \( n_G \) generators, the eigenvalues of \( A \) are

\[
-\lambda_{n_G-1} \leq \cdots \leq -\lambda_1 \leq -\lambda_0
\]

with \( \lambda_0 = 0 \). The natural frequencies of the swing equation (2.3) are the square roots of \( A \)'s
Background Theory

eigenvalues:

\[ \pm j \sqrt{\lambda_{n_G-1}}, \ldots, \pm j \sqrt{\lambda_1}, 0, 0 \ . \]  

(2.5)

We see that the set of nonzero modes comprises \( (n_G - 1) \) oscillatory motions.

2.2 Coherency

Coherency [3, 4, 5] is based on the notion that generators which are closely coupled will move coherently, i.e., the generator angles will have equal or coherent perturbations.

2.2.1 Slow Coherency

In Equation (2.3), \( K_G \) has one eigenvalue at 0, implying that \( A \) is also singular with one eigenvalue at 0. The corresponding mode shape, or eigenvector, is a vector of all ones: \( u = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^T \). This occurs because the system is undisturbed if all the angular perturbations are equal, or coherent.

Now consider a fully decoupled \( L \)-area power system. Each area may be modeled using swing equations, so \( \delta_k = A_k \delta_k \) with \( k = 1, \ldots, L \). Each \( A_k \) will have similar structure to \( A \) in (2.3); they will each have one eigenvalue at 0. If we represent the entire system in matrix form, and now admit the existence of small, order-\( \epsilon \) coupling among the areas, we obtain

\[
\ddot{\delta} = \left( \begin{array}{c}
\ddot{\delta}_1 \\
\ddot{\delta}_2 \\
\vdots \\
\ddot{\delta}_L
\end{array} \right) = \left( \begin{array}{cccc}
A_1 & \epsilon* & \cdots & \epsilon* \\
\epsilon* & A_2 & \cdots & \epsilon* \\
\vdots & \vdots & \ddots & \vdots \\
\epsilon* & \epsilon* & \cdots & A_L
\end{array} \right) \left( \begin{array}{c}
\delta_1 \\
\delta_2(t) \\
\vdots \\
\delta_L(t)
\end{array} \right) = A(\epsilon) \delta ,
\]  

(2.6)

where the \( \epsilon* \)'s represents different matrices with entries of order \( \epsilon \); these are the coupling terms. When \( \epsilon = 0 \), corresponding to \( A(0) \), the blocks are fully decoupled. In the decoupled case, \( A(0) \) is block diagonal with \( L \) eigenvalues at 0, one zero eigenvalue for each block. For each of the associated eigenvectors, the generator angles within each area are coherent.

In the weakly coupled case, we have small \( \epsilon \), corresponding to weak links interconnecting the areas. Although \( A(\epsilon) \) retains one eigenvalue at 0, the remaining \( (L - 1) \) zero eigenvalues of \( A(0) \) become real, negative eigenvalues of order \( \epsilon \); these are the slow modes of the system. For each of the associated \( (L - 1) \) mode shapes, the generator angles within each area remain approximately coherent, i.e., the rows of these \( (L - 1) \) eigenvectors corresponding to generators in an area will
be nearly equal. In this case, the slow dynamics correspond to interactions among areas, with essentially no intra-area variations. These interactions are termed the inter-area dynamics.

The remaining \((n_G - L)\) eigenvalues of \(A(\varepsilon)\) will be close to the nonzero eigenvalues of \(A(0)\). These faster modes tend to be local to one specific area and correspond to the dynamics of machines within that area. These localized interactions are termed the intra-area dynamics. If only the slow modes of the system are excited, only the inter-area dynamics will be present. When this is true, generators within a group will move coherently. This is called slow coherency.

2.2.2 Generator Partitioning and Dynamic Equivalencing in Slow Coherency

In order to perform dynamic equivalencing, we would like to partition the generators of the system into areas. By looking at the slow modes of the system, we may be able to identify slow-coherent machines. These generators can then be grouped into slow-coherent areas. One of these areas is picked as the study area. The study area is the area in which we are interested in keeping the full dynamics.

In order to model the slow inter-area dynamics of the system, we need only one generator per slow-coherent area, since all the other generators in the area move coherently with the retained generators. The generators in each area external to the study area are equivalenced to one aggregate generator model; we then have only one generator model per external area. We reduce the order of the model, but still maintain the inter-area swing dynamics of the system. This model reduction can be carried out using algorithms as in [5].

If we are interested in the fast dynamics of one area (the study area), we keep all the detail, i.e., all the generators, of that area. This would be the case if we were simulating a fault at one of the generators within that area.

In summary, then, we maintain the full model of the study area and use the slow-coherent equivalents for each of the external (non-study) areas.

2.3 Synchrony

A more general notion than slow coherency, termed synchrony, was presented in [14, 15, 16, 17]. Synchrony was used as the basis for another approach to equivalencing called Synchronic Modal Equivalencing (SME).
Background Theory

As in slow-coherency, we are interested in partitioning the generators into areas, choosing a study area, and equivalencing the remaining areas; however, synchrony differs from slow-coherency in two ways. First, synchrony is not limited to looking at the slowest modes. Whereas slow-coherency focuses on behavior in the slowest modes, synchrony selects a subset of modes — a chord \( V \) — that is not necessarily made up of the slowest modes, but which is likely to yield a fruitful partitioning of the system. Second, rather than looking for identical motion of the generators when the chord is excited, synchrony groups generators whose chordal motions are constant linear combinations of the motions of a set of basis generators for each area. In this thesis we will only be interested in one-dimensional synchrony, where we pick only one basis generator per area.

Just as in slow coherency, where the slow modes correspond to the inter-area modes, the modes of chord \( V \) correspond to the inter-area modes in synchrony. The remaining modes correspond to the intra-area dynamics.

2.3.1 Synchronic Modal Equivalencing

Synchronic Modal Equivalencing is reduced-order dynamic modeling using synchrony. First, the generators are partitioned into areas. Next, a study area is chosen. Last, the less-relevant generators, the non-basis generators in external areas, are modeled as dependent current sources, driven by the motions of the basis generators. Load buses that are only connected to each other and to less-relevant generators are termed less relevant load buses, and these may also be equivalenced into the current injectors.

Figure 2.1, taken from [14], shows a simple power system before and after SME-based dynamic equivalencing. Note that all the less-relevant generator and load buses have been equivalenced. There are still load buses \{7–10\} from the full model, which have neither been grouped in the study area nor been equivalenced. This issue will be addressed in Chapter 3 of the thesis.

2.3.2 Generator Partitioning in SME

As mentioned in Section 2.3.1, the generators are partitioned into areas in order to perform the equivalencing in SME. The partitioning algorithm consists of three steps:

- selection of the chord
- selection of the basis generators
- grouping of the non-basis generators.
2.3 Synchrony

A basic overview of the procedure is given here. The full details of the partitioning algorithm can be seen in [14].

The chord $\mathcal{V}$ is selected using a sequential procedure. To begin, two modes of the model (2.3) are selected. These initial modes normally consist of the zero mode and another extensive mode. While different measures of extensiveness exist, the one used in [14] is the spread index, defined as the reciprocal of the variance of the absolute values of the participation factors [13] in the mode. We then look for three synchronic groups within these two modes. The modes corresponding to the inter-area modes of the three groups will include the initial modes plus one more. This additional mode becomes the third mode in the chord. The procedure is then repeated looking for four synchronic groups in the three modes, and so on, until we have the number of modes we desire for the chord.

The matrix of eigenvectors of $A$ in (2.3) corresponding to the chord $\mathcal{V}$, termed the chordal matrix, is constructed as shown:

$$U_\mathcal{V} = \left[ \begin{array}{c|c|c} u_{1\mathcal{V}} & u_{2\mathcal{V}} & \cdots \end{array} \right], \quad (2.7)$$

where $u_{i\mathcal{V}}$ is the eigenvector corresponding to the $i^{th}$ mode of the chord. Each of the rows of $U_\mathcal{V}$ corresponds to one generator. The chord corresponding to the specialized case of slow coherency would only contain the slowest modes; we call this the slow chordal matrix.

The basis generators are selected sequentially from the rows of $U_\mathcal{V}$. The first basis generator chosen is the one with the highest participation [13] in the chord. The second basis generator chosen

---

Figure 2.1: **Before and After SME-Based Equivalencing** [14] On the left is a schematic of a simple power system. On the right we show the same system after equivalencing.
Background Theory

is the one with the highest *synchronic distance* from the first. The synchronic distance of row \( i \) from row \( j \) is given by

\[
d_{ij} = ||a_i|| \sin \phi_{ij}
\]  

(2.8)

where \( a_i \) is the \( i^{th} \) row of \( U_V \), \( ||.|| \) denotes the vector 2-norm, and \( \phi_{ij} \), in the simplest case, is the angle between rows \( i \) and \( j \) of \( U_V \). More general definitions of \( a_i \) and \( \phi_{ij} \) can be used by incorporating appropriate weights, but we do not explore these extensions in this thesis. The third basis generator is that with the highest minimum synchronic distance from the first two. Basis generators are picked in this manner until the number of basis generators equals the number of modes in the chord.

Once the basis generators are picked, the remaining generators are each grouped with one or another of the basis generators, based on their *degree of synchrony* with the basis generators. The degree of synchrony between two generators is the cosine of the angle between the rows of \( U_V \) corresponding to the generators, \( \text{i.e.,} \)

\[
\mu_{ij} = \cos \phi_{ij}
\]  

(2.9)

Each generator is grouped with the basis generator with which it has the highest degree of synchrony. In this way, the generators are partitioned into the number of desired areas.
Chapter 3

Load Bus Partitioning in SME

All previous work with partitioning within the synchrony framework has dealt only with the partitioning of generators into areas; partitioning of load buses has not been performed. Grouping of the load buses, however, is very important in dynamic equivalencing. This is especially true if we wish to simulate a fault at a load bus, because the definition of a study area will depend on how this load bus is assigned.

3.1 Load Bus Partitioning in Slow Coherency

There has been some prior work on load bus partitioning in the slow coherency setting. Some recent work addressing this issue has been done by DeMarco and Wassner [6] and Yusof et al. [20]. It is most clearly seen in the setup used in [6] that the voltage angles of the load buses are also coherent with the generator angles within a slow-coherent area. This observation is explicitly made in [10]. Knowing this, we can easily motivate the partitioning of the loads along with the generators as seen in [20]. Thus, we expand the method explained in Section 2.2.2 (using slow-coherency) to include the partitioning of the load buses. For this, the setup of [6] is used to obtain a generalized eigenanalysis of the structure-preserving swing model — a modification of the swing model in (2.2) that retains load bus variables, as will be fully explained in Section 3.2. We can then construct the generalized slow chordal matrix, which will have rows corresponding to both the generators and the loads, and then group them accordingly.

3.2 Extending SME Partitioning

In this chapter, we will modify the setup and methods of [6] and [20] to naturally extend the methods of SME to include load bus partitioning. Rather than using the slow chordal matrix, we will use the sequential SME procedure to select our chord. We will also group the rows of the chordal matrix — both generator and load bus rows — by looking for synchronic row dependencies as we did with the generators in Section 2.3.2.
3.2.1 Structure-Preserving Swing Model

We first set up the structure-preserving swing model. Unlike in Chapter 2, we want to include both generators and loads in our model. The notation and setup used are similar to that in [6], with some modifications.

The generators are modeled in classical undamped form. Now, however, we also use the algebraic constraints of the loads to obtain the following nonlinear differential-algebraic equation (DAE) description of the power system:

\[
\begin{align*}
\mathbf{J} \ddot{\delta}_G &= p_I^L - p_N^G(\delta, v) \\
0 &= p_L^I(v) - p_L^N(\delta, v) \\
0 &= q_L^I(v) - q_L^N(\delta, v)
\end{align*}
\]

where the vectors \( \delta \) and \( v \) respectively now include the angles and voltage magnitudes at both generator and load buses, \( p_L^I(v) \) and \( q_L^I(v) \) denote the active and reactive power injected at the load buses, and \( p_N^G(\delta, v) \) and \( q_N^L(\delta, v) \) denote the active and reactive power absorbed by the network at the load buses. Throughout the thesis, we will use the subscripts “G” and “L”, respectively, to refer to generator and load parameters or variables. Assume our system has \( n_G \) generators and \( n_L \) loads, and \( n = n_G + n_L \), where \( n \) is the total number of buses. The vector \( \delta \) is the augmented vector of generator and load bus angles, as noted above, while \( p_N(\delta, v) \) is the augmented vector of active power absorbed by the network at generator and load buses:

\[
\begin{align*}
\delta^T &= \begin{bmatrix} \delta_G^T & \delta_L^T \end{bmatrix}^T \\
p_N(\delta, v)^T &= \begin{bmatrix} p_G^N(\delta, v)^T & p_L^N(\delta, v)^T \end{bmatrix}^T
\end{align*}
\]

The \( i^{th} \) component of \( p_N(\delta, v) \) in (3.1) is given by

\[
p_i^N(\delta, v) = v_i \sum_{j=1}^{n} [B]_{ij} v_j \sin(\delta_i - \delta_j)
\]

where the factor \([B]_{ij}\) denotes the transmission line susceptance between buses \( i \) and \( j \), and \( v_i \) is the voltage magnitude at bus \( i \).

To obtain the linearized swing model, we linearize (3.1) about the nominal operating point \((\delta_0, v_0)\). We will assume approximate \((p, \delta)\) and \((q, v)\) decoupling [11] so that we can ignore the reactive power constraints for our analysis throughout the remainder of the thesis (apart from their
3.2 Extending SME Partitioning

In determining the operating point). We now have, in matrix form,

\[
\begin{bmatrix}
J & 0 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
\Delta \delta_G \\
\Delta \delta_L
\end{bmatrix} = -\begin{bmatrix}
\frac{\partial}{\partial \delta} p_G^N(\delta_0, v_0) \\
\frac{\partial}{\partial \delta} p_L^N(\delta_0, v_0)
\end{bmatrix}
\begin{bmatrix}
\Delta \delta_G \\
\Delta \delta_L
\end{bmatrix}
\]

\[
= -\begin{bmatrix}
K_{GG} & K_{GL} \\
K_{LG} & K_{LL}
\end{bmatrix}
\begin{bmatrix}
\Delta \delta_G \\
\Delta \delta_L
\end{bmatrix}
\]  

(3.4)

Within the \( K \) matrix, the submatrix \( K_{GG} \) represents interactions among generators, \( K_{LL} \) represents interactions among loads, \( K_{GL} \) and \( K_{LG} \) represent interactions between generators and loads. Eqn. (3.4) is more compactly written as

\[
M \Delta \ddot{\delta} = -K \Delta \delta . 
\]  

(3.5)

This is our linearized structure-preserving undamped swing model. The vector \( \Delta \delta \in \mathbb{R}^n \) denotes the perturbation of the bus angles from the operating point \((\delta_0, v_0)\), and the matrices \( M \in \mathbb{R}^{n \times n} \) and \( K \in \mathbb{R}^{n \times n} \) are defined as in (3.4). Note that \([K]_{ij}\), the \((ij)\)th entry in \( K \), is

\[
[K]_{ij} = \begin{cases}
-v_i v_j [B]_{ij} \cos(\delta_i - \delta_j) & \text{if } i \neq j \\
-\sum_{t=1, t \neq i}^{n} [K]_{it} & \text{if } i = j
\end{cases}
\]  

(3.6)

In previous SME work, the collapsed version of (3.4), i.e., with algebraic variables eliminated, was used. This allowed for the use of regular (rather than generalized) eigenanalysis to find the chord and partition the generators. This regular, linearized model is easily seen to be

\[
\Delta \ddot{\delta}_G = -J^{-1}(K_{GG} - K_{GL} K_{LL}^{-1} K_{LG}) \Delta \delta_G . 
\]  

(3.7)

This is of the same form as (2.3), so the definition of \( K_G \) is now evident. Note that in this collapsed form, the load bus variables are lost; the model hides the load bus behavior and is therefore not useful for load bus partitioning.
3.2.2 Generalized Eigenanalysis

We are interested in retaining our load bus variables for partitioning. Rather than collapsing
the model, we keep it in the form of (3.5). To find the system modes, we solve the generalized
eigenproblem

\[(\lambda M + K)u = 0, \quad u \neq 0.\]  

(3.8)

The corresponding system modes are of the form

\[\Delta \delta(t) = ue^{tj\sqrt{-\lambda}}.\]  

(3.9)

There exist \(n_G\) finite values of \(\lambda\) for which (3.8) holds. (The remaining \(n_L\) values will be at infinity;
we are only interested in the finite values of \(\lambda\).) The finite \(\lambda\)'s for which (3.8) holds are real, non-
positive, and are precisely the roots \(\lambda_0, \lambda_1, \ldots, \lambda_{n_G-1}\) of the polynomial \(\det(\lambda M + K) = 0\). The
\(\lambda_i\)'s are termed generalized eigenvalues and the associated \(u_i\)'s are generalized eigenvectors. Note,
however, that these \(\lambda_i\)'s are precisely the eigenvalues of \(A\) in (3.7) and (2.3).

We denote by \(U \in \mathbb{R}^{n \times n_G}\) the matrix whose columns are the generalized eigenvectors in (3.8).
The first \(n_G\) rows correspond to generator variables, while the last \(n_L\) rows correspond to load
variables. We correspondingly partition \(U\) into two matrices, \(U_G\) and \(U_L\):

\[U = \begin{bmatrix} U_G \\ U_L \end{bmatrix}, \quad n_G + n_L.\]  

(3.10)

Manipulating (3.8), we get the following system of equations for a chosen vector \(u\) from \(U\):

\[-\lambda J u_G = K_{GG} u_G + K_{GL} u_L \]  

(3.11a)

\[0 = K_{LG} u_G + K_{LL} u_L .\]  

(3.11b)

Solving (3.11b) for \(u_L\) and substituting into (3.11a), we get

\[\lambda u_G = -J^{-1}(K_{GG} - K_{GL} K_{LL}^{-1} K_{LG}) u_G ,\]  

(3.12)

which establishes that the columns of \(U_G\) are precisely the eigenvectors of the regular model (3.7).
3.2.3 Load Bus Partitioning

Using the generalized eigenstructure of the system, we can now extend the SME partitioning algorithm to include grouping of the load buses. We still have the same three major steps: chord selection, basis generator selection, and grouping of the buses. Now, however, we include the grouping of the load buses in the third step.

We are still primarily interested in the generators to determine the area partitioning. Because of this, we use only the generator variables when performing the chord selection. Thus, chord selection can be done with the regular model (3.7) or the $U_G$ matrix in the extended DAE setting. As shown by (3.12), these are equivalent. We denote our chordal matrix by $U_Y \in \mathbb{R}^{n \times p}$, where $p$ is the number of modes of interest to us, i.e., the number of areas we wish to have. $U_Y$ has as its columns the full generalized eigenvectors corresponding to the modes selected in the chord.

For the second step, we select our basis generators from the rows of $U_Y$ corresponding to generator variables. Working with the first $n_G$ rows of $U_Y$, we select our basis generators in standard, sequential SME fashion.

Once we have determined the basis generators, we group each of the remaining non-basis generators and each of the load buses with one of the basis generators. Each row of $U_Y$ is grouped with the basis machine with which it has the highest degree of synchrony. The full chordal matrix $U_Y$ includes the load buses, whereas the chordal matrix derived from (3.7) did not.

When we group the buses, it is necessary to pay attention to all the areas with which the bus has a high degree of synchrony. If a bus has nearly equal degrees of synchrony with more than one area, the bus may not be “well-partitioned”; it may be difficult to definitively assign the bus to one area. If we wish to simulate a perturbation at that bus, it may be necessary to include the full dynamics of both areas in the study area. This subject will be addressed further in Section 3.4.

3.3 Test Results

In this section we present the results of applying the load bus partitioning algorithm to two different models. The line, bus, and operating point data were all taken from EUROSTAG [7] models. All of the calculations necessary to effect the partitioning were done in MATLAB [12].
3.3.1 New England Model

The first model used to test the results of the load bus partitioning is the New England model used in [1, 16]. The model comprises 10 generators, 39 buses, and 46 lines. The line diagram for this model is shown in Fig. 3.1. The loads are numbered 1 to 29; the generators, 30 to 39.

From the model data, the $M$ and $K$ matrices are formed and the generalized eigenvalue/eigenvector pairs of equation (3.8) are found using the QZ algorithm in MATLAB. Here, the network partitioning was carried out for two different cases. In the first case, the same operating point as in [14] was used, but the basis machines were chosen using our eigenanalysis results. For the second case, both the operating point and the basis machines from [14] were used. In both cases, the network was partitioned into three areas.

Table 3.1 lists the eigenvalues for the New England model. The chord selected was $\mathcal{V} = [\lambda_0, \lambda_1, \lambda_3]$. Note that the modes selected do not correspond to the slow chord. Mode 1 corresponds

\[
\begin{array}{cccccc}
\lambda_0 & \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \cdots \\
0 & -20.72 & -67.34 & -84.19 & -100.51 & \cdots
\end{array}
\]

Table 3.1: New England Model: Eigenvalues
to the massive machine \{10\} swinging against the rest of the system, while mode 3 corresponds to \{1, 2, 3\} (and to some extent, \{8\}) swinging against the rest of the system.

**Case I: Nominal Operating Point, Automatic Basis Selection**

Referring to the diagram in Fig. 3.1, the buses were partitioned into the following areas, with the basis generator bus in each area shown in boldface:

(a) \{ 1, 9, 39 \}

(b) \{ 2–8, 10–14, 30, 31, 32 \}

(c) \{ 15–29, 33, 34, 35, 36, 37, 38 \}

We can see from Fig. 3.1 that the areas have reasonably distinct boundaries. They do not cross over each other and only adjacent areas are connected; areas (a) and (c) have no lines connecting them.

**Case II: Nominal Operating Point, Basis Selection from [14]**

To more easily compare our results with those in [14], we forced the basis machines to be the same as there, namely \{30, 35, 39\}. Referring again to Fig. 3.1, the buses were partitioned into the following areas:

(a) \{ 1, 9, 39 \}

(b) \{ 2–8, 10–15, 18, 25, 30, 31, 32, 37 \}

(c) \{ 16, 17, 19–24, 26–29, 33, 34, 35, 36, 38 \}

The areas still have reasonably distinct boundaries. Only one generator bus, \{8\}, and three load buses, \{15, 18, 25\}, switch areas. These four buses are all on the border between areas (b) and (c). In 3.4 we will further discuss the switching buses. As far as the generator grouping is concerned, the results shown above are the same as those obtained in [14] using regular eigenvectors from the collapsed model.

### 3.3.2 France-Spain Model

The second model used is the same as in [17], and is based on the France-Spain power system. The model comprises 23 generators, 83 buses, and 224 transmission lines. Note that this model
is the result of some prior equivalencing process, so the generators and load buses can not be assigned clear identities and geographical locations, which limits our ability to assess the quality of the partitionings obtained with our approach. The approximate geographical locations of the generators in this model are shown in Fig. 3.2. The generators are numbered 1 to 23; the loads, 24 to 83.

The reduced model (3.7) is obtained by a slightly different route than the comparable model in [16]. Nevertheless, the eigenstructures of these models are sufficiently close to each other such that the chord used in [16] can also be extracted for our experiments. We use this chord for all the tests reported here.

As with the New England model, the network partitioning was carried out for different cases. In the first case, we assumed a flat start as in [6] where $\delta_0 = 0$ and $v_0 = [1 \ldots 1]^T$. In the second case, the same operating point as in [16] was used, but the basis machines were chosen using our eigenanalysis results. For the third case, both the operating point and the basis machines from [16] were used.

Table 3.2 lists the eigenvalues for the New England model. The chord selected was $\mathcal{V} = \ldots \ldots$
3.3 Test Results

<table>
<thead>
<tr>
<th>$\lambda_0$</th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>$\lambda_3$</th>
<th>$\lambda_4$</th>
<th>$\lambda_5$</th>
<th>$\lambda_6$</th>
<th>$\lambda_7$</th>
<th>$\cdots$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-3.26</td>
<td>-7.54</td>
<td>-14.76</td>
<td>-16.22</td>
<td>-24.50</td>
<td>-27.79</td>
<td>-41.69</td>
<td>$\cdots$</td>
</tr>
</tbody>
</table>

Table 3.2: France-Spain Model: Eigenvalues

$[\lambda_0, \lambda_1, \lambda_3, \lambda_4, \lambda_6]$. Note that, again, the modes selected do not correspond to the slow chord.

Case I: Flat Start, Automatic Basis Selection

In this case, the $K$ matrix reduces to the admittance matrix of the network. Though some detail about the operating point of the network is lost — the bus voltages and angles at the operating point are not used — the results are still good. Referring to the map in Fig. 3.2, the buses were partitioned into the following areas, with the basis generator in each area shown in boldface:

(a) $\{10, 11, 12, 13\}$
(b) $\{14, 68, 73, 78\}$
(c) $\{18\}$
(d) $\{1, 7, 9, 19, 20, 24-29, 32-39, 59, 61, 62, 82, 83\}$
(e) $\{2, 3, 4, 5, 6, 8, 15, 16, 17, 21, 22, 23, 30, 31, 40-58, 60, 63-67, 69-72, 74-77, 79, 80, 81\}$

Looking strictly at the generator groupings (buses 1-23), we can see from Fig. 3.2 that the areas have geographically reasonable boundaries. Fig. 3.3 shows the number of lines interconnecting the different areas. Although the figure happens to be for Case III below, a very similar structure is observed for all three cases. Note that the partitioning is reasonable in the sense that links exist only between geographically adjacent areas.

Case II: Nominal Operating Point, Automatic Basis Selection

For this case the $K$ specified in (3.6) was used. Referring again to Fig. 3.2, the buses were partitioned into the following areas:

(a) $\{10, 11, 12 \textbf{13}\}$
(b) $\{14, 68, 73, 78\}$
(c) $\{18\}$
(d) $\{1, 7, 8, 9, 19, \textbf{20}, 24-29, 32-39, 59, 61, 62, 82, 83\}$
The areas still have geographically reasonable boundaries. Note that only one generator, \{8\}, and no loads change areas. The generator is on the border between two adjacent areas, (d) and (e). As was mentioned in 3.3.1, the issue of switching buses will be addressed in Section 3.4.

Case III: Nominal Operating Point, Basis Selection from [16]

To more easily compare the results of generator partitioning with those in [16], we forced the basis machines to be the same, \{13, 14, 18, 19, 21\}. The buses then ended up being partitioned into the following areas:

\[
\begin{align*}
(a) & \{ 10, 11, 12, 13 \} \\
(b) & \{ 14, 68, 73, 78 \} \\
(c) & \{ 18 \} \\
(d) & \{ 1, 2, 3, 7, 8, 9, 19, 20, 24-29, 32-39, 51-59, 61, 62, 82, 83 \} \\
(e) & \{ 4, 5, 6, 15, 16, 17, 21, 22, 23, 30, 31, 40-50, 60, 63-67, 69-72, 74-77, 79, 80, 81 \}
\end{align*}
\]

Two more generators, \{2\} and \{3\}, and eight loads \{51-58\} switched from (e) to (d). Load buses \{51-58\} are only connected to each other and to generators \{2, 3\}. Because of this, we would indeed expect these load buses to move along with generators \{2, 3\}.

So far as the generator grouping is concerned, the results shown above are close to those obtained in [16] using regular eigenvectors of a slightly different model than (3.7). The only difference is that the three generators \{4, 5, 6\} are assigned to (e) here and to (d) in [15].

3.4 Degree of Synchrony in Boundary Buses

We now address the issue of the generator and load buses that switch areas when the operating point is varied slightly. Looking at the degrees of synchrony which each such "boundary" bus has with the basis generators gives us more insight.

Looking first at the New England model, we see from Table 3.3 that, in Case I, the generator at bus \{37\} has nearly equal degrees of synchrony with areas (a) and (b). For generator \{37\} and the loads \{15, 18, 25\} in Case II, even though the degrees of synchrony are not nearly equal, the
3.4 Degree of Synchrony in Boundary Buses

Figure 3.3: Schematic Representation of Areas and Their Interconnections. This schematic map shows the number of lines interconnecting the different areas; the letters in the circles correspond to those used in the text to denote the various areas.

<table>
<thead>
<tr>
<th></th>
<th>I(a)</th>
<th>I(b)</th>
<th>II(a)</th>
<th>II(b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>G37</td>
<td>0.6123</td>
<td>0.6891</td>
<td>0.9778</td>
<td>0.8420</td>
</tr>
<tr>
<td>L15</td>
<td>0.5780</td>
<td>0.7307</td>
<td>0.9607</td>
<td>0.8754</td>
</tr>
<tr>
<td>L18</td>
<td>0.5492</td>
<td>0.7477</td>
<td>0.9565</td>
<td>0.8850</td>
</tr>
<tr>
<td>L25</td>
<td>0.5748</td>
<td>0.7128</td>
<td>0.9689</td>
<td>0.8563</td>
</tr>
</tbody>
</table>

Table 3.3: New England Model: Degrees of Synchrony with Different Areas. The entries show, for certain selected buses, the degrees of synchrony with the basis machine for the indicated area. The column heading indicates the case (I, II) and the area. The row label indicates the bus number.

buses still have a very high degree of synchrony with both basis machines. It is therefore not unexpected that the assignment of these buses is quite sensitive. To study a fault at any of these buses, we would probably have to retain both areas (a) and (b) in the study area.

Looking at the France-Spain model illustrates the point even better. First, we recall that only one generator, {8}, and no loads change areas from Case I to Case II. In both Case I and Case II, the degrees of synchrony relating {8} to the basis machines {15} (Group e) and {20} (Group d) were nearly equal, as seen in Table 3.4; hence, the switching of assignment is not surprising. Further, the degrees of synchrony are relatively small in both cases, so the change in assignment is not very significant. In order to simulate a fault associated with {8}, it may be necessary to include both areas (d) and (e) in the study area to maintain sufficient accuracy.

From Case II to Case III, two more generators, {2} and {3}, and eight loads {51–58} switched from (e) to (d). As before, the degrees of synchrony relating the buses that switched were nearly
Load Bus Partitioning in SME

<table>
<thead>
<tr>
<th></th>
<th>I(d)</th>
<th>I(e)</th>
<th>II(d)</th>
<th>II(e)</th>
<th>III(d)</th>
<th>III(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>G2</td>
<td>0.6100</td>
<td>0.6268</td>
<td>0.6165</td>
<td>0.6276</td>
<td>0.6952</td>
<td>0.6787</td>
</tr>
<tr>
<td>G3</td>
<td>0.6100</td>
<td>0.6268</td>
<td>0.6165</td>
<td>0.6276</td>
<td>0.6952</td>
<td>0.6787</td>
</tr>
<tr>
<td>G8</td>
<td>0.6179</td>
<td>0.6185</td>
<td>0.6219</td>
<td>0.6215</td>
<td>0.7002</td>
<td>0.6726</td>
</tr>
<tr>
<td>L51-58</td>
<td>0.6100</td>
<td>0.6268</td>
<td>0.6165</td>
<td>0.6276</td>
<td>0.6952</td>
<td>0.6787</td>
</tr>
</tbody>
</table>

Table 3.4: France-Spain Model: Degrees of Synchrony with Different Areas. The entries show, for certain selected buses, the degrees of synchrony with the basis machine for the indicated area. The column heading indicates the case (I,II,III) and the area. The row label indicates the bus number.

Table 3.4. In order to simulate a fault associated with generators \{2\} or \{3\} or with load buses \{51-58\}, it may be necessary to include both areas (d) and (e) in the study area to maintain sufficient accuracy. The ability to partition the load buses has given us more information about the system.
Chapter 4

Modal Sensitivity

If we consider a fully partitioned network, we may call the lines connecting buses in different areas a cutset. If we set the admittances of the cutset to zero, the system is fully decoupled. We saw in Section 2.2 that the number of zero eigenvalues in this case is equal to the number of areas in the network.

It is necessary to understand what occurs when the line admittances are incrementally increased, joining the decoupled areas to form one network. This is the setting of slow-coherency, specifically, we look at the sensitivity of the system eigenvalues to changes in the line parameters. It is also useful to determine the sensitivity of the eigenvectors to perturbations in the admittances of the cutset. We would like to know to what order the eigenvectors, and particularly those associated with the slow modes, change as the cutset admittances are increased from 0 to $\epsilon$. If the changes are on the order of $\epsilon$ or $\epsilon^2$, the structure of the eigenvectors would only change minimally with small $\epsilon$. This would add some justification to the load bus partitioning of slow coherency [20].

Sensitivity of an eigenvalue to changes in a parameter. We begin with the generalized eigenvalue/eigenvector equation (3.8):

$$ (\lambda M + K)u = 0, \quad u \neq 0. \quad (4.1) $$

From this, it is easily shown that, to first order, the change in a particular $\lambda_i$ can be written

$$ \Delta \lambda_i = -\frac{w_i^T(\Delta K + \lambda \Delta M)u_i}{w_i^T Mu_i}, \quad (4.2) $$

where $w_i$ is the left eigenvector of (4.1) corresponding to $\lambda_i$. (See Appendix B for full details.)

Recalling that in the setup used in this thesis both matrices $M$ and $K$ are symmetric, and that the left and right eigenvectors are the same for a symmetric system, (4.2) can be rewritten as

$$ \Delta \lambda_i = -\frac{u_i^T(\Delta K + \lambda \Delta M)u_i}{u_i^T Mu_i}. \quad (4.3) $$

We also know that $M$ does not depend on the line parameters which we will perturb; it is only
Modal Sensitivity

dependent on generator inertias. Thus, the change in $M$ due to parameter changes is zero, i.e., $\Delta M = 0$. This allows us to state $\Delta \lambda_i$ in its simplest form.

$$\Delta \lambda_i = \frac{u_i^T(-\Delta K)u_i}{u_i^TMu_i}$$  \hfill (4.4)

### 4.1 Eigenvector Sensitivity

The sensitivity of the eigenvectors to parameter perturbations is not as easily derived as (4.4). For the remainder of this chapter we drop the subscript $i$ for clarity. From the derivation in [9] (see Appendix B for full details), we find the change in the eigenvector $u$ to be

$$\Delta u = -[(K + \lambda M)^2 + 2Muu^TM]^{-1}[(K + \lambda M)\Delta(K + \lambda M) + Muu^T\Delta M]u.$$ \hfill (4.5)

The above holds only for symmetric matrices $M$ and $K$. Expanding the $\Delta(K + \lambda M)$ term, we find that

$$\Delta u = -[(K + \lambda M)^2 + 2Muu^TM]^{-1}[(K + \lambda M)(\Delta K + (\Delta \lambda)M + \lambda \Delta M) + Muu^T\Delta M]u.$$ \hfill (4.6)

Since in our case, $\Delta M = 0$, (4.6) reduces to

$$\Delta u = -[(K + \lambda M)^2 + 2Muu^TM]^{-1}[K + \lambda M][\Delta K + M\Delta \lambda]u.$$ \hfill (4.7)

Now, substituting the result for $\Delta \lambda$ from (4.4) into (4.7), we get

$$\Delta u = -[(K + \lambda M)^2 + 2Muu^TM]^{-1}[K + \lambda M] \left[ \Delta K + M \frac{u^T(-\Delta K)u}{u^TMu} \right] u$$ \hfill (4.8)

as the final result.

The first two terms of (4.8), $[(K + \lambda M)^2 + 2Muu^TM]^{-1}$ and $[K + \lambda M]$, are both constant terms; they do not depend on the perturbation of the line parameters. Only the third term, $\left[ \Delta K + M \frac{u^T(-\Delta K)u}{u^TMu} \right]$, depends on $\epsilon$. In fact, within the third term, only $\Delta K$, the effect of the $\epsilon$ perturbation on the $K$ matrix, is dependent on $\epsilon$.

The line parameter terms in $K$ are all of first order, as seen in (3.6). With a perturbation on the line parameters of order $\epsilon$, $\Delta K$ will also be of order $\epsilon$. This also implies that

- 38 -
4.2 Test Results

First we will test the results of the eigen-sensitivity derivations on a three-generator model. Using this model, we will be able to use regular eigenanalysis to test the derivations, both numerically and symbolically. Then we will use a two-generator, one-load model to test the results on a case requiring generalized eigenanalysis.

4.2.1 Three Generators

The three generator model used is shown in Fig. 4.1. For simplicity, we will assume the generators all have equal inertias. We will look at the lossless, flat-start case, where the bus angles are equal, the bus voltages are equal and set to one, and \( b_{ij} \) denotes the susceptance of the line between buses \( i \) and \( j \). In this case, \( K \) reduces to the admittance matrix. First, we assume the decoupled system where \( b_{12} = b_{13} = 0 \), i.e., the system is partitioned into two groups, \( \{1\} \) and \( \{2, 3\} \). The \( K \) matrix
for this system in the form of (2.3) is

$$ K = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b_{23} & -b_{23} \\ 0 & -b_{23} & b_{23} \end{bmatrix} . \quad (4.10) $$

If the generator inertias are equal to 1, the $M$ matrix will simply be the $3 \times 3$ identity matrix. The eigenvalues of $-K$, denoted by $\lambda_0, \lambda_1, \lambda_2$, are:

$$ \lambda = \begin{bmatrix} 0 & 0 & -2b_{23} \end{bmatrix} , \quad (4.11) $$

with corresponding right eigenvectors:

$$ U = \begin{bmatrix} 1 & -2 & 0 \\ 1 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix} . \quad (4.12) $$

If we now begin to couple the two areas by increasing $b_{12}$ to $\epsilon$, the new $K$ and $\Delta K$ matrices are:

$$ K = \begin{bmatrix} \epsilon & -\epsilon & 0 \\ -\epsilon & b_{23} + \epsilon & -b_{23} \\ 0 & -b_{23} & b_{23} \end{bmatrix} , \quad \Delta K = \begin{bmatrix} \epsilon & -\epsilon & 0 \\ -\epsilon & \epsilon & 0 \\ 0 & 0 & 0 \end{bmatrix} . \quad (4.13) $$

Using the formula in (4.4), we can find $\Delta \lambda_i$ (the approximate change in $\lambda_i$) as a function of $u_i$, $\Delta K$, and $M$.

$$ \Delta \lambda_1 = \begin{bmatrix} \epsilon & -\epsilon & 0 \\ -\epsilon & \epsilon & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix} , \quad \Delta \lambda_1 = -\frac{3}{2\epsilon} \quad (4.14) $$

Similarly,

$$ \Delta \lambda_0 = 0 \quad , \quad \Delta \lambda_2 = -\frac{1}{2\epsilon} \quad (4.15) $$

Let us now assume that $b_{23} = 2$. For $\epsilon = 0$, the eigenvalues are now $\lambda = \begin{bmatrix} 0 & 0 & -4 \end{bmatrix}^T$ and the
eigenvectors are as in (4.12). If we increase $b_{12}$ such that $\epsilon = .01$, we would expect from (4.14) and (4.15) that our approximation for the new $\lambda$ would be

$$\hat{\lambda}_{\text{new}} = \begin{bmatrix} 0 & 0 & -4 \end{bmatrix} + \begin{bmatrix} 0 & -0.015 & -0.005 \end{bmatrix} = \begin{bmatrix} 0 & -0.015 & -4.005 \end{bmatrix}. \quad (4.16)$$

Finding the new eigenvalues strictly numerically, we find that

$$\lambda_{\text{new}} = \begin{bmatrix} 0 & -0.01498 & -4.00502 \end{bmatrix} \approx \begin{bmatrix} 0 & -0.015 & -4.005 \end{bmatrix} = \hat{\lambda}_{\text{new}}. \quad (4.17)$$

As we can see from (4.17), the approximation given by (4.4) is very good — the maximum error is less than one-fifth of one percent.

In order to find $\Delta u_i$, we apply (4.8), using the same quantities as above. We find

$$\Delta u_2 = \begin{bmatrix} .0025 \\ -.00125 \\ -.00125 \end{bmatrix}, \quad \hat{u}_{2\text{new}} = \begin{bmatrix} .0025 \\ -1.00125 \\ .99875 \end{bmatrix}. \quad (4.18)$$

There is a problem with (4.8); when $\lambda_i = 0$, the first term of the equation is singular and therefore not invertible. Even if this term is singular, however, we can still conclude that

$$(K + 2Muu^TM)\Delta u = -(K + \lambda M)(\Delta K + M\Delta \lambda)u. \quad (4.19)$$

From (4.19) we can see that the change in $u$ is still only dependent upon the same order $\epsilon$ matrix. We also know that $\lambda_0$ and $u_0$ do not change for a swing-equation model. We can find $\Delta u_1$ by initially setting $b_{12} = .001$ and then increasing $b_{12}$ by $\epsilon = .009$. The final $b_{12}$ still equals .01. Since we are only interested in showing that the approximation is good and that the eigenvector perturbations are therefore only order $\epsilon$, this should suffice. Comparing the results of the approximation to the actual numerical result, we get

$$U_{\text{new}} = \begin{bmatrix} 1 & -2.00157 & .002506 \\ 1 & .997 & -1.001256 \\ 1 & 1.0045 & .99875 \end{bmatrix} \approx \begin{bmatrix} 1 & -2.00157 & .0025 \\ 1 & .99577 & -1.00125 \\ 1 & 1.00327 & .99875 \end{bmatrix} = \hat{U}_{\text{new}}. \quad (4.20)$$

We see that the approximations for the changes in the eigenvectors are good as well. Since (4.8) is valid, the eigenvector perturbations must be of order $\epsilon$. 

---

4.2 Test Results
Modal Sensitivity

Figure 4.2: Two Generator/One Load Model This figure shows a model of a simple two-generator, one-load power system. We assume lossless lines; $b_{ij}$ are the susceptances between buses $i$ and $j$.

4.2.2 Two Generators, One Load

In Fig. 4.2 we have a load rather than a generator at Bus 3. This changes the structure of the $M$ matrix:

$$
M = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}.
$$

(4.21)

It is no longer invertible, so we must use generalized eigenanalysis. This does not, however, affect the validity of our sensitivity formulas.

To begin, we find the generalized eigenvalue/eigenvector pairs of our decoupled system. If we assume $b_{23} = 2$ as above, we find the eigenvalues to be

$$
\lambda = \begin{bmatrix}
0 & 0 & -\infty \\
\end{bmatrix}.
$$

(4.22)

with the corresponding right eigenvectors

$$
U = \begin{bmatrix}
1 & -1 & 0 \\
1 & 1 & 0 \\
1 & 1 & 1 \\
\end{bmatrix}.
$$

(4.23)

In the remainder of the results we will ignore the eigenvalue/eigenvector pair at $-\infty$, since it is not currently of interest to us. Though we will state the numerical results for completeness, we will not find $\Delta \lambda_2$ and $\Delta u_2$. 

- 42 -
First we perturb $b_{12}$, the line between the two generators, such that $b_{12} = \epsilon$. The formula in (4.4) still holds, and yields

$$\Delta \lambda = \begin{bmatrix} 0 & -2\epsilon & * \end{bmatrix} .$$

For $\epsilon = .01$, the estimated and actual results follow.

$$\lambda_{\text{new}} = \begin{bmatrix} 0 & -.02 & -\infty \end{bmatrix} \approx \begin{bmatrix} 0 & -.02 & -\infty \end{bmatrix} = \hat{\lambda}_{\text{new}} \quad (4.25a)$$

$$U_{\text{new}} = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \hat{U}_{\text{new}} \quad (4.25b)$$

The approximation is also good if we perturb line $b_{13}$ between a generator and a load to $\epsilon$. The approximation for $\Delta \lambda$ is the same as in (4.24). For $\epsilon = .01$, the estimated and actual results follow.

$$\lambda_{\text{new}} = \begin{bmatrix} 0 & -.0199 & -\infty \end{bmatrix} \approx \begin{bmatrix} 0 & -.02 & -\infty \end{bmatrix} = \hat{\lambda}_{\text{new}} \quad (4.26a)$$

$$U_{\text{new}} = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & .99005 & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 1 & .99001 & 1 \end{bmatrix} = \hat{U}_{\text{new}} \quad (4.26b)$$

Our approximations in (4.4) and (4.8) are good for both the regular case, involving only generators, and the generalized case with both generators and loads. These formulas show that the effect of $\epsilon$ perturbations on both the eigenvalues and eigenvectors of the system will only be of order $\epsilon$. 

---

4.2 Test Results

---
Modal Patterns

Slow coherency and synchrony exploit modal patterns in power systems. In this chapter, we use a simple example to study variations in relative modal patterns as parameter values are varied. This yields some insight into the relationship between slow coherency and synchrony.

We will begin with three decoupled, identical subsystems or areas. Each area has three generators, so the full system has nine generators. In our example, we connect the three areas as shown in Fig. 5.1.

5.1 System Setup

In our system, we have nine generators connected in three tiers, each tier comprising three generators, as shown in Fig. 5.1. This may be equivalently seen as three concentric circles of three generators each, lying in a plane. We first assume that each of the tiers is strongly connected, and the lines between the tiers are cut, i.e., the system consists of three fully decoupled areas, each tier representing one area. If we look at the flat start case as we have done in previous chapters, the $K$ matrix for each area reduces to the admittance matrix.

5.2 The Decoupled System

We denote by $\bar{A}$ the negative of the admittance matrix for one tier. If we assume identical tiers, $\bar{A}$ will be identical for each tier. If $A$ is the negative of the admittance matrix for the entire system, we have the following for the fully decoupled system:

$$\bar{A} = -\begin{bmatrix} b_{12} + b_{13} & -b_{12} & -b_{13} \\ -b_{12} & b_{12} + b_{23} & -b_{23} \\ -b_{13} & -b_{23} & b_{13} + b_{23} \end{bmatrix}, \quad A = \begin{bmatrix} \bar{A} & 0 & 0 \\ 0 & \bar{A} & 0 \\ 0 & 0 & \bar{A} \end{bmatrix}, \quad (5.1)$$
where $b_{ij}$ is the susceptance between buses $i$ and $j$. As in Chapter 4, we will assume the generator inertias are identical and equal to 1; the $M$ matrix is then the identity matrix.

Setting $b_{12} = b_{13} = b_{23} = 10$, we get the following eigenvalues for $\bar{A}$:

$$\bar{\lambda} = \begin{bmatrix} -30 & -30 & 0 \end{bmatrix}$$

with the corresponding right eigenvectors:

$$\bar{U} = \begin{bmatrix} .8003 & .1619 & .5774 \\ -.2599 & -.7740 & .5774 \\ -.5404 & .6121 & .5774 \end{bmatrix}.$$
The system matrix $A$ then has eigenvalues:

$$\lambda = \begin{bmatrix} \bar{\lambda}^T & \bar{\lambda}^T & \bar{\lambda}^T \end{bmatrix}$$

(5.4)

with the corresponding right eigenvector matrix:

$$U = \begin{bmatrix} \bar{U} & 0 & 0 \\ 0 & \bar{U} & 0 \\ 0 & 0 & \bar{U} \end{bmatrix}.$$  

(5.5)

The eigenvalues of each area are repeated, so we have six eigenvalues at $\lambda = -30$ and three zero eigenvalues, one for each area.

5.3 The Coupled System

We now weakly couple the systems by adding the lines between the areas with susceptances $b_{14} = b_{25} = b_{36} = .001$ and $b_{47} = b_{58} = b_{69} = .002$. We will denote our eigenvalues in ascending order as $\lambda_8 \leq \lambda_7 \leq \cdots \leq \lambda_1 \leq \lambda_0 = 0$. The eigenvalues of the coupled system have shifted to:

$$\lambda = \begin{bmatrix} -30.0047 & -30.0047 & -30.0013 & -30.0013 & -30 & -30 & -0.0047 & -0.0013 & 0 \end{bmatrix}$$

(5.6)

with corresponding eigenvectors:


(5.7)

Let $u_i$ denote the column of $U$, or eigenvector, corresponding to $\lambda_i$. We now examine $U$ in (5.7), recalling that in synchrony we are looking for rows to be exact linear combinations of other rows, looking only at the chord $\mathcal{V}$.

First we look at the slow chord, $\mathcal{V} = [\lambda_0, \lambda_1, \lambda_2]$. If we let $a_k$ denote the $k^{th}$ row of the
chord, we note that \( a_1 = a_2 = a_3, a_4 = a_5 = a_6, \) and \( a_7 = a_8 = a_9. \) Both synchrony and slow-coherency methods would lead us to group the buses into the following three groups, with the basis generators chosen by SME shown in bold: \( \{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}. \) With the line parameters in this case, SME chooses the slow chord for use in the chordal matrix, when started with \( [\lambda_0, \lambda_1]. \)

We can also see synchrony within these groups by looking at a chord other than the slow chord. If we consider the complement of the slow chord, \( V = [\lambda_3 \ldots \lambda_8], \) we will see that

\[
\begin{align*}
    a_1 + a_2 + a_3 &= 0 \\
    a_4 + a_5 + a_6 &= 0 \\
    a_7 + a_8 + a_9 &= 0
\end{align*}
\]

Within each of the groups, one of the generator motions is a linear combination of the other two.

Suppose, however, we chose our chord to be \( V = [\lambda_0 \lambda_3 \lambda_4]. \) With this chord, we note that \( a_1 = a_4 = a_7, a_2 = a_5 = a_8, \) and \( a_3 = a_6 = a_9, \) leading to the grouping of \( \{1, 4, 7\}, \{2, 5, 8\}, \) and \( \{3, 6, 9\}. \) There are no basis generators shown here because SME would not choose this chord. As above, if we look at the complement of this chord, synchrony is also evident.

\[
\begin{align*}
    a_1 + a_4 + a_7 &= 0 \\
    a_2 + a_5 + a_8 &= 0 \\
    a_3 + a_6 + a_9 &= 0
\end{align*}
\]

Even though the chord chosen by SME with these line parameters corresponds to the slow chord, we do see that exciting a different set of modes could lead to distinctly different groupings.

At the opposite end of the spectrum, when we have weak coupling within each tier and strong coupling between the tiers (for this example \( b_{12} = b_{13} = b_{23} = 10, b_{14} = b_{36} = b_{25} = 75, \) \( b_{47} = b_{69} = b_{58} = 150), \) the eigenvalues become:

\[
\lambda = \begin{bmatrix}
-384.90 & -384.90 & -354.09 & -125.10 & -125.10 & -95.10 & -30 & -30 & 0
\end{bmatrix}
\]

\( \text{(5.10)} \)
5.3 The Coupled System

with corresponding right eigenvectors:

\[
U = \begin{bmatrix}
-0.1196 & -0.1244 & 0.1220 & -0.5576 & -0.3220 & -0.4553 & 0.4360 & -0.1792 & 0.3333 \\
-0.0479 & 0.1658 & 0.1220 & -0.5577 & -0.3219 & -0.4553 & -0.3732 & -0.2880 & 0.3333 \\
0.1675 & -0.0414 & 0.1220 & -0.0001 & 0.6440 & -0.4553 & -0.0628 & 0.4672 & 0.3333 \\
-0.4673 & 0.4642 & -0.4553 & 0.1494 & 0.0863 & 0.1220 & 0.4360 & -0.1792 & 0.3333 \\
0.1788 & -0.6186 & -0.4553 & -0.1494 & 0.0863 & 0.1220 & -0.3732 & -0.2880 & 0.3333 \\
-0.6252 & 0.1544 & -0.4553 & 0.0000 & -0.1725 & 0.1220 & 0.0628 & 0.4672 & 0.3333 \\
-0.3267 & -0.3398 & 0.3333 & 0.4082 & 0.2358 & 0.3333 & 0.4360 & -0.1792 & 0.3333 \\
-0.1309 & -0.4529 & 0.3333 & -0.4083 & 0.2356 & 0.3333 & -0.3732 & -0.2880 & 0.3333 \\
0.4576 & -0.1131 & 0.3333 & 0.0001 & -0.4714 & 0.3333 & -0.0628 & 0.4672 & 0.3333 \\
\end{bmatrix}.
\]

Our slow chord, \( V = [\lambda_0 \lambda_1 \lambda_2] \), then has \( a_1 = a_4 = a_7 \), \( a_2 = a_5 = a_8 \), and \( a_3 = a_6 = a_9 \), leading to the grouping of \{1, 4, 7\}, \{2, 5, 8\}, and \{3, 6, 9\}. Results similar to (5.8) and (5.9) also follow for (5.11).

In both of these weakly coupled cases, synchrony chooses the slow chord and the results are the same as for slow coherency. We would expect this since slow-coherency theory comes from the weakly coupled case seen in Chapter 2. The most interesting case is when the the line susceptances between the tiers are of the same order as those within a tier. If we let \( b_{12} = b_{13} = b_{23} = 10 \), \( b_{14} = b_{25} = b_{36} = 7.5 \) and \( b_{47} = b_{58} = b_{69} = 15 \) (note that only the ratio of the \( b_{ij} \)'s matter, not the actual values), our eigenvalues are

\[
\lambda = \begin{bmatrix}
\end{bmatrix}
\]

with the corresponding right eigenvector matrix

\[
U = \begin{bmatrix}
0.0727 & -0.1565 & 0.6039 & 0.2236 & 0.1220 & -0.3324 & 0.3342 & -0.4553 & 0.3333 \\
-0.1719 & 0.0152 & -0.1083 & -0.6348 & 0.1220 & -0.1232 & -0.4550 & -0.4553 & 0.3333 \\
0.0991 & 0.1412 & -0.4956 & 0.4112 & 0.1220 & 0.4557 & 0.1208 & -0.4553 & 0.3333 \\
-0.2715 & 0.5839 & -0.1618 & -0.0599 & -0.4553 & -0.3324 & 0.3342 & 0.1220 & 0.3333 \\
0.6414 & -0.0568 & 0.0290 & 0.1701 & -0.4553 & -0.1232 & -0.4550 & 0.1220 & 0.3333 \\
-0.3699 & -0.5271 & 0.1328 & -0.1102 & -0.4553 & 0.4557 & 0.1208 & 0.1208 & 0.3333 \\
0.1988 & -0.4275 & -0.4421 & -0.1637 & 0.3333 & -0.3324 & 0.3342 & 0.3333 & 0.3333 \\
-0.4696 & 0.0416 & 0.0793 & 0.4647 & 0.3333 & -0.1232 & -0.4550 & 0.3333 & 0.3333 \\
0.2708 & 0.3859 & 0.3628 & -0.3010 & 0.3333 & 0.4557 & 0.1208 & 0.3333 & 0.3333 \\
\end{bmatrix}.
\]

Note that \( u_2 \) in (5.7) has shifted to \( u_4 \) in (5.13), i.e., \( u_2 \) in (5.7) has the same mode shape as \( u_4 \) in (5.13). The slow chord no longer has coherent generators, making grouping according to slow
coherency more difficult. The SME algorithm, however, chooses $\mathcal{V} = [\lambda_0 \quad \lambda_1 \quad \lambda_4]$, preserving the grouping we had in the first case with (5.7). SME will not switch the partitioning until the mode shapes corresponding to $u_1$ and $u_2$ in (5.7) both correspond to faster modes, as in (5.11); then, it will go straight to the grouping seen for (5.11), both of which are reasonable partitionings.
This thesis focused on the development of synchrony theory and its application to Synchronic Modal Equivalencing (SME). The majority of the initial work in this area was done by Ramaswamy [14, 15, 16, 17]. We have explored in more detail the basic notion of synchrony, as well as extended the SME partitioning algorithm previously presented in [14]. The remainder of this chapter gives a summary of the thesis and suggestions for future work in this area.

6.1 Summary

In Chapter 2, we presented the background material necessary to understand the work done in the thesis. This includes the theory behind both slow coherency and synchrony.

Chapter 3 focused on the extension of the SME partitioning algorithm to include the grouping of load buses in the network. By using generalized eigenanalysis, the modes of the chordal matrix retained their load bus components for partitioning. The partitioning algorithm was then tested on two power system models: the New England model and the France-Spain model. Partitioning of load buses has implications for both study-area selection and fault simulation.

In the next chapter, we looked at the sensitivity of the eigenvalues and eigenvectors to line parameter perturbations. Chapter 4 is important to both the concepts of slow coherency and synchrony. In the chapter, we saw that the perturbations of both the eigenvalues and their corresponding eigenvectors were of the same order as the perturbations to the line parameters. Using two simple three bus examples, we also showed the validity of these derivations for small perturbations for systems with both generators and loads.

We looked at the relationship between slow coherency and synchrony in Chapter 5 by looking at a simple power system model. By weakly coupling decoupled areas, we see the evolution of the eigenvectors. As the couplings become stronger, we see how the modal patterns of the system changed and how this could affect our chord selection and partitioning.
6.2 Future Work

In this thesis, we have extended the Synchronic Modal Equivalencing algorithm to include load buses. There are several areas which still need to be investigated within this framework.

6.2.1 Simulations

The linear and non-linear simulation of reduced order power systems which have been equivalenced using the extended SME algorithm need to be performed. This could be done with packages such as MATLAB [12] and EUROSTAG [7]. Of special interest is the simulation of faults at load buses and faults on lines connecting areas with "fuzzy boundaries" mentioned in Chapter 3. The reduced-order simulations should be compared to the full model simulations for accuracy and necessary simulation time.

6.2.2 Weighting Factors

The SME procedure described in [14] uses participation factor-based weighting of the eigenvectors for chord selection. In this thesis, we use equal weighting of the eigenvectors. The notion of participation factors has not been developed for generalized eigenanalysis. Work in this area may give better insight into the weighting of the eigenvectors in the generalized eigenanalysis case we use for load bus partitioning.
Appendix A

Basis Selection

A.1 Background

Due to size and complexity of high-order power systems, model reduction is necessary to make the problem approachable and the results understandable. A lot of work has been done in this area using the theory of Synchronic Modal Equivalencing (SME). The details of this theory are readily seen in [14].

When working within the SME framework, the choice of basis vectors in terms of which the system external to the study area is represented is extremely important to the ability of the reduced-order system to correctly model the full system. Referring to Chapter 6 of [14], it is necessary to choose a matrix $V_b$, whose rows are the basis vectors, from the rows of a matrix $V \in \mathbb{R}^{m \times n}$, whose columns are a selected set of eigenvectors. $V$ is the chordal matrix, denoted $U_Y$ in Chapter 2 of this thesis. Let $V_z$ denote the submatrix of $V$ that remains when the chosen basis vectors are removed. The rows of $V_z$ are now represented as linear combinations of the basis vectors. In other words,

$$V_z = K_b V_b \quad \text{(A.1)}$$

or, equivalently,

$$K_b = V_z V_b^{-1} \quad \text{(A.2)}$$

Castrillón Candás [1] and Ramswamy [14] present different methods for choosing the basis $V_b$. We would like an algorithm which gives us a good basis as well as uses a reduced amount of computing time.

The matrices used here were taken from a model of the France-Spain power system used in Chapter 3. The modes were chosen according to [14]. Two different cases were used — a four- and a five-mode case. In the two different cases, the matrix $V$ had dimensions of $23 \times 4$ and $23 \times 5$ respectively.
A.2 The Norm Idea

Equation A.2 came from our assumption that $V_z$ can be written entirely as a linear combination of the vectors of $V_b$. The matrix $K_b$ turns out to function as a correction factor to a baseline dynamic equivalent. Suppose we want to minimize the magnitude of this correction factor, *i.e.*, minimize the norm of $K_b$. Results in [1] suggest that this may be a reasonable objective. We will later address the issue of which norm is appropriate and whether it makes a difference which norm we use.

Before we run our first trials, it is interesting to note, for a general matrix, the relative computation times required for different norms. Our choices for norm selection are:

- the 1-norm,
- the 2-norm,
- the $\infty$-norm, and
- the Frobenius-norm (for convenience we will use the notation “F-norm”).

We find that using the 1-norm takes the least amount of time.\(^1\) The $\infty$-norm does not take too much longer. The F-norm takes about three times as long and the 2-norm takes over twice the amount of time as the F-norm.

A.2.1 Full Combinatorial Search

The first idea for choosing $V_b$ is to try every possible basis and choose the one which gives the minimum norm of $K_b$. A simple combinatorial search of all possible bases is not difficult to program, but may take too much time to run.

A.2.2 QR decomposition

Since checking the norm of every possible basis takes a lot of time, it makes sense to look at alternate methods of finding a good basis, still with the idea of minimizing the norm of $K_b$. One method is to find a reasonable preliminary basis and then try altering that basis to see if better results can be achieved. Referring to (A.2), it makes sense to choose a basis where the norm of $V_b$

---

\(^1\)All computations were done using Matlab [12] on a Sun workstation. Note that though the actual times will vary from computer to computer and depend on the order of the system, the ranking of the times should not.
A.2 The Norm Idea

is large. In this way, we would hopefully keep the norm of $K_b$ small. This objective suggests the following strategy.

For the first row of $V_b$, we choose the row of $V$ with the largest norm. Using a matrix $P_1$, we permute this row to the top of $V$. Then, using an orthogonal rotation matrix $Q_1^T$, we force the first row of $P_1 V$ have the form $\begin{bmatrix} * & 0 & \cdots & 0 \end{bmatrix}$. We define the matrix $V'$ to be the remaining rows of the rotated matrix with the first column removed. The result is

$$Q_1 = P_1 V \quad \text{(A.3)}$$

We now consider $V'$. We choose the row of $V'$ which has the largest norm, hence the row of $P_1 V$ that has the largest orthogonal distance from the first basis vector. This row will correspond to the second basis vector. As we did above, we permute this row to the top of $V'$ and perform an orthogonal transformation so that it has the form $\begin{bmatrix} * & 0 & \cdots & 0 \end{bmatrix}$. The result is

$$Q_1 Q_2 = P_2 P_1 V \quad \text{(A.4)}$$

We repeat this algorithm $n$ times, until we have as many basis vectors as there are selected modes, i.e., the number of columns in $V$. The final result is of the form

$$Q_1 \cdots Q_n = P_n \cdots P_1 V \quad \text{(A.5)}$$

or, equivalently,

$$R' Q = P V \quad \text{(A.6)}$$

The procedure we have just outlined is exactly equivalent to the QR decomposition function found in Matlab [12]. Matlab will return the permutation matrix, $P$, from which we may determine the rows corresponding to our preliminary basis.

Once we select a preliminary basis using QR decomposition, we use two different combinatorial approaches to improve upon the initial guess. In the first method, we substitute one of
the remaining rows of $\mathbf{V}$ in for the first row of $\mathbf{V}_b$. The new norm of $\mathbf{K}_b$ is calculated. This is repeated by substituting each unused row of $\mathbf{V}$ into the first row of $\mathbf{V}_b$. The same substitution is subsequently done with the other rows of $\mathbf{V}_b$. If the newly calculated norm of $\mathbf{K}_b$ is smaller than the previous norm, we keep this changed basis. If it is not, we continue to use our original basis. If there is a change in basis, we begin the iteration process over again by substituting in a new first row.

If our preliminary basis is bad, it is possible that this process may take nearly as much time as the full combinatorial search. It is also possible that, by allowing only one row to change at each iteration, we may arrive at a basis which does not give the lowest possible norm of $\mathbf{K}_b$. Our hope is that the initial choice will be good enough to reduce the computing time. We also hope the basis selected, while possibly not optimal in the sense of minimizing the norm of $\mathbf{K}_b$, will still be satisfactory. Thus, we may sacrifice a little in terms of minimizing the norm of $\mathbf{K}_b$, but we will save a lot of computing time.

In our second method we hope to achieve the same goal but decrease the computing time further. We do a similar combinatorial search, but this time suggest that all but one of the initial row vectors is correct. Our final answer, therefore, will have at most one changed vector. In our search, we change only one row vector. If we find a basis which results in a lower norm we will keep that basis, but we will never start over. We assume that our initial guess was fairly good and that with only minimal changes we can achieve a satisfactory basis.

A.2.3 Different Norms

Thus far, we have assumed that our choice of norm does not make a difference in the basis selection. This was an invalid assumption; using different norms does result in different bases. The basis which results is dependent upon the norm we use. As we will see in A.3, in all cases some of the basis vectors were the same but many times one of them changed depending upon the norm used.

A.3 Results

A.3.1 Basis results

Tables A.1 and A.2 depict the different results for basis selection. It is important to note in Table A.1 that different norms can not be compared with any meaning, i.e., the magnitude of the 1-norm should not be compared to the magnitude of the 2-norm.
A.3 Results

<table>
<thead>
<tr>
<th>Norm</th>
<th>Basis</th>
<th>Norm of $K_b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 modes:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>12 14 15 18 20</td>
<td>7.4715</td>
</tr>
<tr>
<td>2</td>
<td>10 14 15 18 20</td>
<td>2.4709</td>
</tr>
<tr>
<td>$\infty$</td>
<td>13 14 15 18 20</td>
<td>1.3641</td>
</tr>
<tr>
<td>Fro</td>
<td>13 14 15 18 20</td>
<td>3.3863</td>
</tr>
<tr>
<td>Pure QR</td>
<td>13 14 15 18 20</td>
<td></td>
</tr>
<tr>
<td>Ramas</td>
<td>13 14 18 19 21</td>
<td></td>
</tr>
<tr>
<td>4 modes:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>10 15 18 20</td>
<td>7.5996</td>
</tr>
<tr>
<td>2</td>
<td>10 15 18 20</td>
<td>2.4795</td>
</tr>
<tr>
<td>$\infty$</td>
<td>12 15 18 20</td>
<td>1.2607</td>
</tr>
<tr>
<td>Fro</td>
<td>13 15 18 20</td>
<td>3.5211</td>
</tr>
<tr>
<td>Pure QR</td>
<td>13 15 18 20</td>
<td></td>
</tr>
<tr>
<td>Ramas</td>
<td>13 18 19 21</td>
<td></td>
</tr>
</tbody>
</table>

Table A.1: **Basis selection using different norms.** Norm indicates the norm used in doing a combinatorial search (1-norm, 2-norm, $\infty$-norm, Frobenius norm). Pure QR and Ramas denote the use of the QR decomposition without further searching and the algorithm in [14], respectively. Basis indicates the row vectors chosen for $V_b$.

There are several important items to note. First, as we can see from Table A.2, the norms of $K_b$ from the different algorithms do not vary greatly. For comparison, it is easy to choose a basis which gives a $K_b$-norm greater than $4 \times 10^5$ for any norm type. This implies that any of the bases may work just as well. It is also interesting to see that choosing the basis using the method outlined in [1] gave the same basis as the Pure QR.

The similarity in basis selection seen in Table A.1, regardless of which norm or algorithm is used, (basis vectors such as \{18\} show up consistently) implies the importance of certain machines in the system. Other machines may not be as important, being represented equally as well by the choice of a different vector (machines \{13\} and \{10\} appear to be very closely related in the 4-mode case).

The similarity of the bases found by minimizing the norm of $K_b$ to the results in [1] and [14] also shows some validity to our reasoning. It may be very difficult to determine which of these bases represents a "better" model-reduction. A combinatorial search to minimize the norm may take too long for extremely high-order systems, but using the QR decomposition as an initial guess, or possibly even as the final basis choice, enables a good basis to be found efficiently.
Basis Selection

<table>
<thead>
<tr>
<th></th>
<th>Ramas</th>
<th>Pure QR</th>
<th>Comb</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 modes:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-norm</td>
<td>7.8279</td>
<td>7.4830</td>
<td>7.4715</td>
</tr>
<tr>
<td>2-norm</td>
<td>2.5329</td>
<td>2.4783</td>
<td>2.4709</td>
</tr>
<tr>
<td>∞-norm</td>
<td>1.4871</td>
<td>1.3641</td>
<td>1.3641</td>
</tr>
<tr>
<td>Fro-norm</td>
<td>3.5839</td>
<td>3.3863</td>
<td>3.3863</td>
</tr>
<tr>
<td>4 modes:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-norm</td>
<td>7.9166</td>
<td>7.6158</td>
<td>7.5996</td>
</tr>
<tr>
<td>2-norm</td>
<td>2.5585</td>
<td>2.5208</td>
<td>2.4795</td>
</tr>
<tr>
<td>∞-norm</td>
<td>1.3713</td>
<td>1.2679</td>
<td>1.2607</td>
</tr>
<tr>
<td>Fro-norm</td>
<td>3.6944</td>
<td>3.5211</td>
<td>3.5211</td>
</tr>
</tbody>
</table>

Table A.2: Norms of different basis selections. This table shows the norms of the different bases found using three different methods: Ramaswamy, pure QR, and combinatorial.

A.4 Matlab Coding

All of the programming was done in Matlab 4.2 [12] on a SUN workstation. The following Matlab files were written:

- `kmincomb.m`: original trial program (3 modes)
- `kminfind.m`: original with time elapsed
- `kmincom5.m`: combinatorially finds basis minimizing norm(\(K\))
- `kminqr5.m`: do QR decomp then go through combinatorially
- `kminqr5h.m`: do QR then hold \((p - 1)\) modes
- `kminqrs5.m`: do QR then go through combinatorially, select the norm
- `kqr5.m`: just do QR
- `normfind.m`: finds norm of given basis
- `kmncom5.m`: selects \((p - 1)\) modes without study area
- `allbasis.m`: shell program for running all others
The derivations in Appendix B are drawn, with only slight modification, from [9].

B.1 Eigenvalue Sensitivity

We begin with the generalized eigenvalue/eigenvector equation (3.8):

\[(\lambda_i M + K)u_i = 0, \quad u_i \neq 0.\]  
(B.1)

The corresponding equation for the generalized left eigenvector corresponding to \(\lambda_i\) is

\[w_i^T(\lambda_i M + K) = 0.\]  
(B.2)

The derivation for the change in \(\lambda_i\) with respect to the change in \(K\) and \(M\) follows from taking a first-order difference of the equation \(w_i^T(\lambda_i M + K)u_i = 0:\)

\[
\Delta w_i^T(\lambda_i M + K)u_i + w_i^T(\lambda_i M + K)\Delta u_i + w_i^T(\Delta\lambda_i M + \lambda_i \Delta M + \Delta K)u_i = 0
\]  
(B.3)

\[
\Delta \lambda_i(w_i^T M u_i) = -w_i^T(\Delta K + \lambda_i \Delta M)u_i
\]  
(B.4)

\[
\Delta \lambda_i = \frac{-w_i^T(\Delta K + \lambda_i \Delta M)u_i}{w_i^T M u_i}
\]  
(B.5)
Sensitivity Derivations

B.2 Eigenvector Sensitivity

This derivation is for symmetric $K$ and $M$, where $w_i = u_i$. We again begin with the generalized eigenvalue/eigenvector equation (3.8):

$$(\lambda_i M + K)u_i = 0, \quad u_i \neq 0. \quad (B.6)$$

For the remainder of the derivation we will drop the subscript $i$ for brevity. Finding the change in (B.6) with respect to line parameter changes, we get

$$\Delta(K + \lambda M)u + (K + \lambda M)\Delta u = 0, \quad (B.7)$$

or

$$\Delta Fu + F\Delta u = 0, \quad (B.8)$$

where $F = K + \lambda M$. This may also be written

$$F\Delta u = -\Delta Fu. \quad (B.9)$$

Since $w_i = u_i$ for symmetric matrices, we know that

$$\lambda_i M u_i = -K u_i \quad (B.10a)$$

$$\lambda_i u_i^T M = -u_i^T K \quad (B.10b)$$

Multiplying (B.10b) by a right eigenvector, we get

$$\lambda_i u_i^T M u_j = -u_i^T K u_j. \quad (B.11)$$

Then, using (B.10a), we get

$$\lambda_i u_i^T M u_j = \lambda_j u_i^T M u_j \quad (B.12)$$

For distinct eigenvalues $\lambda_i$ and $\lambda_j$, (B.12) only holds if $u_i^T M u_j = 0$, i.e., if the eigenvectors are orthogonal with respect to $M$. If we scale the eigenvectors to be orthonormal, then

$$u_i^T M u_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}. \quad (B.13)$$
In the case where we have repeated eigenvalues, the eigenvectors corresponding to the repeated eigenvalues can still be picked to be orthogonal in the sense of (B.13) [18]. Then, finding the change in (B.13), we see that

$$\Delta u^T M u + u^T \Delta M u + u^T M \Delta u = 0 .$$  \hspace{1cm} (B.14)

Since $M$ is a symmetric matrix, $\Delta u^T M u = u^T M \Delta u$ and (B.14) may be equivalently written

$$2u^T M \Delta u = -u^T \Delta M u .$$  \hspace{1cm} (B.15)

Augmenting (B.9) with (B.15), we get the following matrix equation:

$$\begin{bmatrix} F \\ 2u^T M \end{bmatrix} \Delta u = - \begin{bmatrix} \Delta F \\ u^T \Delta M \end{bmatrix} u .$$  \hspace{1cm} (B.16)

Premultiplying (B.16) by $\begin{bmatrix} F \\ u^T M \end{bmatrix}^T$, we have

$$\begin{bmatrix} F^T \\ 2u^T M \end{bmatrix} \Delta u = - \begin{bmatrix} F^T \\ u^T \Delta M \end{bmatrix} \begin{bmatrix} \Delta F \\ u^T \Delta M \end{bmatrix} u .$$  \hspace{1cm} (B.17)

Recalling that $M$ and $F$ are both symmetric and performing the matrix multiplication, we get the matrix equation

$$[F^2 + 2Mu u^T M] \Delta u = -[F \Delta F + Mu u^T \Delta M] u .$$  \hspace{1cm} (B.18)

Assuming the left-hand term to be invertible and substituting in for $F$, we get our final equation for $\Delta u$:

$$\Delta u = -[(K + \lambda M)^2 + 2Muu^T M]^{-1}[(K + \lambda M) \Delta (K + \lambda M) + Mu u^T \Delta M] u .$$  \hspace{1cm} (B.19)

This is the form of the equation used in (4.5).
Appendix C

Matlab Load/Gen Partitioning Program

All the code in this chapter was written for Matlab by the author of this thesis, with the exception of INCIDENCE.m and SETUPSYSTEM.m which were written by DeMarco and Wassner [6]. The function srp2.m is a modification of srp.m written by Ramaswamy [14].

```matlab
function [References, GenGroupings, LoadGroupings] = srplgl()

% Syntax: [References, GenGroupings, LoadGroupings] = srplgl()
% This is the main module of the Synchrony Recognition Program which
% will partition a high order power system according to the method of
% synchrony. The major modification over 'srpl' written by Ganesh
% Ramaswamy is the grouping of load buses.
% This program requires:
% 1. All files necessary to run srpl.m (written by G.N. Ramaswamy)
% 2. Three data files: initangle.dat, gendata.dat, linedata.dat
%   For info on these files see the end of this program.
% 3. The following .m files: srp2, INCIDENCE, SETUPSYSTEM,
%    frsper2, frspset2, groups, vangle, voltangle
% Written by: Jesko M. Hagee
% Last revised: October 2, 1996.

disp(' '); disp(' '); disp(' '); disp(' '); disp(' ');
disp(' '); disp(' '); disp(' '); disp(' '); disp(' ');
disp(' '); disp(' '); disp(' '); disp(' '); disp(' ');
disp(' '); disp(' '); disp(' '); disp(' '); disp(' ');
disp(' '); disp(' '); disp(' '); disp(' '); disp(' ');

%% Set up model in same way as needed for Chris DeMarco's process.
%% Results in two matrices: FSNwd (Network data for France/Spain),
%% FSInrt (Generator inertias for France/Spain).
```

---

(Additional code segments follow, but they are not transcribed here.)
Matlab Load/Gen Partitioning Program

[FSNwd,FSInrt]=frspset2;

%% Form E,R matrices for generalized eig. problem

disp(''); disp(' Setting up network matrices ...');
[E,R]=frsper2(FSNwd,FSInrt);

nbus = size(R,1);
gen = rank(E);
nload = nbus-ngen;
vectnum = [nbus ngen nload];

%%
%% Partition E,R matrices into load/generator components
%%

Egg = E(1:ngen,1:ngen);
Rgg = R(1:ngen,1:ngen);
Rgl = R(1:ngen,ngen+1:nbus);
Rlg = R(ngen+1:nbus,1:ngen);
Rll = R(ngen+1:nbus,ngen+1:nbus);

if Rll==[]
    Rreduced = Rgg;
else
    Rreduced = Rgg-Rgl*inv(Rll)*Rlg;
end

Aganesh = inv(Egg)*Rreduced;

modes = srp2(Aganesh);

[q_high,ChordSize] = size(modes);
OldModes = modes(q_high,:);
[Vred, Dred] = myeig(Aganesh);
VReg=Vred;
RegEigVal=Dred;

%% Solve generalized eig. problem using Matlab's qz algorithm. We
%% will get generalized eigenvalues and a matrix of corresponding
%% generalized eigenvectors.

[AA,BB,Q,Z,VGen]=qz(R,E);
GenEigVal=diag(AA)./diag(BB);
WGen=VGen;

%% Find the modes of the generalized problem which correspond to the
%% selected modes of the regular eigenvector problem.

disp(''); disp(' Partitioning Load and Generator Buses ...');
[NewModes]=vangle(VReg,VGen(1:ngen,nload+1:nbus),OldModes);
NewModes=NewModes+nload*ones(1,ChordSize);

NewChordV=VGen(:,NewModes);
NewChordW=WGen(:,NewModes);

%% Select reference machines

disp(' ')
Weight=ones(nbus,ChordSize);
References=selectref(ChordSize,NewChordV(1:ngen,:),Weight(1:ngen,:));

%% Find degree of synchrony between rows to separate buses into
%% areas.
Weight2=ones(nbus,ChordSize);
[Mu,Kg]=syncdeg(NewChordV(References,:),NewChordV,Weight2)

[GenGroupings,LoadGroupings]=groups(Kg,vectnum);
%% [References,GenGroupings] = machconv(References,GenGroupings);
References;
GenGroupings;
LoadGroupings;

% DATA FILE SETUP

% The program was written such that all data needed can be taken from
% an output file from EUROSTAG.
%
% Requires 3 data files setup as follows:
%
% 1. linedata.dat -- text file
% Text file with four columns and number of rows equal to
% the number of lines in the system. In each row, the first two
% columns give the buses connected by the line; the last two
% columns, the resistance and reactance of the line.
%
% 2. gendata.dat -- text file
% Text file with three columns and number of rows equal to the
% number of generators. In each row, the first column gives
% the bus number for the generator; the second, the apparent
% rating; the third, the Constant of Inertia.
%
% 3. initangle.dat -- text file
% Text file with three columns and number of rows equal to the
% number of buses. In each row, the first column gives the bus
% number; the last two, the real and imaginary parts of the
% initial value of the voltage variables.
function modes = srp2(A)

% Syntax: modes = srp2(A)
%
% This is a modified version of srpl.m for use with the srplg.m routine.
% Modified by: Jesko M. Hagee
%
% This is the main module of the (simplified) Synchrony Recognition
% Program which also provides a convenient user interface. The explicit
% input is the A matrix from second-order undamped swing equations, but
% the user should initialize the global variables names, title1 and
% title2 to provide the necessary information to write the output file
% (see machname.m for an example). The user will be prompted for all
% other information, including the filename to write the output. The
% output is written in a format compatible with SMAS3. In addition to
% the output file, the program provides the explicit output modes, which
% contains the modes used for the system partitioning.
%
% The primary subroutines used are:
% selectref (selects reference machines using a simplified algorithm)
% selectchord (selects the chord)
% syncdeg (computes the degree of synchrony)
% Other subroutines are: myeig, dopl, permute, no, split.

% Written by: Ganesh N. Ramaswamy
% Last revised: December 18, 1994.

% disp(' '); disp(' '); disp(' '); disp(' '); disp(' ');
% disp(' '); disp(' '); disp(' '); disp(' '); disp(' ');
% disp(' MASSACHUSETTS INSTITUTE OF TECHNOLOGY');
% disp(' ');
% disp(' SYNCHRONY RECOGNITION PROGRAM');
% disp(' Version: December 18, 1994');
% disp(' Written by: Ganesh N. Ramaswamy');

disp(' '); disp(' Running Synchrony Recognition Program 2 (written by G.N. Ramaswamy)');
disp(' ');
disp(' Initializing ... ');
disp(' ');
q_low = input(' Enter the lower limit on the number of areas: ');
q_high = input(' Enter the final number of areas for partitioning: ');
disp(' ');
disp(' How do you wish to build the chord for aggregation?');
disp(' Sequential inclusion of modes (slow to fast) [Enter 1]');

disp(' Intelligent selection (default) [Enter 0]');
flag1 = input(' Enter 0 or 1 and press RETURN: ');
if flag1 == []
    flag1 = 0;
end
disp(' How do you wish the initial modes to be selected?');
disp(' User input for the q_low most extensive modes [Enter 2]');
disp(' Use the q_low slowest modes (default) [Enter 1]');
disp(' Use the q_low slowest (non-zero) modes [Enter 0]');
flag2 = input(' Enter 0, 1, or 2 and press RETURN: ');
disp(' ');
disp(' Performing Eigenanalysis ...');
[V, D] = myeig(A);
W = inv(V)';
unless, n = size(A);
if flag2 == []
    flag2 = 1;
end
if flag2 == 2
    p = input(' Please input indices for user selected modes: ');
else
    p = [(2-flag2):(q_low+1-flag2)];
end

% outfile = input(' Please enter name of output file (in quotes): ');
% if outfile == []
%     outfile = 'srplout';
%     disp(' Output will be written to file srpl-out');
% end
% fptr = fopen(outfile, 'w+');
% disp(' ');
%
% global title1 title2 titles;
% fprintf(fptr, '%s
', title1);
% fprintf(fptr, '%s
', title2);
% disp(' Program running ...');
modes = zeros((q_high-q_low+1), q_high);
disp(' ');

iter = 1;
for q = q_low:q_high
    modes(iter, 1:q) = p;
    iter = iter+1;
end
Va = V(:, p);
Wa = W(:, p);
b = selectref(q, Va, Wa);
[Mu, K] = syncdeg(Va(b,:), Va, Wa);
    [t1, t2] = max(abs(K'));
col1 = t2';
col2 = zeros(n, 1);
Matlab Load/Gen Partitioning Program

col2(b) = ones(q, 1);
col3 = sum(K');
col4 = [1:n]';
result = [col1 col2 col3 col4];
% fprintf(fptr, '%s %7.0f\n', ' NUMBER OF AREAS:', q);
% fprintf(fptr, '%s\n', ' AREAS');
% for k=l:n
% fprintf(fptr, '%s %22.0f %4.0f %14.5f %14.0f
', names(k,:), ... 
% result(k,1), result(k, 2), result(k, 3), result(k,4));
% end
% fprintf(fptr, '%5.0f\n', -999);
disp(' *');
if flag1 == 1
p = [p (max(p)+1)];
end
if flag1 ~= 1
b = selectref(q+1, Va, Wa);
[Mu, K] = syncdeg(Va(b,:), Va, Wa);
K = K(dopl(b, n), :);
Kg = K((q+2):n, :);
An = permute(A, dopl(b, n));
Vn = V(dopl(b,n), :);
Ar = An(1:(q+1),1:(q+1))+An(1:(q+1),(q+2):n)*Kg;
p = selectchord(p, Ar, D, Vn, 1:(q+1))
end
end
end

function A = INCIDENCE (Nwd)
% Written by Chris DeMarco

% SETUP CONSTANTS

nbus = max(max(Nwd(:,1:2)));  % # of buses
nbra = size(Nwd,1);  % # of branches

% MAIN PROGRAM STARTS

%=====================================================================

--- Output ---
A(1,1) = nbus; % # of buses = # of rows in A
A(1,2) = nbra; % # of branches = # of columns in A
A(1,3) = 0;

A(2:nbra+1,1) = Nwd(:,1); % source node of branches := 1
A(2:nbra+1,2) = [1:nbra]';
A(2:nbra+1,3) = ones(nbra,1);

A(nbra+2:2*nbra+1,1) = Nwd(:,2); % destination node of branches := -1
A(nbra+2:2*nbra+1,2) = [1:nbra]';
A(nbra+2:2*nbra+1,3) = -ones(nbra,1);

A = spconvert(A); % convert to Matlab sparse format

%%% COMMENTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

The function INCIDENCE is invoked from the function SETUPSYSTEM and composes
the incidence matrix A according to the network description Nwd. Matrix A
has as many rows and columns as the network has buses and branches, respec-
tively. Since there is only a relatively small number of non-zero entries in
A, it will be returned in the Matlab sparse format.

ARGUMENTS : Nwd - network description matrix (see function SEPARATION)

RETURNS : A - Incidence matrix of the network described by Nwd

In the routine two constants are needed several times. The first is the number
of branches nbra, obtained from the number of rows in Nwd, i.e. the number
of given branch data. The second constant, the number of buses nbus, is equal
to the largest bus number in the network description matrix Nwd.
In the main part of the routine matrix A is composed in the external sparse
format. For every non-zero element k in A, a row has to be created. This row
contains row and column indices of k and as the third entry the value of k
by itself (see Matlab command SPCONVERT). Since Matlab needs for a matrix,
given in this format, a row that contains information about the matrix
dimension, this requirement is fulfilled in the first three lines. Next,
assuming buses in the first column of Nwd as the source for each branch,
the row index, the column index, and a "one" for each branch is assigned.
According to the definition of the incidence matrix, the row index of the
"one" entry for branch k is given by the number of its source bus. The
column index for branch k is k by itself. The next 3 lines perform a similar
procedure creating the minus-one-entries in the incidence matrix for the
destination bus of each branch.
Finally, matrix A is converted to the internal Matlab sparse format.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Matlab Load/Gen Partitioning Program

function [A,D,E,b_ad] = SETUPSYSTEM (Nwd,Inrt)
% Written by Chris DeMarco

% SETUP CONSTANTS
nbus = max(max(Nwd(:,1:2)));  % total # of buses within the network
mgen = length(Inrt(:,1));    % # of generator buses within the network
div = (60*pi);               % constant for scaling the Inertias

% MAIN PROGRAM STARTS
A = INCIDENCE(Nwd);          % form incidence matrix

d_dg = ones(nbus,1);         % form diagonal for matrix D with the first..
d_dg(1:mgen) = Inrt(:,1)/div; % mgen elements equal to scaled inertias ...
d_dg = d_dg.^(-0.5);        % and apply one over square root
D = sparse(diag(d_dg));      % D is sparse diagonal matrix

e_dg = zeros(nbus,1);        % form diagonal for matrix E with the first..
e_dg(1:mgen) = ones(mgen,1); % mgen elements equal to one
E = sparse(diag(e_dg));      % E is sparse diagonal matrix
b_ad = Nwd(:,3);             % obtain vector of branch admittances

% COMMENTS
% This function is used by the main routine SEPARATION in order to form all the
% matrices necessary for modeling the system. These matrices will be returned
% in the Matlab sparse format. It is provided, that the buses have been numbered
% in a way, so that all generators appear first in the ordering of all buses.
% ARGUMENTS :  Nwd - network description matrix (see function SEPARATION)
%               Inrt - matrix of Inertias for generator buses
% RETURNS :    A - Incidence matrix of the network;
%               D - diagonal matrix for modification of A with normalized
%               inertias
%               E - diagonal matrix for generalized eigenvalue problem
%               b_ad - vector of branch admittances in the network
% First, setup three constants that will be used in the main part of the
% function. The buses in the network are supposed to be numbered from one to the
% total number of buses nbus. Therefore, this number can be obtained from the
% largest of all entries in the first two columns of the network description
% matrix Nwd. The number of generator buses within the network mgen is equal to
% the number of entries in the first column of the inertia matrix Inrt.
% The third constant div is used for scaling the inertia values by the frequency
% in radiant per second.
As the first step in the main part of the routine, the function INCIDENCE is invoked to create the incidence matrix A of the network. Then in a procedure of four steps the matrix D is formed. First, create a one-vector of dimension equal to the number of buses. Assign to the first elements in this vector the generator's inertia, scaled by constant div. Apply one over square root to this vector in order to fulfill the existing conditions for matrix D. Finally, with the resulting vector the diagonal matrix D is formed and stored in the sparse format. Due to the relatively small number of non-zero entries in D, this is advantageous for further computations with regard to time-consumption. In the next three steps, matrix E is created in a similar way. A zero-vector of dimension nbus is formed. Then the first components, referring to generator buses, are set to one. With this vector, E is created as diagonal matrix and converted to the sparse format. In a last step, the vector of branch admittances b_ad is extracted from the given network description matrix.

%====================================================================End of Module==============================%

function [E,R]=frsper2(Nwd,Inrt)

[Ad,Dd,Ed,b_ad]=SETUPSYSTEM(Nwd,Inrt); [magdelta]=voltangle;

Ybus=Ad*diag(imag(b_ad))*diag(exp(abs(Ad)'*log(magdelta(:,2))))*diag(cos(Ad'*magdelta(:,1)))*Ad'; R=Ybus;

nbus=size(R,1); ngen=size(Inrt,1); nload=nbus-ngen;

M=diag(Inrt(:,2)); E=[M zeros(ngen,nload); zeros(nload,nbus)];

%====================================================================End of Module==============================%

function [FSNwd,FSInrt]=frspset2()

load linedata.dat
load gendata.dat
FSInrtl=gendata; % Input from data file
FSNwdl=linedata; % Input from data file
clear linedata
clear gendata

%XXXXXXXXXXXXXXXXX SETUP MATRIX WITH LINE NUMBERS AND ADMITTANCES XXXXXXXXXXX

% Convert Resistance/Reactance to Impedance %
for count=1:size(FSNwdl,1)
FSNwd2(count,1:2)=FSNwd1(count,1:2);
FSNwd2(count,3)=1/(FSNwd1(count,3)+i*FSNwd1(count,4));
end
Matlab Load/Gen Partitioning Program

%%% Sort by Bus Number %%
[ya,inda]=sort(FSNwd2(:,2));
for i=1:size(inda)
  FSNwd3a(i,:)=FSNwd2(inda(i),:);
end
[y,ind]=sort(FSNwd3a(:,1));
for i=1:size(ind)
  FSNwd3(i,:)=FSNwd3a(ind(i),:);
end

FSNwd=FSNwd3;  %%%%%% USE THIS AS FINAL MATRIX (FSNwd)

XXXXXXXXXXXXXXXXXXXXXXXXXX SETUP GENERATOR INERTIA MATRIX %%%%%

for i=1:size(FSInrtl,i)
  FSInrt2(i,1)=FSInrtl(i,1);
  FSInrt2(i,2)=FSInrtl(i,3)*FSInrtl(i,2)/(pi*5000);
end

FSInrt=FSInrt2;  %%%%%% USE THIS AS FINAL MATRIX (FSInrt)

%===================================End of Module====================================%
function [newmode]=vangle(Vold,Vnew,oldmode)
for i=1:length(oldmode)
    Vtemp=Vold(:,oldmode(i));
    newmode(i)=1;
    angold=Vtemp'*Vnew(:,1)/(norm(Vtemp)*norm(Vnew(:,1)));
    for j=2:size(Vold,2)
        ang=Vtemp'*Vnew(:,j)/(norm(Vtemp)*norm(Vnew(:,j)));
        if abs(ang)>abs(angold)
            angold=ang;
            newmode(i)=j;
        end
    end
end

function [magdelta]=voltangle()
% JM Hagee 09Jul96
load initangle.dat
numbus=size(initangle,1);
magdelta=zeros(numbus,2);

%insort=sort(initangle(:,1));
%test=[1:83];
%for i=1:numbus;
%    if(insort(i)~=test(i))
%        disp('error')
%    end
%end

for i=1:numbus
    magdelta(initangle(i,1),1)=atan(initangle(i,3)/initangle(i,2));
    magdelta(initangle(i,1),2)=sqrt((initangle(i,3))^2+(initangle(i,2))^2);
end


