# Applications of Phasor Measurements to the Real-time Monitoring of a Power System

by

David Edward Barber

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Approved:

Dr. A. Phadke, Co-Chairman

Dr. L. Mili, Co-Chairman

Dr. J. De La Ree Lopez

Dr. H. VanLandingham

March 1994

Blacksburg, Virginia

c.2

LD
5655
V855
1994
B373
c.2

# Applications of Phasor Measurements to the Real-time Monitoring of a Power System

by

David Edward Barber

Committee Chairmen: Dr. A. Phadke and Dr. L. Mili

Bradley Department of Electrical Engineering

**(ABSTRACT)**

This thesis discusses applications of phasor measurement units to power system monitoring and synchronous generator modeling. Adjustments to a previously developed PMU placement algorithm are described which observe generator and tie line flows explicitly and reduces the number of PMUs required for a system, still observing the major dynamic components of a system. This adjusted methodology leaves some buses unobserved. A method for estimating the state of the unobserved region is developed based on using constant admittance or constant current load models. These models are accurate for a small neighborhood around the operating point when they were calculated. To determine the maximum error expected for any given system estimate, an equation relating the maximum error in the voltages to the maximum change in load power is derived. Once the issue of power system monitoring has been presented, the application of PMUs to the synchronous generator modeling is explored. This thesis deals with the on-line identification of the generator transient model using a recursive version of the generalized least squares algorithm. Simulations have been performed to demonstrate the validity and difficulties with these methods.

# ACKNOWLEDGEMENTS

I wish to acknowledge and give thanks to all the following: to my wife, Monica, for her support through the entire process; to my parents, George and Sarah Barber, for their support both emotionally and financially; to my employer, General Public Utilities, for giving me the time to write, distribute, and defend this thesis; to Dr. Arun Phadke and Dr. Lamine Mili for their patient guidance and support; to Dr. De La Ree for providing the initial spark of interest into the field of power engineering; to Dr. Vanlandingham for providing information on system identification; and finally, to all four of the aforementioned professors for participating as part of my graduate committee.

# Table of Contents

# List of Figures

# List of Tables

# CHAPTER 1

# INTRODUCTION

Although the power demands upon a power system continue to grow, it has become increasingly difficult to add generation or high voltage transmission to the system. Utilities must find ways, if possible, to meet the demand with their current network. By using less conservative limits on branch flows and spinning reserve, a company would be able to supply more load with the present system at a lower level of security. The security of this operating approach would be improved with faster and more accurate monitoring and control.

The measurements collected through a SCADA system are designed to capture only quasi-steady state operating conditions, which precludes any adequate monitoring and control of the system during transient conditions. This deficiency is being alleviated thanks to the advent of the phasor measurement unit (PMUs). Indeed, these units are endowed with a fast sampling rate, fast enough to track the modes of oscillations that originate from a major disturbance occurring in the system[1,2]. In addition to the fast sampling rate, the PMUs provide synchronized measurements that help to eliminate the problems of time-skew. The PMUs are also useful since they measure the magnitudes and the phase angles of the voltages and currents, and the frequency at a bus. Several possible applications for PMUs are suggested in [1,2]. This thesis addresses two applications of PMUs. One is the monitoring of a system's dynamic state during transients. The other is the identification of the transient synchronous generator model.

A prerequisite to system monitoring and control is the adequate placement of the PMUs in the system. A scheme for PMU placement was proposed in [3]. It looks for the minimal set of PMUs that makes the system observable. It is found that typically 25% to 30% of the buses need to be provided with PMUs to estimate the whole state of the system. Two drawbacks of this scheme can be foreseen. First, many electrical utilities will not install the required number of PMUs in their systems due to the high cost of a PMU. Second, the method lumps generation with loads at the generator buses, which precludes the modeling of the synchronous generators.

The paper proposes another approach that further reduces the number of PMUs needed to capture the transients. It stems from the recognition that only a small fraction of the system components play a decisive role in the making of the system dynamics. These are the large generators, the tie-lines, the extra-high voltage sub-network, and the major dynamic loads. These components form a sub-system that will be made observable through PMUs. For the sake of clarity, it will be termed the *internal system.* The remaining sub-network, which is not observed by PMUs, will be termed the *external system.* It contains only buses with small generators and quasi-static or static loads. In the sequel, all the loads are assumed to be of the latter type and hence are attached to the external system. The approach has been applied to a test 640-bus system taken from an actual utility's bulk transmission system. The internal system consists of 170 buses forming 42 disconnected regions, the largest one including a 32-bus 500 kV sub-network and 29 including only a single bus. It turns out that this system can be observed by 77 PMUs, when PMUs are forced on the terminal buses of generators with a capacity of 400MW or higher. This represents 45% of the total number of buses of the internal system.

To determine the stability margins, the knowledge of the dynamic state of both the internal and the external systems is required during the transients. The internal system being observed by PMUs, its state can be readily estimated through a state estimator. The latter processes in a tracking mode the successive snapshots provided by the PMUs. A more challenging task is the calculation of the state of the external system. Methods based on load flow calculations or state estimation were proposed in [4,5,6,7]. In this paper, another approach is investigated. It assumes that the state of the external system is observed by a redundant collection of SCADA-based measurements and thereby is estimated cyclically by a robust static state estimator. Although these estimates are updated at a rate that is too slow compared with the frequencies of the transients, it allows us to calculate equivalent impedances or equivalent currents for the loads. As a result, the dynamic state of the external system can be inferred from that of the internal system. This can be done very quickly because the state vectors of the two systems are linearly related. However, the state estimate of the external system is good as long as the system operating point remains in a certain neighborhood of the base case at which the equivalent impedances or currents have been calculated. Beyond that neighborhood, their accuracy deteriorates. Analytical expressions of this neighborhood have been derived for constant impedance loads and for constant current loads. The respective neighborhoods are expressed as a function of the changes in the load powers of the external network. These have been verified through simulations carried out on the 32-bus EHV sub-network of the test system.

Another issue addressed in the paper is the on-line identification of the model of a synchronous generator. It is assumed that the voltage phasor, current phasor, and frequency at the generator terminal are known through PMUs. It is also assumed that the rotor angle of the machine is measured in real-time, at the same sampling rate as that of

the PMUs. To comply with the requirements of rapidity and simplicity imposed by an on-line environment, our choice fell on a fast recursive version of the generalized least-squares technique described in [8]. The generator model to be identified is the well-known two-axis transient model with the voltages, frequency, and rotor angle used as state variables (see [9] for instance). The performance of the method is demonstrated on a synchronous generator provided with an IEEE type 1 exciter using values from a real synchronous generator.

This thesis is organized as follows. Chapter 2 describes the original placement methodology, its drawbacks and the suggested improvements, which significantly reduce the number of PMUs needed to capture the transients of a power system. Chapter 3 deals with the calculation of the state of the unobserved system from that of the observed system. In addition, it derives the expressions of the errors in the calculated state as functions of the changes in the loads when the latter are approximated by equivalent impedances or by equivalent currents. Chapter 4 is devoted to the on-line identification of the two-axis transient model of a synchronous generator by means of a recursive generalized least-squares using PMU measurements.

# CHAPTER 2

# PMU PLACEMENT FOR OBSERVABILITY AND GENERATOR MONITORING

The state of the system is needed for use in the determination of stability margins and system security. It is obvious that placing a PMU at every bus in the system would observe the whole system, but this would be prohibitively costly. To reduce the cost, Baldwin, *et al*, [3] have developed an algorithm that determines a minimal placement set of PMUs that observes the whole system, termed *the original placement method.* In this chapter a few refinements of the method are discussed which further reduce the number of PMUs to be placed and make generator modeling possible by forcing the explicit measurement of generator injections.

## 2.1. The Original Placement Algorithm

The Original Placement Algorithm is an integer minimization technique under the constraint that the system is observable. The function to be minimized is the number of PMUs to be placed in the system whereas the controlled variables are the buses where the PMUs are located. The algorithm is a dual-search method using a modified bisecting

technique and a simulated annealing method. To accelerate the search, an initial placement is provided by a graph theoretic approach.

The implementation of the algorithm involves four procedures. The first procedure determines the observable components of a system given a specific PMU placement set. The second procedure creates an initial placement set. The third procedure is a simulated annealing method that finds the optimal placement for a specific number of PMUs given a subset of the initial placement set. The fourth procedure is a bisection method that controls the number of PMUs, which in turn is used by the simulated annealing method will use. The flow diagram of this algorithm is as follows:

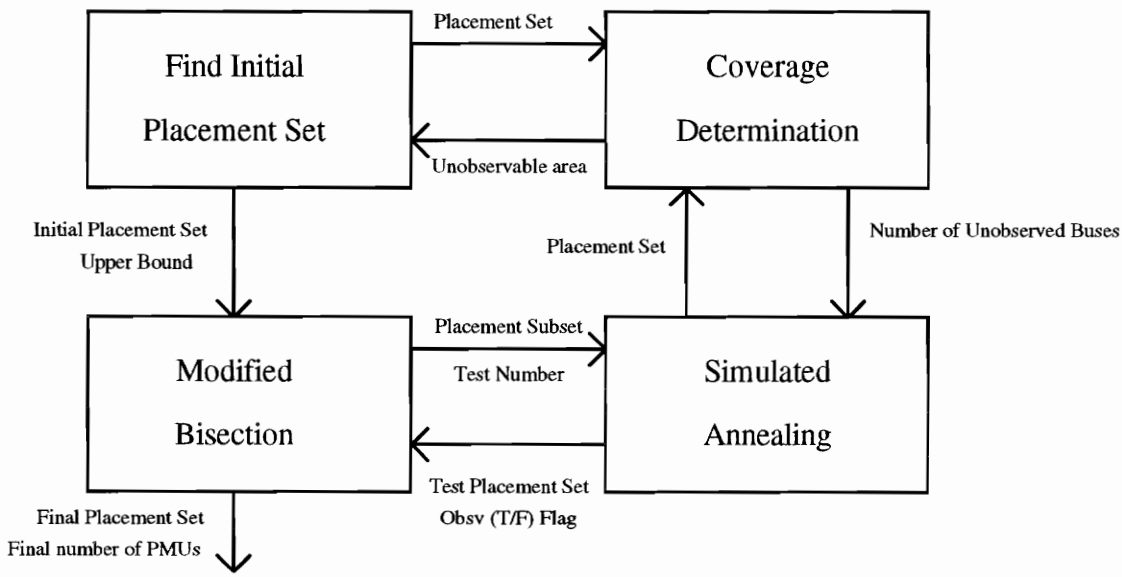


**Figure 2.1. Flow Chart of Original Placement Method**

### 2.1.1. Determination of Observable Region

The observable region consists of the system components which can be monitored by the PMUs. These components are either measured directly by the PMU or calculated from PMU measurements using Ohm's Law or Kirchhoff's Current Law. The PMU measures the voltage phasor for the bus where it is located and the current phasors on all incident branches to that bus.

The algorithm is demonstrated using the IEEE 14 bus system in figure 2.2, with a PMU assumed to be placed on bus 4. It initializes by setting the observable region to the null set. Since a PMU is located at bus 4, the voltage phasor at bus 4, the current phasors on branches 2-4, 3-4, 4-5, 4-7, 4-9, and the load injection at bus 4 are observable and added to the observable region. From the voltage at bus 4 and the currents on branch 2-7, the voltage phasor at bus 7 is determined through Ohm's Law. Similarly the voltage phasor at bus 9 is also calculated. Using the voltage phasors at bus 7 and bus 9, the current on branch 7-9 is available through Ohm's Law. From the 4-7 and 7-9 branch currents, the 7-8 branch current is determined using Kirchhoff's Current Law. By iterating through the system's buses and applying Ohm's and Kirchhoff's Laws where applicable, all the components of the observable region are determined. Figure 2.2 shows the observed voltages marked with a V and the observed branches marked with a dashed-dotted line.

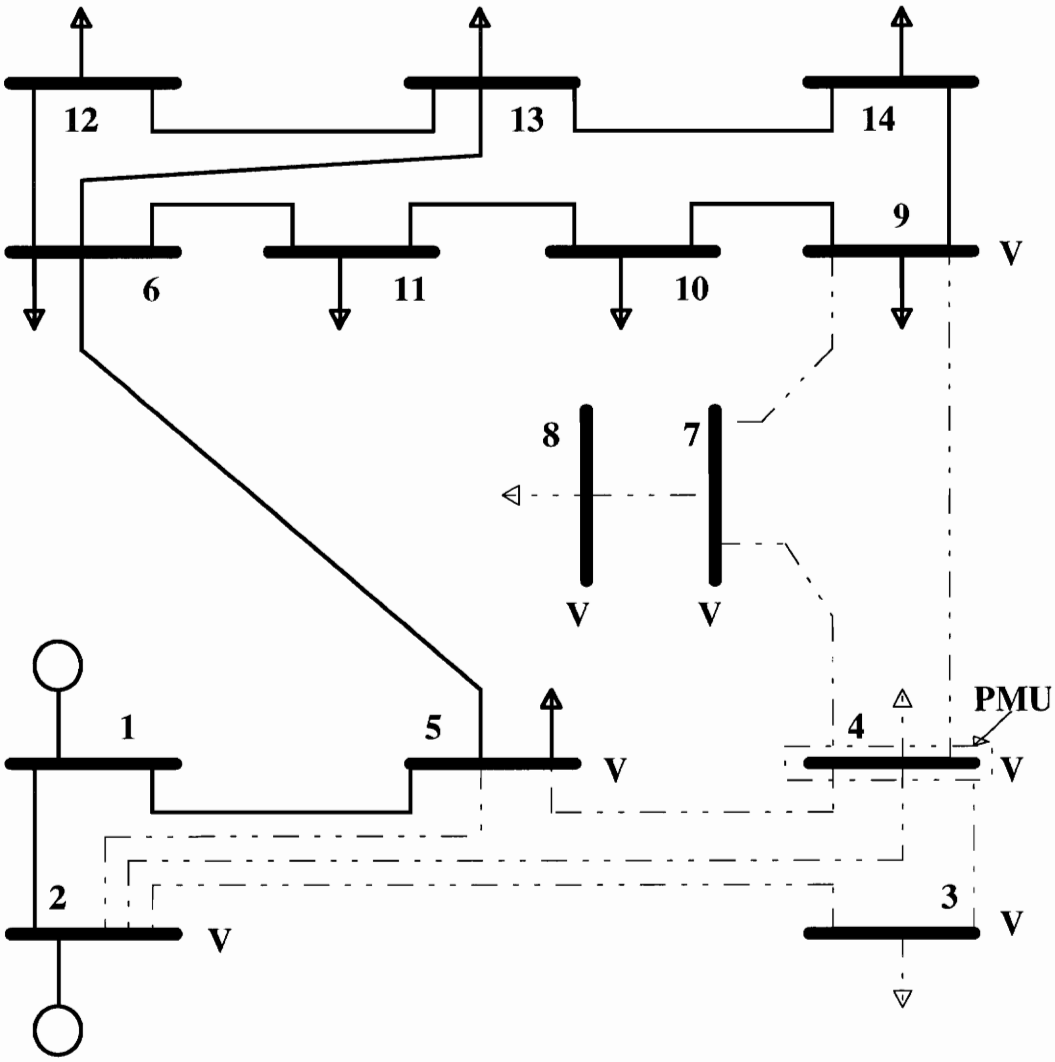


**Figure 2.2. IEEE 14 Bus System with a PMU at Bus 4**

### 2.1.2. Initial Placement Set

The optimal placement algorithm will need an upper bound on the number of PMUs to place in the system. If the upper bound is far away from the optimum, the algorithm will have a high convergence time. This is the case, if the upper bound is set to the total number of buses. To find a more appropriate number, an initial placement set that observes the whole system, is needed.

This initial placement scheme proceeds as follows. In an effort to maximize the number of components observed by an individual PMU, the first step places a PMU on the bus with the greatest number of incident branches. The second step determines the region observed by the current PMU placement set. The third step defines the unobservable region as the region containing all of the components not in the observable region. These three steps are iteratively applied to the remaining unobserved region, until the system is completely observed.

The following example of the initial placement procedure uses the IEEE 14 bus system in figure 2.3. The initial unobservable region is the whole system. Bus 4 is the bus with the most incident branches. Therefore a PMU is placed at this bus. The observable region consists of the voltages at buses 2, 3, 4, 5, 7, 8, & 9, currents on branches 2-3, 2-4, 2-5, 3-4, 4-5, 4-7, 4-9, 7-8, 7-9, and loads at buses 3, 4 and 8. Because the bus in the remaining unobservable region with the most injections is bus 6, a PMU is placed at this bus. From the PMU at 6, the voltages at bus 6, 11, 12, the current on branches 5-6, 6-11, 6-12, 6-13, 12-13, and loads at bus 6 and 12 are now observable. The unobservable system is now broken into three parts and three more PMUs are needed to make the system observable. PMUs are then placed at buses 1, 10, & 14. The system is then found to be observable. It took 5 PMUs to make the 14 bus case observable using this method.

This may not be optimal since it maximizes the number of observable buses for each PMU individually, instead of as a whole, but it provides a reasonable upper bound to the search.

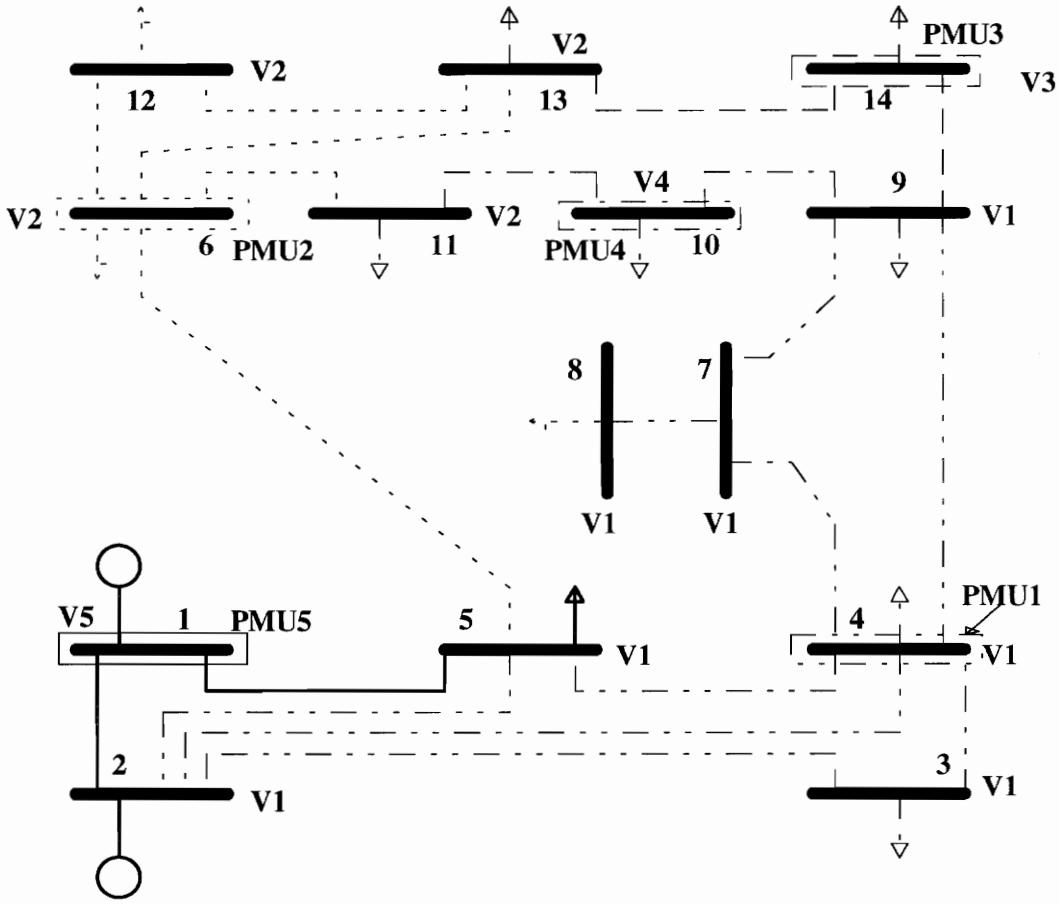


**Figure 2.3. IEEE 14 Bus Case with Regions Observed by Each PMU.**

### 2.1.3. Simulated Annealing

Given a number of PMUs, n, and a subset of the initial placement set containing n PMUs, the simulated annealing method determines if a placement set of n PMUs exists which will observe the system. This method is a minimization technique for discrete valued functions, such as relationship between the size of the unobserved region and the PMU placement set. The optimal solution is found by an iterative procedure that continues until either the system is fully observable or a predetermined limit on the number of iterations has been reached.

The first step randomly selects a PMU in the system and move it to a different bus in the system. The second step decides whether or not that move should be kept. The size of the unobservable region for the new placement set is determined and a change in the size can be calculated. A change that decreases the size of the unobservable region is always accepted. To avoid the local minima in search of the global minimum, the probability of accepting a positive change is greater in the early stages of the search and decreases as the iteration count increases. This probability is based on the Boltzmann probability function $e^{(-\Delta E/T)}$ where $\Delta E$ is the change in the size of the unobservable region and T is a value adjusted by the program to control the probability of accepting an increasing move. With this probability function, a negative change is always accepted, no matter the value of T.

If at any time the size of the unobserved region goes to zero, the routine exits returning the present placement set and an observability flag. The flag is true if the system is observable with the given number of PMUs. Otherwise, the flag is false. When the number of iterations reaches a predetermined maximum, the procedure exits returning the observability flag as false.

### 2.1.4. Bisection Algorithm

While the simulated annealing algorithm finds the optimal placement for a given number of PMUs, the bisection method controls the number of PMUs that the simulated annealing algorithm tries to place. The initial upper bound is given by the initial placement set, and the lower bound is set to 0 PMUs. The algorithm then selects a number (n) of PMUs between the bounds, selects a subset of the initial placement set with n PMUs, and sends the information to the simulated annealing procedure. The simulated annealing procedure then tells the bisection routine whether or not the system was observable by the n PMUs. If the system is observable, this number of PMUs becomes the upper bound. If not, this number becomes the lower bound. When the difference between the upper and the lower bound is 1, the optimal solution has been found. The application of the dual search method to the IEEE 14 bus system results in the placement of 3 PMUs as specified in Figure 2.4

## 2.2. Observed Shortcomings of the Original Placement Method.

Although the original placement method does a good job of finding a minimal set of PMUs that observe a system, the resultant information may not be completely utilized as explained below.

The cost to implement this system may be too high for the restricted budgets of some utilities. Indeed, the cost for the installation of each PMU includes, in addition to the price of the unit itself, the cost of the crew and the cost of the communication channels back to the control center, which is dependent on the number of lines incident to the bus.

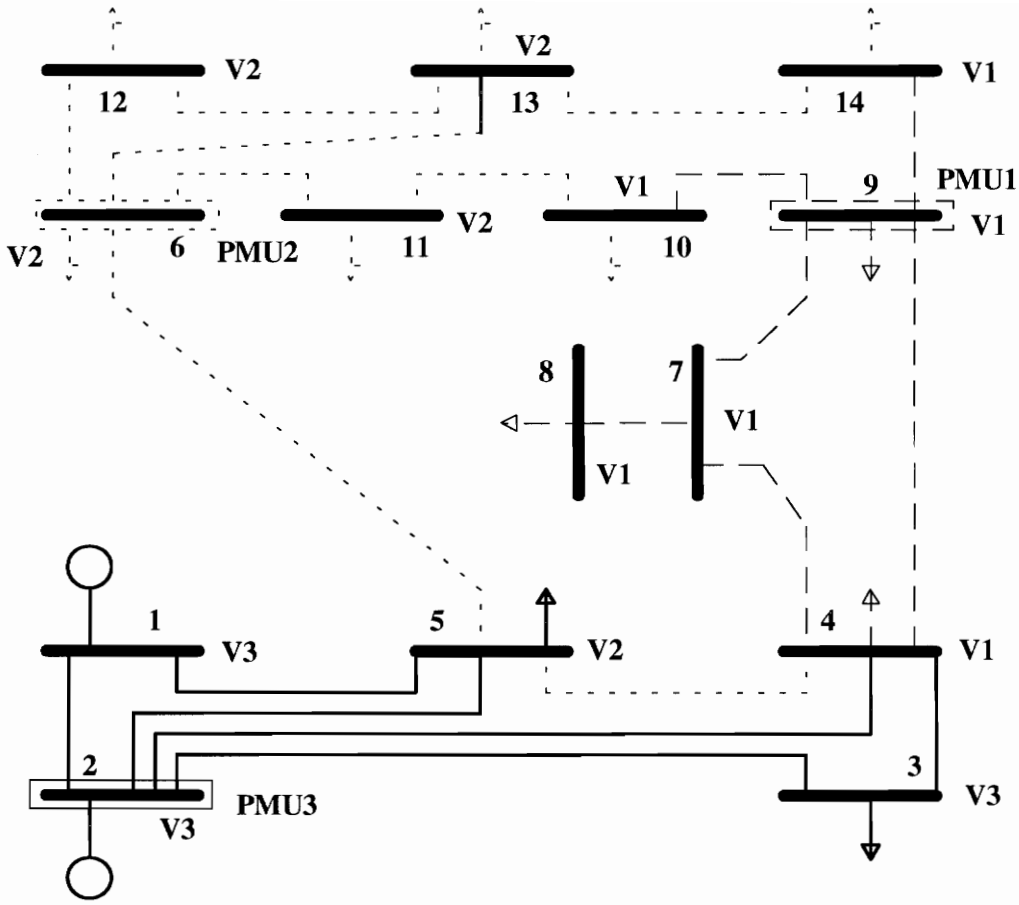


**Figure 2.4. IEEE 14 bus case with Optimal PMU Placement.**

Besides the installation cost, there is the cost associated with receiving and storing the PMUs' data which is significant. Obviously, it is proportional to the number of PMUs to be placed. If the whole system is made completely observable, a large number of PMUs would be needed and the cost would be prohibitive. Therefore, a reduction of the required number of PMUs is needed. This reduction should not result in a significant reduction of information derived from the PMU data.

In addition, the placement scheme should be adjusted to allow the separation of generator and load injections at a common bus. Indeed, the original placement scheme has

the generator and load injections at a common bus lumped together, which precludes the calculation of the generation separately from the load.

## 2.3. Minimal PMU Placement for Capturing System Transients.

The dual search algorithm has been modified so that only a sub-system is observed by PMUs. This subsystem consists of the components that have the greatest effect on the system dynamics. These components are the largest generators, the EHV transmission lines, the tie-lines, and the major dynamic loads. Note that although this sub-system may consist of several disconnected regions, the complete sub-system is observable since the phase angles of the voltages are being measured. This is an improvement over conventional state estimation techniques [10,11] since state estimation requires the measurements of the flows on the paths connecting the disconnected regions before the complete sub-system is observable. The modification of the placement algorithm consists of first providing the terminal buses of the largest generators with PMUs in the graph-theoretic procedure and then not allowing the simulated-annealing method to move the PMUs off the terminal buses.

## 2.4. Application of Placement on a Test System.

The test system was taken from an actual utility and contains 640 bulk transmission buses, with 32 500kV buses and 30 tie-line terminal buses, and over 60 generator plants. Placement on the test system was approached in three different ways:

First, the modified algorithm was applied to the test system together with 30 boundary buses belonging to neighboring systems. The reduced system contains 170 buses including the 30 tie-line terminal buses, the 32 EHV buses, and all the terminal buses of the generators. This system consists of 13 isolated regions, 12 of which are small and 29 isolated buses. The last region is a relatively large connected network that has the EHV network as a backbone and encompasses 49 buses. The reduced system is made observable by providing 45% of its buses with PMUs yielding a total number of 77 PMUs, where some PMUs are forced on the terminals of the generators with outputs greater than 400 MW. The measurement configuration of the connected EHV sub-network is displayed in Figure 2.5.

Second, the algorithm was reapplied to the test system without forcing PMUs on generator terminals. The number of PMUs needed to observe the reduced system, without forcing PMUs on generator buses, is 69 PMUs. This PMU configuration on the connected EHV sub-network is displayed in Figure 2.6.

Third, the algorithm was applied to the test system's EHV sub-system only, since the PMUs would be incrementally placed on the EHV system first. Then, once the EHV system is observed, the utility would decide whether or not to make the rest of the reduced system observable under the placement specified earlier. Thirteen PMUs will observe the 32 bus EHV network with PMUs forced at the EHV buses connected to the generation stations. This placement configuration is displayed in Figure 2.7.

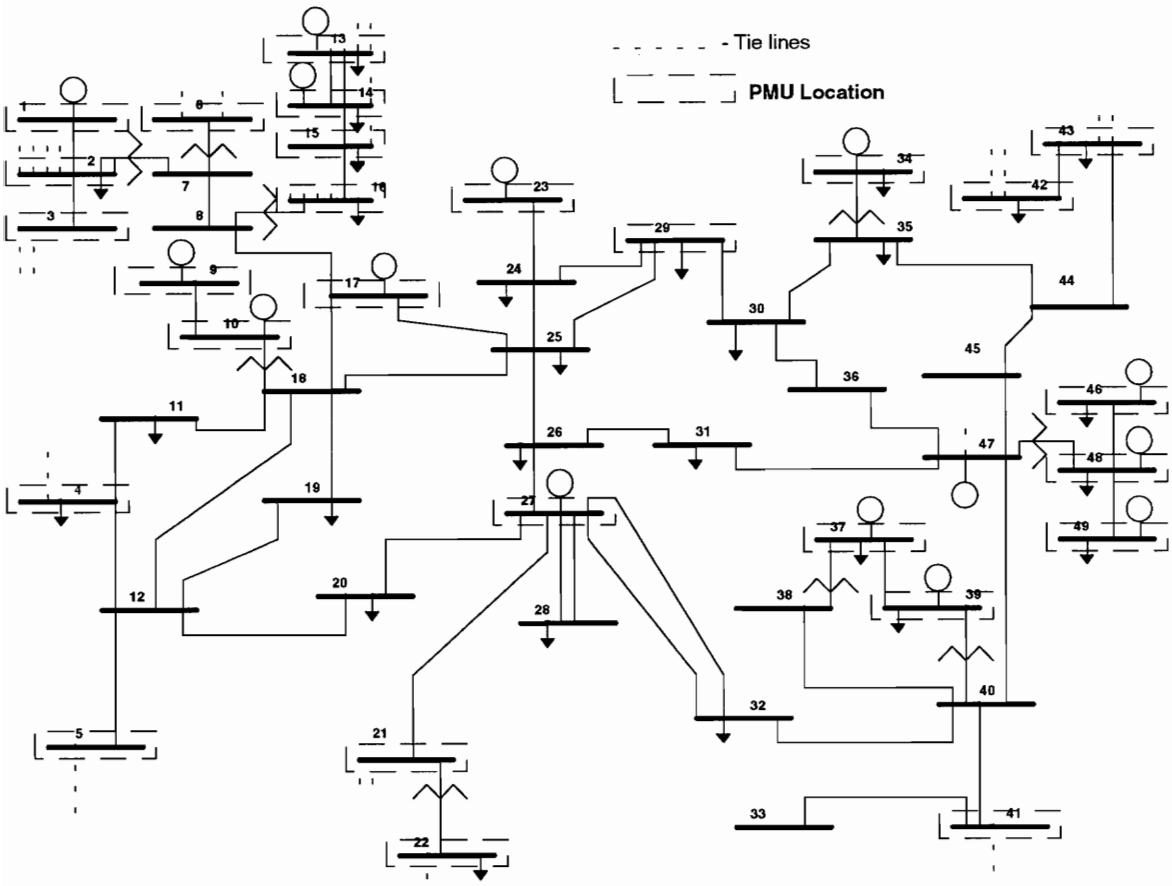


**Figure 2.5. Placement for Observability with Forcing on Generators**

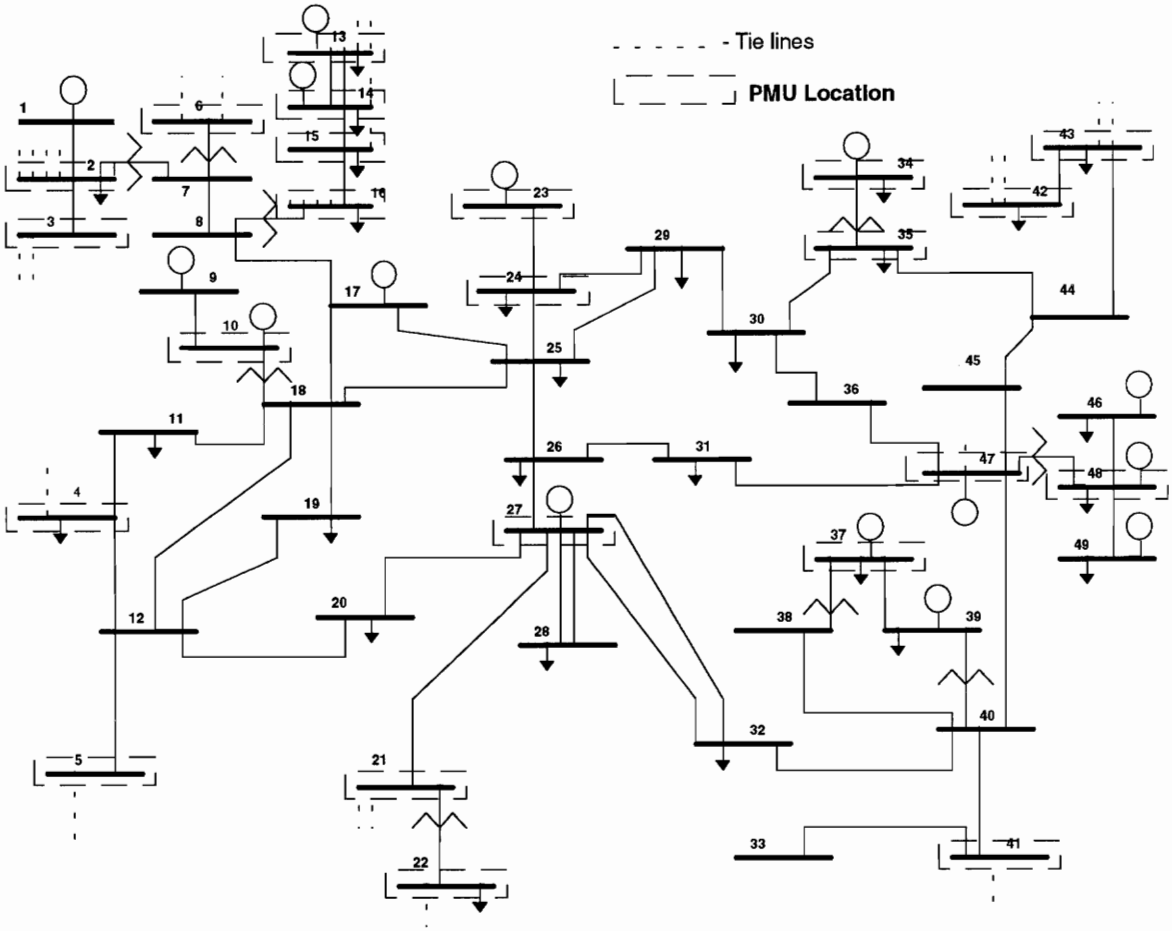


**Figure 2.6. Placement for Observability**

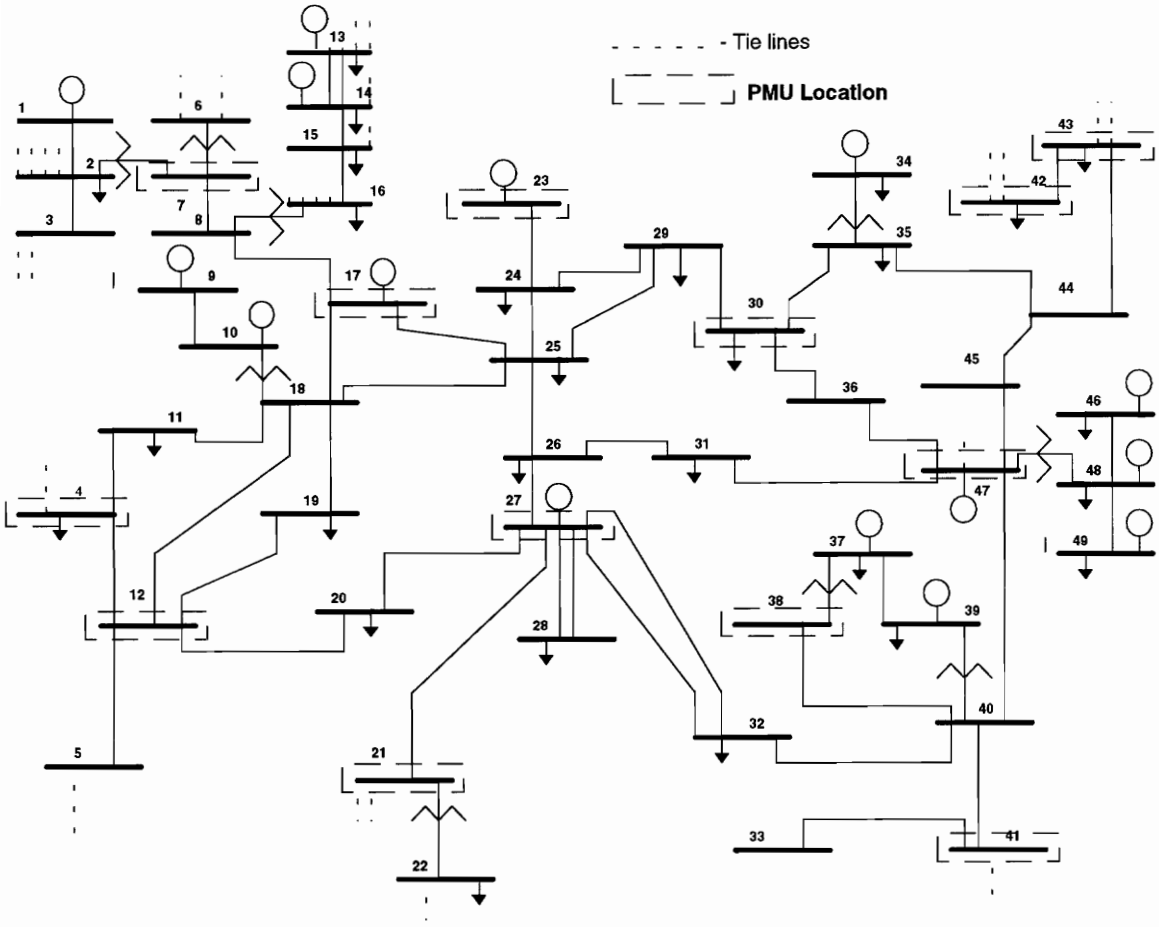


**Figure 2.7. Placement for just the EHV Network Observable**

# CHAPTER 3

# STATE CALCULATION OF THE EXTERNAL SYSTEM

The placement scheme from the previous chapter observes only part of the system called the internal system. The buses not in the internal system are unobserved by the PMUs. This unobserved subsystem is called the external system. Since the state of the entire system is required for stability margin determination and security analysis, a method to estimate the state of the external state from that of the internal state is required. This method must be accurate and fast enough for the sampling rate of the PMUs.

Several techniques for external state estimation appear in the literature involving external equivalents, load flow calculations solutions or state estimation. The external equivalent approach is one of the first methods used to deal with external systems[4,5,6]. The equivalents are determined off-line for normal system configurations. On-line, these equivalents are adjusted so the calculated boundary flows match the measured flows. The equivalents allow for less on-line computation at the expense of the accuracy. In addition, the equivalents do not allow for the inclusion of the changes in the system configurations or of any measured data that might be available in the external system.

In the load flow methods, a full load flow solution on the external system is calculated with the boundary buses of the internal system set to swing buses and the PQ loads of the external system extrapolated from the internal system loads[6,7]. This assumes that the ratio of the external to the internal loads is constant throughout time. The load flow on the external system is then solved while making the calculated boundary

flows match the measured ones. While the load flow solutions allow for the inclusion of system configuration changes and measured data from the external system, the required on-line computation time and memory storage are drawbacks of this method.

In the state estimation techniques, pseudo-measurements in the external system are derived from the state of the internal system and then state estimation is performed on the external system[7]. Care is taken to make the flows at the boundary buses match the measured flows. The state estimation techniques are more flexible than the load flow or equivalent techniques, but the computation time is even longer.

Since a method is needed which is fast enough for the PMU sampling rates, a linear non-iterative approach is developed. The approach assumes that although the external system is not provided with PMUs, it is observed by a set of conventional SCADA-based measurements. It is also supposed that the external system's state is estimated by means of a static state estimator. Therefore, either an equivalent admittance or an equivalent current for each load calculated at each static state estimation run and a linear expression between the voltage phasors of the internal and the external system can be derived. It allows for a very quick assessment of the external state vector whenever a snapshot of the internal system state is carried out from the PMUs.

## 3.1. Estimation of External State using Constant Admittance Loads

### 3.1.1. Relationship between Internal and External State Vectors

Assume without loss of generality that the internal system encompasses all the generator buses whereas the external system consists of the load buses. Consider the voltage-current relationship

$$\begin{bmatrix} \underline{\mathbf{I}}_{\mathrm{g}} \\ \underline{\mathbf{I}}_{\ell} \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{Y}}_{\mathrm{gg}} & \underline{\mathbf{Y}}_{\mathrm{g}\ell} \\ \underline{\mathbf{Y}}_{\ell\mathrm{g}} & \underline{\mathbf{Y}}_{\ell\ell} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{V}}_{\mathrm{g}} \\ \underline{\mathbf{V}}_{\ell} \end{bmatrix}, \tag{3.1}$$

where the subscripts g and $\ell$ represent the generator (internal) and load (external) quantities, respectively. If the loads are converted to constant admittances, then the latter can be added to the sub-matrix $\underline{\mathbf{Y}}_{\ell\ell}$ and the current $\underline{\mathbf{I}}_{\ell}$ can be set to zero, yielding

$$\underline{\mathbf{V}}_{\ell} = -\left(\underline{\mathbf{Y}}_{\ell\ell} + \underline{\mathbf{Y}}_{app}\right)^{-1} \underline{\mathbf{Y}}_{\ell g} \underline{\mathbf{V}}_{g} \ , \tag{3.2}$$

where $\underline{\mathbf{Y}}_{app}$ is a diagonal matrix consisting of the apparent constant admittances.

This expression relates the voltage phasor vector of the internal system, $\underline{\mathbf{V}}_{g}$ , to that of the external system, $\underline{\mathbf{V}}_{\ell}$. As the system evolves, the apparent admittances of the loads change, while the admittances from the base case are still used. The error in the apparent admittance matrix $\underline{\mathbf{Y}}_{app}$ leads to an error in the external voltages calculated from (3.2). To get an understanding of the relative size of these errors, an analytical expression relating these errors is derived next.

### 3.1.2. Errors Analysis of the External State Estimates

Let $\underline{\mathbf{Y}}_{app}$ be the base case apparent admittance matrix calculated at the most recent static state estimation of the external system. Let $\underline{\mathbf{V}}_g^a$ and $\underline{\mathbf{V}}_\ell^a$ be the current system state vectors of the internal and external systems, respectively. They are assumed to be different from those of the base case. The actual apparent admittance matrix associated with the current state is designated by $\underline{\mathbf{Y}}_{app}^a$, which yields an incremental change of $\Delta\underline{\mathbf{Y}}_{app} = \underline{\mathbf{Y}}_{app}^a - \underline{\mathbf{Y}}_{app}$. The external voltage vector $\underline{\mathbf{V}}_\ell^a$ is related to $\underline{\mathbf{V}}_g^a$ through

$$\underline{\mathbf{V}}_\ell^a = -\left(\underline{\mathbf{Y}}_{\ell\ell} + \Delta\underline{\mathbf{Y}}_{app} + \underline{\mathbf{Y}}_{app}\right)^{-1} \underline{\mathbf{Y}}_{\ell g}\, \underline{\mathbf{V}}_g^a \,. \tag{3.3}$$

Note that $\underline{\mathbf{Y}}_{\ell g}$ is independent of the state of the system. The vector $\underline{\mathbf{V}}_\ell^a$ is approximated by the estimate

$$\hat{\underline{\mathbf{V}}}_\ell = -\left(\underline{\mathbf{Y}}_{\ell\ell} + \underline{\mathbf{Y}}_{app}\right)^{-1} \underline{\mathbf{Y}}_{\ell g}\, \underline{\mathbf{V}}_g^a \,. \tag{3.4}$$

From (3.3), it is inferred that

$$\underline{\mathbf{Y}}_{\ell g}\, \underline{\mathbf{V}}_g^a = -\left(\underline{\mathbf{Y}}_{\ell\ell} + \Delta\underline{\mathbf{Y}}_{app} + \underline{\mathbf{Y}}_{app}\right)\underline{\mathbf{V}}_\ell^a \,. \tag{3.5}$$

Substituting (3.5) into (3.4) yields

$$\hat{\underline{\mathbf{V}}}_\ell = \left(\underline{\mathbf{Y}}_{\ell\ell} + \underline{\mathbf{Y}}_{app}\right)^{-1}\left(\underline{\mathbf{Y}}_{\ell\ell} + \Delta\underline{\mathbf{Y}}_{app} + \underline{\mathbf{Y}}_{app}\right)\underline{\mathbf{V}}_\ell^a \,, \tag{3.6}$$

and

$$\Delta\underline{\mathbf{V}}_\ell = \underline{\mathbf{V}}_\ell^a - \hat{\underline{\mathbf{V}}}_\ell = -\left(\underline{\mathbf{Y}}_{\ell\ell} + \underline{\mathbf{Y}}_{app}\right)^{-1} \Delta\underline{\mathbf{Y}}_{app}\, \underline{\mathbf{V}}_\ell^a \,. \tag{3.7}$$

Let $\underline{\mathbf{S}}_{\ell i}$ be the external complex power of the $i$th bus at the base case and let $\Delta\underline{\mathbf{S}}_{\ell i}$ be its incremental change. The change in the admittance matrix is given by

$$\Delta \underline{\mathbf{Y}}_{app} = diag(\frac{\Delta \underline{\mathbf{S}}_{\ell i}^{*} + \underline{\mathbf{S}}_{\ell i}^{*}}{\left\| \underline{\mathbf{V}}_{\ell i}^{a} \right\|^{2}}) - \underline{\mathbf{Y}}_{app}$$

$$= diag\left( \frac{\Delta \underline{\mathbf{S}}_{\ell i}^{*} + \underline{\mathbf{S}}_{\ell i}^{*}}{(\hat{\underline{\mathbf{V}}}_{\ell i} + \Delta \underline{\mathbf{V}}_{\ell i})(\hat{\underline{\mathbf{V}}}_{\ell i} + \Delta \underline{\mathbf{V}}_{\ell i})^{*}} \right) - \underline{\mathbf{Y}}_{app} \quad , \tag{3.8}$$

where $\underline{\mathbf{Y}}_{app} = diag(\frac{\underline{\mathbf{S}}_{\ell i}^{*}}{\left\| \underline{\mathbf{V}}_{\ell i} \right\|^{2}})$ and $diag(\cdot)$ denotes a diagonal matrix. Assuming that $\Delta \underline{\mathbf{V}}_{\ell} \cong 0$, that is, $\hat{\underline{\mathbf{V}}}_{\ell} \cong \underline{\mathbf{V}}_{\ell}^{a}$, and substituting (3.8) into (3.7) yields

$$\Delta \underline{\mathbf{V}}_{\ell} = -\left( \underline{\mathbf{Y}}_{\ell\ell} + \underline{\mathbf{Y}}_{app} \right)^{-1} \left[ diag(1/\hat{\underline{\mathbf{V}}}_{\ell i}^{*})(\Delta \underline{\mathbf{S}}_{\ell}^{*} + \underline{\mathbf{S}}_{\ell}^{*}) - diag(\frac{\underline{\mathbf{S}}_{\ell i}^{*}}{\left| \underline{\mathbf{V}}_{\ell} \right|^{2}}) \hat{\underline{\mathbf{V}}}_{\ell} \right] . \tag{3.9}$$

Taking the infinity norm of both sides of the equation results in

$$\left\| \Delta \underline{\mathbf{V}}_{\ell} \right\| = \left\| \left( \underline{\mathbf{Y}}_{\ell\ell} + \underline{\mathbf{Y}}_{app} \right)^{-1} diag(1/\hat{\underline{\mathbf{V}}}_{\ell i}^{*}) \Delta \underline{\mathbf{S}}_{\ell}^{*} \right\|$$
$$+ \left\| \left( \underline{\mathbf{Y}}_{\ell\ell} + \underline{\mathbf{Y}}_{app} \right)^{-1} (diag(1/\hat{\underline{\mathbf{V}}}_{\ell i}^{*}) \underline{\mathbf{S}}_{\ell}^{*} - diag(\frac{\underline{\mathbf{S}}_{\ell i}^{*}}{\left| \underline{\mathbf{V}}_{\ell} \right|^{2}}) \hat{\underline{\mathbf{V}}}_{\ell}) \right\| . \tag{3.10}$$

Recall that the infinity norm of a vector is the maximum absolute entry of the vector.

### 3.1.3. Simulation Results

Formula (3.10) has been verified on the 32 bus EHV sub-system of the test system used in Chapter 2. Four PMUs were placed at the EHV generator buses. The resulting internal system consists of 17 buses. The remaining 15 buses form the external system. These regions are indicated in Figure 3.1. Two cases of load flow calculations were considered. In the first case, the loads at each bus were randomly changed whereas in the second case the loads were all changed an equal amount. As seen in Tables 3.1 and 3.2

the calculated bound represents the largest error for the given internal state. Indeed, the calculated bounds on $\|\Delta \underline{\mathbf{V}}_\ell\|$ are always greater than the actual $\|\Delta \underline{\mathbf{V}}_\ell\|$, as it should be. It is observed that for the randomly changed load cases, the actual errors are much less than the calculated bound. By contrast, these actual errors are very close to the calculated bound for the cases with equally changed loads.

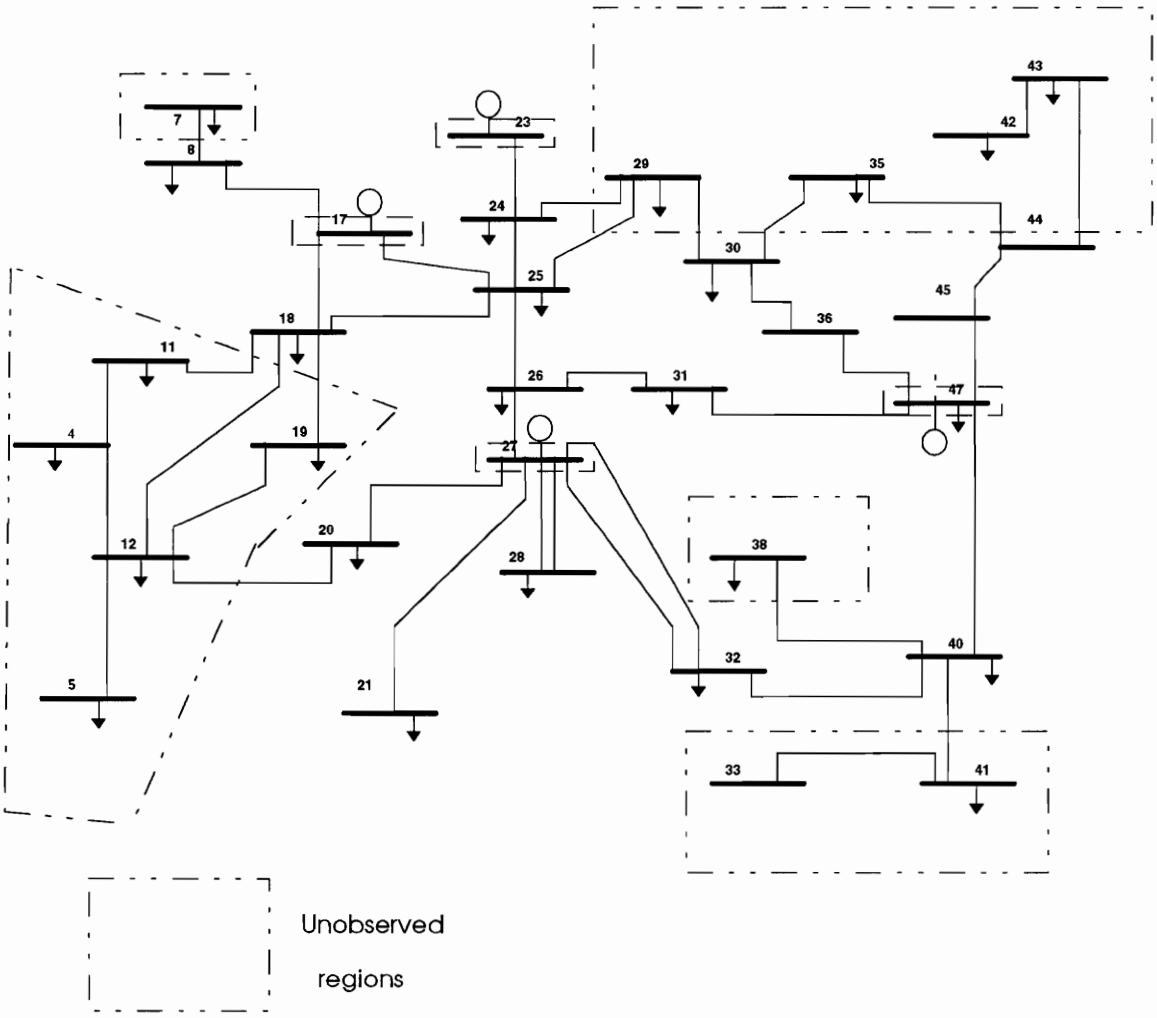


**Figure 3.1. The Test System's EHV Network Used in Chapter 3 Simulations**

**Table 3.1. Actual and Calculated ||ΔV|| when the Loads are Randomly Changed - Constant Impedance Loads.**

| max \|ΔS\| in p.u. | Actual max \|ΔV\| in p.u. | Calculated max \|ΔV\| limit | Difference Calc-Actual |
|---|---|---|---|
| 0.02292 | 0.00053 | 0.00263 | 0.00210 |
| 0.04241 | 0.00078 | 0.00495 | 0.00418 |
| 0.06014 | 0.00091 | 0.00695 | 0.00604 |
| 0.07214 | 0.00100 | 0.00819 | 0.00719 |
| 0.09269 | 0.00146 | 0.01063 | 0.00917 |
| 0.11293 | 0.00170 | 0.01274 | 0.01105 |
| 0.14643 | 0.00291 | 0.01674 | 0.01383 |
| 0.19881 | 0.00258 | 0.02243 | 0.01985 |
| 0.23441 | 0.00592 | 0.02739 | 0.02147 |
| 0.32884 | 0.00624 | 0.03714 | 0.03090 |
| 0.44015 | 0.00774 | 0.04974 | 0.04200 |
| 0.53880 | 0.00710 | 0.06164 | 0.05455 |
| 0.66290 | 0.00896 | 0.07753 | 0.06856 |
| 0.79000 | 0.00854 | 0.09049 | 0.08195 |
| 0.90906 | 0.01359 | 0.10289 | 0.08931 |
| 0.97008 | 0.01789 | 0.11282 | 0.09493 |
| 1.07075 | 0.01338 | 0.12110 | 0.10773 |
| 1.26850 | 0.01921 | 0.14460 | 0.12539 |
| 1.42246 | 0.02167 | 0.16290 | 0.14123 |
| 1.62381 | 0.02631 | 0.18341 | 0.15711 |

**Table 3.2. Actual and Calculated ||ΔV|| when the Loads are Equally Changed - Constant Impedance loads.**

| max ||ΔS|| in p.u. | Actual max ||ΔV|| in p.u. | Calculated max ||ΔV|| limit | Difference Calc-Actual |
|---|---|---|---|
| 0.01717 | 0.00129 | 0.00198 | 0.00069 |
| 0.05303 | 0.00373 | 0.00730 | 0.00357 |
| 0.08839 | 0.01053 | 0.01222 | 0.00169 |
| 0.12374 | 0.01469 | 0.01704 | 0.00235 |
| 0.16957 | 0.01345 | 0.02411 | 0.01066 |
| 0.20592 | 0.01713 | 0.02333 | 0.00620 |
| 0.28094 | 0.01819 | 0.03285 | 0.01466 |
| 0.33588 | 0.02363 | 0.04776 | 0.02413 |
| 0.38439 | 0.02517 | 0.05540 | 0.03023 |
| 0.47045 | 0.04702 | 0.06845 | 0.02144 |
| 0.54801 | 0.03856 | 0.07911 | 0.04054 |
| 0.60087 | 0.03742 | 0.06945 | 0.03203 |
| 0.70443 | 0.05484 | 0.10375 | 0.04891 |
| 0.76014 | 0.10197 | 0.11391 | 0.01195 |
| 0.81317 | 0.05768 | 0.10385 | 0.04617 |
| 0.87775 | 0.06847 | 0.13141 | 0.06294 |
| 0.93692 | 0.10465 | 0.11875 | 0.01410 |
| 1.00763 | 0.14203 | 0.15416 | 0.01213 |
| 1.18440 | 0.08372 | 0.17741 | 0.09369 |
| 1.32949 | 0.10449 | 0.16939 | 0.06490 |
| 1.41421 | 0.15294 | 0.17975 | 0.02681 |

## 3.2. External State Estimation using Constant Current Loads

Another way of modeling the loads is as constant current injections. This load model also results in a linear relationship between the state vectors of the internal and external system. The development of this relationship and the resulting error bound equation uses the same assumptions as that of the constant admittance loads

### 3.2.1. Relationship between the Internal and External State Vectors

Consider the voltage-current relationship (3.1). If the loads are converted to constant current injections, then $\underline{\mathbf{I}}_{\ell}$ can be assigned these injections. They will be designated by $\underline{\mathbf{I}}_{\ell app}$, yielding

$$\underline{\mathbf{V}}_{\ell} = \left(\underline{\mathbf{Y}}_{\ell\ell}\right)^{-1} \left(\underline{\mathbf{I}}_{\ell app} - \underline{\mathbf{Y}}_{\ell g} \underline{\mathbf{V}}_{g}\right) . \tag{3.11}$$

Similarly to the constant admittance case, this expression relates the voltage phasor vector of the internal system, $\underline{\mathbf{V}}_{g}$, to that of the external system $\underline{\mathbf{V}}_{\ell}$. It indicates that the apparent currents of the loads change with the evolution of the system, while the values from the base case are still used. Therefore, the error in the apparent currents leads to errors in the external voltages calculated from (3.11). An analytical expression relating these errors is developed similarly to that of the constant admittance case.

### 3.2.2. Errors Analysis of the External State Estimates.

Let $\underline{\mathbf{I}}_{\ell\,app}$ be the base case apparent current injections calculated at the most recent state estimation of the external system. Let $\underline{\mathbf{V}}_g^a$ and $\underline{\mathbf{V}}_\ell^a$ be the current system state vectors of the internal and external systems, respectively. The actual apparent current injection vector is designated as $\underline{\mathbf{I}}_{\ell\,app}^a$ , which yields an incremental change of $\Delta\underline{\mathbf{I}}_{\ell app} = \underline{\mathbf{I}}_{\ell\,app}^a - \underline{\mathbf{I}}_{\ell app}$. The external voltage vector $\underline{\mathbf{V}}_\ell^a$ is related to $\underline{\mathbf{V}}_g^a$ through

$$\underline{\mathbf{V}}_\ell^a = \underline{\mathbf{Y}}_{\ell\ell}^{-1}(\Delta\underline{\mathbf{I}}_{\ell app} + \underline{\mathbf{I}}_{\ell app} - \underline{\mathbf{Y}}_{\ell g}\,\underline{\mathbf{V}}_g^a)\ . \tag{3.12}$$

Note that $\underline{\mathbf{Y}}_{\ell g}$ is independent of the state of the system. The vector $\underline{\mathbf{V}}_\ell^a$ is approximated by the estimate

$$\hat{\underline{\mathbf{V}}}_\ell = \underline{\mathbf{Y}}_{\ell\ell}^{-1}(\underline{\mathbf{I}}_{\ell app} - \underline{\mathbf{Y}}_{\ell g}\,\underline{\mathbf{V}}_g^a)\ . \tag{3.13}$$

From (3.12), it is inferred that

$$\underline{\mathbf{Y}}_{\ell g}\,\underline{\mathbf{V}}_g^a = (\Delta\underline{\mathbf{I}}_{\ell app} + \underline{\mathbf{I}}_{\ell app} - \underline{\mathbf{Y}}_{\ell\ell}\,\underline{\mathbf{V}}_\ell^a)\ . \tag{3.14}$$

Substituting (3.14) into (3.13) yields

$$\hat{\underline{\mathbf{V}}}_\ell = -\underline{\mathbf{Y}}_{\ell\ell}^{-1}\Delta\underline{\mathbf{I}}_{\ell app} + \underline{\mathbf{V}}_\ell^a, \tag{3.15}$$

and

$$\Delta\underline{\mathbf{V}}_\ell = \underline{\mathbf{V}}_\ell^a - \hat{\underline{\mathbf{V}}}_\ell = \underline{\mathbf{Y}}_{\ell\ell}^{-1}\Delta\underline{\mathbf{I}}_{\ell app}\ . \tag{3.16}$$

Let $\underline{\mathbf{S}}_{\ell i}$ be the external complex power of the $i$th bus at the base case and let $\Delta\underline{\mathbf{S}}_{\ell i}$ be its incremental change. The change in the apparent current is given by

$$\Delta\underline{\mathbf{I}}_{\ell\,app} = diag\left(\frac{\Delta\underline{\mathbf{S}}_{\ell i}^* + \underline{\mathbf{S}}_{\ell i}^*}{\underline{\mathbf{V}}_{\ell i}^{a*}}\right) - \underline{\mathbf{I}}_{\ell app} = diag\left(\frac{\Delta\underline{\mathbf{S}}_{\ell i}^* + \underline{\mathbf{S}}_{\ell i}^*}{(\hat{\underline{\mathbf{V}}}_{\ell i} + \Delta\underline{\mathbf{V}}_{\ell i})^*}\right) - \underline{\mathbf{I}}_{\ell app}\ , \tag{3.17}$$

where $\underline{\mathbf{I}}_{\ell app} = diag(\frac{\underline{\mathbf{S}}^*_{\ell i}}{\underline{\mathbf{V}}^*_{\ell i}})$ and where $diag(\cdot)$ denotes a diagonal matrix. Assuming that $\Delta\underline{\mathbf{V}}_\ell \cong 0$, that is, $\hat{\underline{\mathbf{V}}}_\ell \cong \underline{\mathbf{V}}^a_\ell$, and substituting (3.17) into (3.16) yields

$$\Delta\underline{\mathbf{V}}_\ell = -\underline{\mathbf{Y}}^{-1}_{\ell\ell}\left[ diag(1/\hat{\underline{\mathbf{V}}}^*_{\ell i})(\Delta\underline{\mathbf{S}}^*_\ell + \underline{\mathbf{S}}^*_\ell) - diag(\frac{\underline{\mathbf{S}}^*_{\ell i}}{\underline{\mathbf{V}}^*_\ell}) \right]. \tag{3.18}$$

Taking the infinity norm of both sides of the equation results in

$$\|\Delta\underline{\mathbf{V}}_\ell\| = \left\|\underline{\mathbf{Y}}^{-1}_{\ell\ell} diag(1/\hat{\underline{\mathbf{V}}}^*_{\ell i})\Delta\underline{\mathbf{S}}^*_\ell\right\| + \left\|\underline{\mathbf{Y}}^{-1}_{\ell\ell} diag(1/\hat{\underline{\mathbf{V}}}^*_{\ell i})\underline{\mathbf{S}}^*_\ell) - \underline{\mathbf{Y}}^{-1}_{\ell\ell} diag(\frac{\underline{\mathbf{S}}^*_{\ell i}}{\underline{\mathbf{V}}^*_\ell})\right\|. \tag{3.19}$$

### 3.2.3. Simulation Results

The relationship (3.19) has been verified on the test system used for the constant admittance loads. The same set of cases have been performed. Their results are summarized in Tables 3.3 and 3.4. Again, it is observed that the calculated $\|\Delta\underline{\mathbf{V}}_\ell\|$ are always greater than the actual ones. The actual maximum error is closer to the calculated bound when the loads are changed equally than when the loads are changed randomly.

**Table 3.3. Actual and Calculated ||ΔV|| when the Loads are Randomly Changed - Constant Current Loads.**

| max \|ΔS\| in p.u. | Actual max \|ΔV\| in p.u. | Calculated max \|ΔV\| limit | Difference Calc-Actual |
|---|---|---|---|
| 0.02292 | 0.00056 | 0.00269 | 0.00212 |
| 0.04241 | 0.00069 | 0.00499 | 0.00430 |
| 0.06014 | 0.00087 | 0.00706 | 0.00619 |
| 0.07214 | 0.00096 | 0.00846 | 0.00749 |
| 0.09269 | 0.00143 | 0.01088 | 0.00945 |
| 0.11293 | 0.00171 | 0.01322 | 0.01152 |
| 0.14643 | 0.00290 | 0.01715 | 0.01425 |
| 0.19881 | 0.00258 | 0.02327 | 0.02069 |
| 0.23441 | 0.00564 | 0.02747 | 0.02184 |
| 0.32884 | 0.00627 | 0.03849 | 0.03222 |
| 0.44015 | 0.00771 | 0.05155 | 0.04384 |
| 0.53880 | 0.00677 | 0.06287 | 0.05610 |
| 0.66290 | 0.00962 | 0.07849 | 0.06887 |
| 0.79000 | 0.00861 | 0.09300 | 0.08439 |
| 0.90906 | 0.01384 | 0.10617 | 0.09233 |
| 0.97008 | 0.01801 | 0.11482 | 0.09681 |
| 1.07075 | 0.01397 | 0.12487 | 0.11091 |
| 1.26850 | 0.01950 | 0.14922 | 0.12972 |
| 1.42246 | 0.02174 | 0.16755 | 0.14581 |
| 1.62381 | 0.02632 | 0.18938 | 0.16306 |

**Table 3.4. Actual and Calculated ||ΔV|| when the Loads are Equally Changed - Constant Current Loads.**

| max \|ΔS\| in p.u. | Actual max \|ΔV\| in p.u. | Calculated max \|ΔV\| limit | Difference Calc-Actual |
|---|---|---|---|
| 0.01717 | 0.00126 | 0.00202 | 0.00076 |
| 0.05303 | 0.00459 | 0.00631 | 0.00173 |
| 0.08839 | 0.00951 | 0.01048 | 0.00097 |
| 0.12374 | 0.01326 | 0.01461 | 0.00135 |
| 0.16957 | 0.01224 | 0.02077 | 0.00853 |
| 0.20592 | 0.01705 | 0.02412 | 0.00707 |
| 0.28094 | 0.01904 | 0.03313 | 0.01409 |
| 0.33588 | 0.02866 | 0.04174 | 0.01308 |
| 0.38439 | 0.02734 | 0.04816 | 0.02081 |
| 0.47045 | 0.04030 | 0.05953 | 0.01923 |
| 0.54801 | 0.04646 | 0.06953 | 0.02307 |
| 0.60087 | 0.03870 | 0.07087 | 0.03218 |
| 0.70443 | 0.04909 | 0.09101 | 0.04192 |
| 0.76014 | 0.09274 | 0.09929 | 0.00656 |
| 0.81317 | 0.07242 | 0.08684 | 0.01442 |
| 0.87775 | 0.06267 | 0.11543 | 0.05277 |
| 0.93692 | 0.09229 | 0.09853 | 0.00624 |
| 1.00763 | 0.12919 | 0.13488 | 0.00569 |
| 1.18440 | 0.09867 | 0.15818 | 0.05951 |
| 1.32949 | 0.09673 | 0.13070 | 0.03396 |
| 1.41421 | 0.13549 | 0.13847 | 0.00298 |

## 3.3. Comparison Between the Constant Current and Constant Admittance Load Models

Since the two methods of external state estimation were performed on the exact same cases, the comparison between the two methods is just a comparison of the differences between the two results. Table 3.5 gives the minimum, maximum, average, and variance in both the actual error and the error bound. It is observed that the calculated bounds for the constant admittance cases tend to be higher than the calculated bound for the constant current cases. The difference in the actual maximum error levels averages out to about zero. Given the results of the cases that were run, it is concluded that the constant current load model would provide better results for external state estimation since the constant current bounds are closer to the actual maximum errors.

**Table 3.5. Constant Impedance (CI) vs. Constant Current (CC) in both Actual Error and Error Bound.**

| Statistics | Actual C.I. - Actual C.C. | Calc C.I. - Calc C.C. |
|---|---|---|
| Minimum | -0.02799 | -0.00697 |
| Maximum | 0.019923 | 0.047596 |
| Average | -0.000041 | 0.004882 |
| Variance | 0.0000431 | 0.00011 |

## 3.4. Other Considerations in the Use of this Method

Now that the equations for the error bound of both the constant admittance and constant current techniques have been validated, the remaining issue is how these bounds might be used in reality. The estimates for the state of the external system are provided through (3.2) or (3.11). The equations for the bound on the voltages provide the approximate maximum possible error in the voltage estimate given the state of the internal system and the load models in the external system. The problem with this method is that the loads in the external system are not known. These loads can be approximated from daily load curves for each bus or estimated from the state of the internal system as discussed in the load flow techniques using external state estimates[6,7]. Using these approximated loads in the bound equations affects the accuracy of the bound, not the accuracy of the voltage estimate. This approximated bound is only used to check if the maximum error in the voltage estimates is lower than a threshold value, say 1%. If the bound is less than the threshold, the error in the voltage estimate is assumed small and the voltage estimate can then be accepted.

# CHAPTER 4

# PARAMETER IDENTIFICATION OF SYNCHRONOUS GENERATORS

Once the state of the system is calculated, the next issue to be addressed is to determine how to enhance the real-time monitoring of a power system during transient conditions. VanLandingham *et al.* [12] and Pillay *et al.* [13] have investigated state estimation techniques using Kalman filters and nonlinear observers to identify the internal state of generators. The accuracy of their methods relies on the accuracy of the model parameters, which can be poor. The model parameters may not match the manufacturer's specifications for a number of reasons, such as aging, damage to the machine caused by imbalances and transients, errors in the manufacturer's measurements, or parameters for a different model are supplied. Therefore, a model identification technique is needed to update these models. An on-line, recursive technique that estimates the parameters under operating conditions will overcome the following: 1) unavailability of the machines, 2) unknown disturbances on the system, and 3) limitation on the amount of memory storage.

The algorithm that implements the identification techniques proposed in [14] is summarized in Figure 4.1.

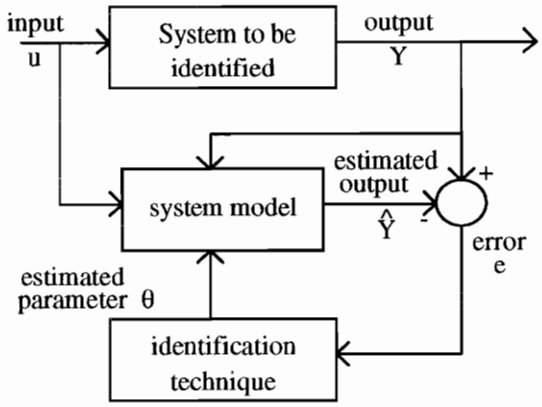


**Figure 4.1. Flowchart of the Parameter Identification Methods (taken from [14])**

Sugiyama *et al.* [15] have developed a parameter identification method under operating conditions while assuming that the field current is metered. This is not always possible, however, since some exciters are on the moving shaft itself. The authors apply curve fitting to the frequency domain analysis of recorded data to determine the transfer function associated with each axis of the generator. In [16], Keyhani, Hao, and Dayal discuss the effect of noise on the frequency domain parameter estimation. Their method is designed for off-line analysis and does not identify the excitation parameters.

Keyhani *et al.* [14,17,18] have investigated the Maximum Likelihood techniques to the identification of the two-axis flux model of the generator shown in Figure 4.2. Although this model is very detailed, its identification requires the knowledge of the field current. The Maximum Likelihood technique estimates the value of the parameters that would maximize the likelihood function [14], which is expressed as

$$\underline{L}(\underline{\theta}) = \prod_{k=1}^{N}\left[\frac{1}{\sqrt{(2\pi)^m \det\left(Cov(e(k),e(k))\right)}}\exp\left(-\frac{1}{2}e^T(k)Cov^{-1}(e(k),e(k))e(k)\right)\right]. \quad (4.1)$$

The Maximum Likelihood technique is well suited for off-line identification. It is too slow for on-line identification unless the probability distribution of the errors is already known.

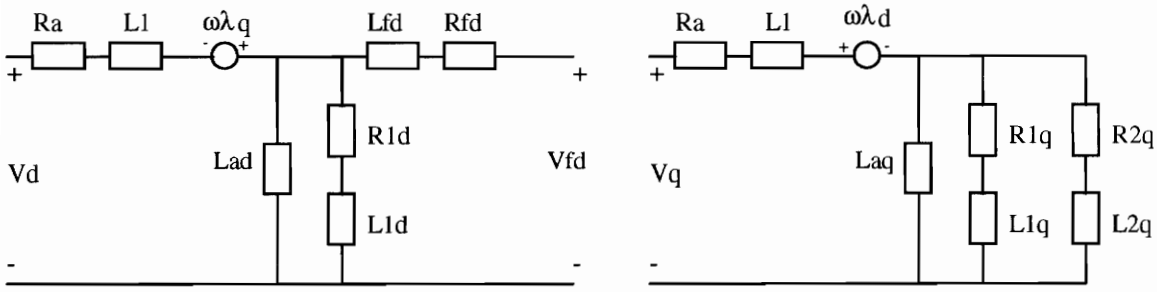


**Figure 4.2. Standard Flux Linkage Generator Circuit Model (taken from [14]).**

The approaches described in [20,22] proceeds as follows. The state equations

$$\begin{aligned} \dot{\underline{\mathbf{x}}} &= \mathrm{f}(\underline{\mathbf{x}},\underline{\theta},\underline{\mathbf{u}},t) \\ \underline{\mathbf{y}} &= \mathrm{h}(\underline{\mathbf{x}},\underline{\theta},\underline{\mathbf{u}},t) \end{aligned} , \tag{4.2}$$

are first linearized

$$\begin{aligned} \dot{\underline{\mathbf{x}}} &= \frac{\partial \mathrm{f}}{\partial \underline{\mathbf{x}}}\partial\underline{\mathbf{x}} + \frac{\partial \mathrm{f}}{\partial \underline{\theta}}\partial\underline{\theta} \\ \underline{\mathbf{y}} &= \frac{\partial \mathrm{h}}{\partial \underline{\mathbf{x}}}\partial\underline{\mathbf{x}} + \frac{\partial \mathrm{h}}{\partial \underline{\theta}}\partial\underline{\theta} \end{aligned} , \tag{4.3}$$

and then an augmented state vector $[\underline{\mathbf{x}} \quad \underline{\theta}]^T$ is formed. The augmented state vector can then be estimated using the extended Kalman filter [19,20] or an observer [21] to estimate. Here, the relationship (4.2) requires linearization at each time step, which can be time consuming.

In this thesis, a recursive version of the generalized least-squares method will be used. The advantage of this method is that it requires no matrix inversion, which makes it

suitable for on-line applications [8]. It will be summarized in Section 4.2. But first, the generator model to be identified is described.

## 4.1. Synchronous Generator Modeling

Various models of synchronous generators are mentioned in the literature. They exhibit different levels of complexity and involve different sets of state variables. The classical one-axis model of the generator shown in figure 4.3 was used in [22]. This model is too simplistic and gives misleading results. The two-axis model with flux decay, used by Keyhani *et al.* [14,17,18], VanLandingham *et al.* [12], and Pillay *et al.* [13], (see Figure 4.2), requires the measurement of the field current for the system to be observable. However, this quantity is not made available by the PMUs, which only measure the voltage and current phasors along with the frequencies on the buses where they are located. To circumvent this difficulty, the selected machine representation is the two-axis transient model where the state variables are the voltages, frequency and rotor angle. This model is observed by the PMUs under the assumption that the generator impedances together with the generator rotor angle are known. The two-axis transient model involves one damper winding on the q-axis and one field winding on the d-axis [9]. Its block diagram is displayed in Figures 4.3 and 4.4. A linearized version of this model has been derived by Varghese, Sauer, and Pai [23].

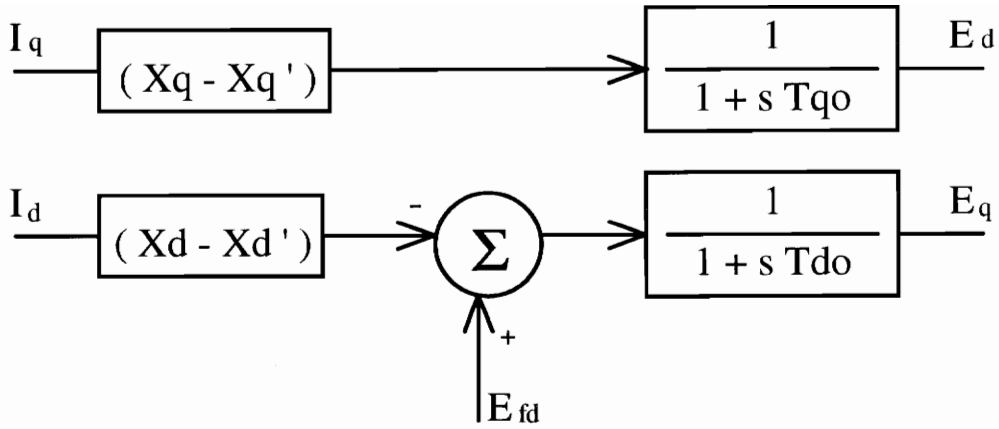


**Figure 4.3. Generator Circuit Model (taken from [9])**

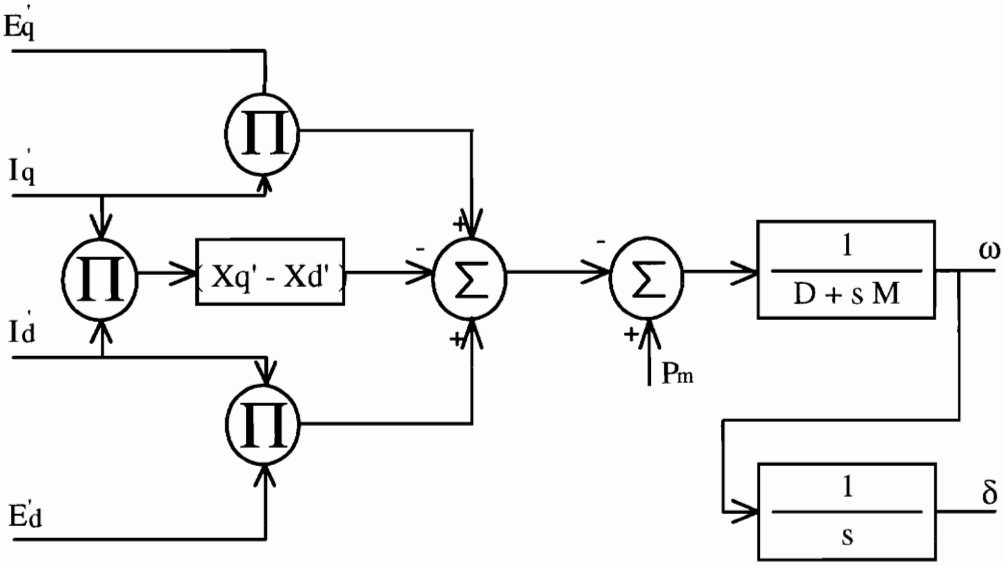


**Figure 4.4. Generator Shaft Model (taken from [9])**

The excitation system is also identified as part of the synchronous generator. Previous work in the literature has either left out the excitation system or treated it as a

first order system. One realistic model for the excitation system is provided by the IEEE type 1 model [24]. Its block diagram is depicted in Figure 4.5.

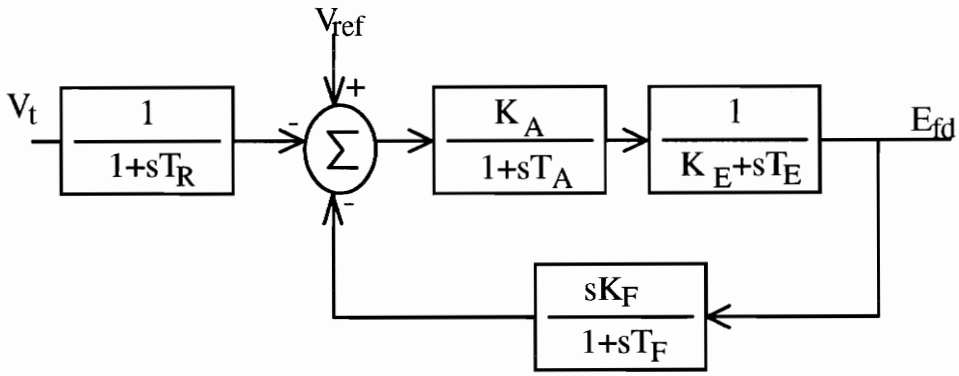


**Figure 4.5. IEEE Type I Excitation System (Taken from [24])**

The linearized model for both the generator and excitation models leads to an eighth order system [9] given by

$$\dot{\underline{\mathbf{x}}} = \underline{\mathbf{A}}\underline{\mathbf{x}} + \underline{\mathbf{B}}\underline{\mathbf{u}}, \tag{4.4}$$

$$\underline{\mathbf{y}} = \underline{\mathbf{C}}\underline{\mathbf{x}} + \underline{\mathbf{D}}\underline{\mathbf{u}}, \tag{4.5}$$

where

$$\underline{\mathbf{x}} = \begin{bmatrix} \Delta E_d^{'} & \Delta E_q^{'} & \Delta\omega & \Delta\delta & \Delta E_{fd} & \Delta V_F & \Delta V_R & \Delta V_1 \end{bmatrix}^T, \tag{4.6}$$

$$\underline{\mathbf{u}} = \begin{bmatrix} \Delta I_d & \Delta I_q & \Delta \mathrm{T}_m & \Delta V_{ref} \end{bmatrix}^T, \tag{4.7}$$

$$\underline{\mathbf{A}} = \begin{bmatrix} \frac{-1}{T_{qo}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-1}{T_{do}} & 0 & 0 & \frac{1}{T_{do}} & 0 & 0 & 0 \\ \frac{-I_{do}}{M} & \frac{-I_{qo}}{M} & \frac{-D}{M} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-K_E}{T_E} & 0 & \frac{1}{T_E} & 0 \\ 0 & 0 & 0 & 0 & \frac{-K_F K_E}{T_F T_E} & \frac{-1}{T_F} & \frac{K_F}{T_F T_E} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-K_A}{T_A} & \frac{-1}{T_A} & \frac{-K_A}{T_A} \\ \frac{V_{do}}{V_{to} T_R} & \frac{V_{qo}}{V_{to} T_R} & 0 & 0 & 0 & 0 & 0 & \frac{-1}{T_R} \end{bmatrix}, \quad (4.8)$$

$$\underline{\mathbf{B}} = \begin{bmatrix} 0 & \frac{X_q - X_q^{'}}{T_{qo}} & 0 & 0 \\ \frac{-(X_d - X_d^{'})}{T_{do}} & 0 & 0 & 0 \\ \frac{-C1}{M} & \frac{-C2}{M} & \frac{1}{M} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{K_A}{T_A} \\ \frac{C3}{T_R} & \frac{C4}{T_R} & 0 & 0 \end{bmatrix}, \quad (4.9)$$

$$\underline{\mathbf{C}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.10}$$

$$\underline{\mathbf{D}} = \begin{bmatrix} -r_s & X_q^{'} & 0 & 0 \\ -X_d^{'} & -r_s & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.11}$$

and

$$C1 = E_{do}^{'} - (L_q^{'} - L_d^{'}) I_{qo}, \tag{4.12}$$

$$C2 = E_{qo}^{'} - (L_q^{'} - L_d^{'}) I_{do}, \tag{4.13}$$

$$C3 = -\frac{V_{do}}{V_{to}} r_s - \frac{V_{qo}}{V_{to}} X_d^{'}, \tag{4.14}$$

$$C4 = -\frac{V_{qo}}{V_{to}} r_s + \frac{V_{do}}{V_{to}} X_q^{'}. \tag{4.15}$$

The notation is that of Anderson and Fouad ([9], pp. 285-292).

Although a state space representation of a system is not unique, one representation can be changed to another through a similarity transformation. The new state space representation will have the same outputs for the same inputs as the original representation. Therefore, the system needs to be converted to a domain that has only one representation for a given set of input and output data. The Laplace transform fits this requirement. For an *n*th order system with two inputs, the functions to be identified are defined as

$$Y(z)=\sum_{i=1}^{2}\frac{a_{i0}+a_{i1}z^{-1}+\cdots+a_{im_i}z^{-m_i}}{1+b_1z^{-1}+\cdots+b_nz^{-n}}U_i(z) \tag{4.16}$$

Several methods are able to identify the parameters $a_{ij}$ and $b_k$ of (4.13). The least-squares method is the fastest one, but produces biased results at the Gaussian model. The weighted least-squares can produce unbiased results if the proper weights are known a priori, a condition rarely satisfied in practice. On the other hand, the generalized least-squares method gives an unbiased parameter estimate, but at the cost of a longer computation time and a greater complexity. A faster method is the iterative generalized least-squares method outline in [8]. It is described next.

## 4.2. The Generalized Least-Squares Identification Method

Let $\underline{\theta}_k$ denote the parameter vector, $\hat{\underline{\theta}}_k$ denote the parameter estimate vector, $\underline{\phi}_k$ denote the measurement vector, and $\hat{\underline{e}}_k$ denote the residual vector. They are given by

$$\underline{\theta}_k=\begin{bmatrix}a_{1,0} & \cdots & a_{1,m_1} & a_{2,0} & \cdots & a_{2,m_2} & b_1 & \cdots & b_n\end{bmatrix}^T, \tag{4.17}$$

$$\underline{\phi}_k=\begin{bmatrix}u_{1,k} & \cdots & u_{1,k-m1} & u_{2,k} & \cdots & u_{2,k-m2} & -y_{k-1} & \cdots & -y_{k-n}\end{bmatrix}^T, \tag{4.18}$$

$$\hat{\underline{e}}=\underline{y}_k-\underline{\phi}_k^T\hat{\underline{\theta}}_k. \tag{4.19}$$

One may assume that the residual vector is an s-order autoregressive model given by

$$\hat{\underline{e}}=\underline{G}_k\underline{\Psi}+\underline{w}_k, \tag{4.20}$$

where

$$\underline{\Psi}=\begin{bmatrix}f_1 & f_2 & \cdots & f_s\end{bmatrix}^T, \tag{4.21}$$

$$\underline{\mathbf{w}}_k = \begin{bmatrix} w_1 & w_2 & \cdots & w_k \end{bmatrix}^T, \tag{4.22}$$

$$\underline{\mathbf{g}}_k = \begin{bmatrix} -\hat{\underline{\mathbf{e}}}_{k-1} & \cdots & -\hat{\underline{\mathbf{e}}}_{k-s} \end{bmatrix}^T, \tag{4.23}$$

and

$$\underline{\mathbf{G}}_k = \begin{bmatrix} \underline{\mathbf{g}}_1 & \underline{\mathbf{g}}_2 & \cdots & \underline{\mathbf{g}}_k \end{bmatrix}^T. \tag{4.24}$$

Here, the noise vector $\underline{\mathbf{w}}_k$ is supposed to be a white sequence. In order to obtain unbiased estimates for $\underline{\boldsymbol{\theta}}_k$, the inputs and outputs are first filtered through

$$\tilde{u}_{ik} = u_{ik} + \sum_{j=1}^{s} \hat{f}_j u_{i,k-j}, \tag{4.25}$$

$$\tilde{y}_k = y_k + \sum_{j=1}^{s} \hat{f}_j y_{k-j}, \tag{4.26}$$

and then used in the following algorithm:

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k + \frac{\mathbf{P}_k \underline{\tilde{\boldsymbol{\phi}}}_{k+1} (\underline{\tilde{\mathbf{y}}}_{k+1} - \underline{\tilde{\boldsymbol{\phi}}}_{k+1}^T \hat{\boldsymbol{\theta}}_k)}{1 + \underline{\tilde{\boldsymbol{\phi}}}_{k+1}^T \mathbf{P}_k \underline{\tilde{\boldsymbol{\phi}}}_{k+1}}, \tag{4.27}$$

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \frac{\mathbf{P}_k \underline{\tilde{\boldsymbol{\phi}}}_{k+1} \underline{\tilde{\boldsymbol{\phi}}}_{k+1}^T \mathbf{P}_k^T}{1 + \underline{\tilde{\boldsymbol{\phi}}}_{k+1}^T \mathbf{P}_k \underline{\tilde{\boldsymbol{\phi}}}_{k+1}}, \tag{4.28}$$

$$\hat{\boldsymbol{\psi}}_{k+1} = \hat{\boldsymbol{\psi}}_k + \frac{\mathbf{R}_k \underline{\mathbf{g}}_{k+1} (\hat{e}_{k+1} - \underline{\mathbf{g}}_{k+1}^{\mathrm{T}} \hat{\mathbf{y}}_k)}{1 + \underline{\mathbf{g}}_{k+1}^T \mathbf{R}_k \underline{\mathbf{g}}_{k+1}}, \tag{4.29}$$

$$\mathbf{R}_{k+1} = \mathbf{R}_k - \frac{\mathbf{R}_k \underline{\mathbf{g}}_{k+1} \left[ \mathbf{R}_k \underline{\mathbf{g}}_{k+1} \right]^T}{1 + \underline{\mathbf{g}}_{k+1}^T \mathbf{R}_k \underline{\mathbf{g}}_{k+1}}, \tag{4.30}$$

where

$$\underline{\tilde{\mathbf{A}}}_{k+1} = \begin{bmatrix} \underline{\tilde{\mathbf{A}}}_k \\ \underline{\tilde{\phi}}_{k+1}^T \end{bmatrix}, \tag{4.31}$$

$$\underline{\mathbf{G}}_{k+1} = \begin{bmatrix} \underline{\mathbf{G}}_k \\ \underline{\mathbf{g}}_{k+1}^T \end{bmatrix}, \tag{4.32}$$

$$\underline{\mathbf{P}}_k = \left[ \underline{\tilde{\mathbf{A}}}_k^T \underline{\tilde{\mathbf{A}}}_k \right]^{-1}, \tag{4.33}$$

$$\underline{\mathbf{R}}_k = \left[ \underline{\mathbf{G}}_k^T \underline{\mathbf{G}}_k \right]^{-1}, \tag{4.34}$$

and where

$$\underline{\tilde{\phi}}_k = \left[ \tilde{u}_{1k} \quad \cdots \quad \tilde{u}_{1,k-m1} \quad \tilde{u}_{2k} \quad \cdots \quad \tilde{u}_{2,k-m2} \quad -\tilde{y}_{k-1} \quad \cdots \quad -\tilde{y}_{k-n} \right]^T. \tag{4.35}$$

Once the parameter vector $\underline{\theta}_k$ is estimated in the discrete-z domain, it needs to be converted to the continuous-time Laplace domain. Several methods for accomplishing this transformation are available. These are the step-invariant method, which assumes constant inputs between time steps, the ramp-invariant method, which assumes a linear input variation between time steps, and the bilinear transformation method, which uses the relationship

$$z^{-1} = \frac{2 - sT}{2 + sT}. \tag{4.36}$$

The latter formula is the linearization of $z = e^{sT}$. It is very fast to calculate. Appendix A derives the expressions relating the continuous time coefficients in terms of the discrete coefficients. Appendix B gives the expressions for the resulting coefficients in terms of the parameters of the model. They are determined by means of

$$\underline{\mathbf{H}} = \underline{\mathbf{C}}(s\underline{\mathbf{I}} - \underline{\mathbf{A}})^{-1}\underline{\mathbf{B}}. \tag{4.37}$$

The parameter estimates are obtained as the solution of a set of simultaneous nonlinear equations, which can be solved using Newton's method. Under the assumption that the parameters are close to their nominal values, the Jacobian matrix can be maintained constant at the nominal values. Although this matrix is non-singular, it is found to be poorly conditioned. This difficulty can be overcome by assuming that $\mathbf{T}_E$ is known, an assumption that will be supposed to hold in the sequel.

## 4.3. Simulation Results

The generalized least-squares identification method was coded using Matlab. This code is listed in Appendix C. The identification method was tested for several values of the size of the autoregressive error model, s, and for a couple of sets of parameters. The generator impedances used in all cases are listed in Table 4.1. Table 4.2 shows the results for one set of parameters obtained from 500 different identification runs where s=3. Table 4.3 and 4.4 show the means and variances resulting from identification with values of s ranging from 1 to 10, but with only 40 runs for value of s. Table 4.5 shows the results for the second set of parameters for values of s of 3 and 6. The measurement noise is assumed to be Gaussian with zero mean and a standard deviation of .01. One requirement of the method is that the system has all its modes excited when the identification is carried out. This is accomplished by ensuring that the power spectrum of the inputs is broad enough. It can be observed from Tables 4.2, 4.3, and 4.5 that the results are satisfactory on the q-axis equation and the shaft dynamics equation, but they are poor in the d-axis differential equation. The latter has characteristic equations with several larger roots that cause inaccuracies in the results when using the bilinear transformations [8]. This

inaccuracy affects all of the identified coefficients for the specific output $\hat{E}_q$, yielding a poor identification of the parameters.

**Table 4.1. Generator Impedances**

| Impedance parameter | Value |
|---|---|
| r | 0.00005 |
| Xd | 1.463 |
| Xq | 1.407 |
| Xd' | 0.222 |
| Xq' | 0.349 |

**Table 4.2. Identification Results of the Transient Model of a Synchronous Generator**

| Parameter | Actual | Identified no noise | Identified : Mean | 1% gaussian noise Variance |
|---|---|---|---|---|
| Tqo' | 0.20000 | 0.20062 | 0.20054 | 0.00000 |
| Tdo' | 4.50000 | 4.55693 | -0.11155 | 0.35096 |
| TE | .05 | N/A | N/A | N/A |
| TF | 0.35000 | 0.40350 | 0.01945 | 0.00032 |
| TA | 0.20000 | - 0.36922 | -0.00706 | 0.00207 |
| TR | 0.02000 | 0.02250 | 0.00009 | 0.00000 |
| KE | 1.00000 | 2.36214 | -455.24785 | 21375.60617 |
| KF | 0.08000 | 0.07154 | -4.96195 | 4.71338 |
| KA | 25.00000 | 59.45453 | -11068.22320 | 13084340.35184 |
| M | 0.40910 | 0.43604 | 0.43603 | 0.00000 |
| D | 0.50000 | 0.53669 | 0.54056 | 0.00000 |

**Table 4.3. Identified Parameter Averages with Different Values of s**

| Parameter | Actual | s | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1.000 | 2.000 | 3.000 | 4.000 | 5.000 | 6.000 | 8.000 | 10.000 |
| Tqo' | 0.200 | 0.196 | 0.194 | 0.198 | 0.198 | 0.198 | 0.199 | 0.199 | 0.199 |
| Tdo' | 4.500 | -0.374 | -0.230 | 0.081 | 0.002 | -1.979 | -0.183 | -0.202 | -0.171 |
| TE | 0.050 | 0.050 | 0.050 | 0.050 | 0.050 | 0.050 | 0.050 | 0.050 | 0.050 |
| TF | 0.350 | 0.025 | 0.027 | 0.031 | 0.031 | 0.033 | 0.032 | 0.032 | 0.025 |
| TA | 0.200 | 0.006 | 0.019 | 0.367 | -0.258 | -0.002 | 0.003 | -0.008 | -0.006 |
| TR | 0.020 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| KE | 1.000 | -361.61 | -363.15 | -316.33 | -346.99 | -287.09 | -334.47 | -303.49 | -332.70 |
| KF | 0.080 | -2.818 | -2.175 | -1.802 | -1.131 | -1.787 | -2.348 | -1.241 | -1.962 |
| KA | 25.000 | -8.9e3 | -9.0e3 | -7.8e3 | -8.6e3 | -7.1e3 | -8.2e3 | -7.5e3 | -8.2e3 |
| M | 0.409 | 0.434 | 0.435 | 0.434 | 0.434 | 0.433 | 0.434 | 0.433 | 0.433 |
| D | 0.500 | 0.569 | 0.568 | 0.555 | 0.554 | 0.556 | 0.554 | 0.553 | 0.557 |

**Table 4.4. Variances of Identified Parameters with Different Values of s**

| Parameter | s | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1.000 | 2.000 | 3.000 | 4.000 | 5.000 | 6.000 | 8.000 | 10.000 |
| Tqo' | 4.0E-05 | 4.1E-05 | 3.1E-05 | 3.3E-05 | 1.8E-05 | 3.7E-05 | 3.0E-05 | 2.7E-05 |
| Tdo' | 1.4E+00 | 2.6E-01 | 3.5E+00 | 5.8E-01 | 2.2E+02 | 1.8E-01 | 3.9E+00 | 1.5E+00 |
| TE | 2.4E-33 | 2.4E-33 | 2.4E-33 | 2.4E-33 | 2.4E-33 | 2.4E-33 | 2.4E-33 | 4.4E-34 |
| TF | 3.2E-04 | 2.5E-04 | 1.8E-04 | 3.0E-04 | 2.1E-04 | 3.9E-04 | 2.4E-04 | 6.9E-05 |
| TA | 1.1E-04 | 4.8E-03 | 8.1E+00 | 3.7E+00 | 8.4E-04 | 4.3E-03 | 8.8E-03 | 6.7E-03 |
| TR | 4.3E-07 | 6.2E-09 | 1.1E-07 | 3.7E-09 | 1.5E-07 | 8.6E-09 | 3.7E-09 | 3.2E-08 |
| KE | 5.5E+04 | 3.5E+04 | 3.7E+04 | 3.6E+04 | 2.4E+04 | 4.2E+04 | 3.6E+04 | 3.1E+04 |
| KF | 1.3E+01 | 1.4E+01 | 9.9E+00 | 9.0E+00 | 1.2E+01 | 1.3E+01 | 9.9E+00 | 1.3E+01 |
| KA | 3.6E+07 | 2.3E+07 | 2.5E+07 | 2.3E+07 | 1.7E+07 | 2.8E+07 | 2.4E+07 | 2.1E+07 |
| M | 9.3E-06 | 6.5E-06 | 5.6E-06 | 5.8E-06 | 2.5E-05 | 8.8E-06 | 2.1E-05 | 8.4E-06 |
| D | 1.5E-06 | 1.1E-06 | 7.9E-05 | 7.0E-05 | 2.9E-04 | 7.4E-05 | 1.4E-04 | 7.9E-05 |

**Table 4.5. Identification with a Different Set of Parameters**

| Parameter | Actual | 3.000 mean | 3.000 variance | 6.000 mean | 6.000 variance |
|---|---|---|---|---|---|
| Tqo' | 0.500 | 0.491 | 3.43E-04 | 0.496 | 3.23E-04 |
| Tdo' | 4.000 | 0.001 | 2.11E-05 | 0.001 | 2.85E-05 |
| TE | 0.10 | 0.100 | 3.81E-32 | 0.100 | 3.81E-32 |
| TF | 0.750 | 0.000 | 1.92E-07 | 0.000 | 2.77E-06 |
| TA | 0.300 | 0.000 | 4.80E-11 | 0.000 | 3.04E-11 |
| TR | 0.20 | 0.000 | 2.03E-11 | 0.000 | 2.23E-10 |
| KE | 0.5 | 5.4e4 | 6.49E+08 | 4.5e4 | 6.09E+08 |
| KF | 0.25 | 1.05e3 | 6.30E+06 | 643.50 | 4.38E+06 |
| KA | 6.000 | 6.4e5 | 9.87E+10 | 5.4e5 | 9.13E+10 |
| M | 0.8 | 0.821 | .000174 | 0.805 | .00686 |
| D | 0.2 | 0.235 | .0000155 | 0.268 | .06520 |

# CHAPTER 5

# CONCLUSIONS

The study investigates a PMU placement methodology that significantly reduces the number of PMUs to be located in the network while capturing the transients of the system. It consists of placing the minimum number of PMUs that observe only the extra-high voltage sub-network together with the terminal buses of the major generators, dynamic loads, and tie-lines. The state of the remaining sub-network is inferred very quickly from that of the sub-system observed by the PMUs. The approach makes use of a linear relationship that is derived from converting all loads to equivalent impedances or to equivalent currents. These equivalents are updated whenever the system operating point moves beyond a certain neighborhood expressed in terms of the incremental changes in the load powers. The derived expression for this neighborhood only provides bounds on the voltage magnitudes. A bound on the voltage angle is still needed.

The study also explores the on-line identification of the parameters of a synchronous generator from the data provided by a PMU located at its terminal bus. The use of a recursive version of the generalized least-squares method is advocated. It is based on the assumptions that (i) the parameters of the two-axis transient model are close to their nominal values, (ii) the rotor angle is measured in real-time, and (iii) the measurement errors are Gaussian with zero means and small variances. Simulations showed that good results are obtained when the order and the poles of the transfer functions are small. However, when the order or some of the poles are large, the bilinear

transformation leads to inaccurate results. This is especially true for the Eq to Id and Eq to Iq transfer functions, which are sixth order function with large poles. Hence the need to investigate alternative methods that identify directly in the continuous time domain. This avoids the uneasy conversion from the discrete to the continuous time domain. Potential candidates include the Kalman filter methods and the ramp-invariant methods that identify a canonical state-space model of the system.

# CHAPTER 6

# REFERENCES


[1] Arun G. Phadke and James S. Thorp, "Improved Control and Protection of Power Systems Through Synchronized Phasor Measurements", in *Control and Dynamic Systems*, Vol 43, edited by L. C. Leondes, Academic Press, 1991, pp. 335-376.

[2] IEEE Subcommittee Report, "Synchronized Sampling and Phasor Measurements for Relaying and Control", Paper No. 93 WM 039-8 PWRD, Presented at the IEEE PES Winter Meeting, Feb 1 - Feb 5, 1993, Columbus, Ohio.

[3] T. L. Baldwin, L. Mili, and M. B. Boisen, Jr and R. Adapa, "Power System Observability with Minimal Phasor Measurement Placement", *IEEE Transactions on Power Systems*, Vol. 8, No. 2, May 1993, pp. 707-715.

[4] C. N. Lu, K.C. Liu, and S. Vemuri, "An External Network Modeling Approach for On-Line Security Analysis", *IEEE Transactions on Power Systems*, Vol. 5, No. 2, May 1990, pp. 565-573.

[5] J. F. Dopazo, M. H. Dwarakanath, J. J. Li, and A. M. Sasson, "An External System Equivalent Model Using Real-Time Measurements for System Security Evaluation", *IEEE Transactions of Power Apparatus and Systems*, Vol. PAS-96, No. 2, March/April 1977, pp. 431-446.

[6] F. F. Wu and A. Monticelli, "Critical Review of External Network modeling for Online Security Analysis", *Electrical Power and Energy Systems*, Vol. 5, Oct. 1983, pp.222-235.

[7] K. I. Geisler and A. Bose, "State Estimation Based External Network Solution For On-line Security Analysis", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-102, No. 8, August 1983, pp. 2447-2454.

[8] N. K. Sinha and B. Kuszta, *Modeling and Identification of Dynamic Systems*, Van Nostrand Reinhold Company, New York, 1983.

[9] P. M. Anderson and A. A. Fouad, *Power System Control and Stability*, The Iowa State University Press, Ames, IA, 1977.

[10] E. Hanschin and F. D. Galiana, "Hierarchical State Estimation For Real-Time Monitoring of Electric Power Systems", *8th PICA Conference*, Minneapolis, Minnesota, June 4-6, 1973. pp. 304-312.

[11] T. V. Cutsem and M. Ribbens-Pavella, "Critical Survey of Hierarchical Methods for State Estimation of Electric Power Systems", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-102, No. 10, 1983, pp. 3415-3425.

[12] H. F. VanLandingham, D. K. Lindner, A. G. Phadke, A. Jayakumar, and J. S. Thorp, "A Nonlinear Observer Applied to State Estimation of a Synchronous Generator", *Proceedings of the 1984 International Conference on Industrial Electronics, Control, and Instrumentation*, Tokyo, Japan, pp. 559-563.

[13] P. Pillay, A. G. Phadke, D. K. Lindner, and J. S. Thorp, "State Estimation for a Synchronous Machine: Observer and Kalman Filter Approach", *Proceedings of the 1986 Conference on Information Sciences and Systems*, Princeton, NJ, pp. 520-525.

[14] A. Keyhani and S. I. Moon, "Maximum Likelihood Estimation of Synchronous Machine Parameters and Study of Noise Effect from DC Flux Decay Data", *IEE-Proceedings-C*, Vol. 139, No. 1, January 1992, pp.76-80.

[15] T. Sugiyama, T. Nishiwaki, S. Takeda, and S. Abe, "Measurements of Synchronous Machine Parameters Under Operating Conditions", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-101, No. 4. April 1982, pp. 895-904.

[16] A. Keyhani, S. Hao, and G. Dayal, "The Effects of Noise on Frequency Domain Parameter Estimation of Synchronous Machine Models", *IEEE Transactions on Energy Conversion*, Vol. 4, No. 4, December 1989, pp. 600-607.

[17] A. Keyhani, S. Hao, and G. Dayal, "Maximum Likelihood Estimation of Solid-Rotor Synchronous Machine Parameters from SSFR Test Data", *IEEE Transactions on Energy Conversion*, Vol. 4, No. 3, September 1989, pp. 551-558.

[18] A. Keyhani, H. Tsai, and T. Leksan, "Maximum Likelihood Estimation of Synchronous Machine Parameters From Standstill Time Response Data", Paper No. 93 WM 023-2 EC, presented at the IEEE PES Winter Meeting, Feb. 1 - 5, 1993, Columbus, Ohio.

[19] M. Namba, T. Nishiwaki, S. Yokokawa, K. Ohtsuka, and Y. Ueki, "Identification of Parameters for Power System Stability Analysis using Kalman Filter", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-100, No. 7, July 1981, pp. 3304-3311.

[20] N. Jaleeli, M. S. Bourawi, and J. H. Fish III, "A Quasilinearization Based Algorithm for the Identification of Transient and Subtransient Parameters of Synchronous Machines", *IEEE Transactions on Power Systems*, Vol. PWRS-1, No. 3, August 1986, pp. 46-52.

[21] A. Keyhani and S. M. Miri, "Observers for Tracking of Synchronous Machine Parameters and Detection of Incipient Faults", *IEEE Transactions on Energy Conversion*, Vol. EC-1, No. 2, June 1986, pp. 184-192.

[22] Zhiyuan, Nan, Zhengiang Mi, Jintao Ma, and Yihan Yang, "On-line Parameter Identification of Synchronous Machines in Electric Power Systems", *Proceedings of the 9th IFAC/IFORS Symposium on Identification and System Parameter Estimation*, July 8-12, 1991, Budapest, Hungary, pp. 1106-1110.

[23] A. Varghese, P. W. Sauer, and M. A. Pai, "Synchronous Machine Block Diagram Analysis with Fast Dynamics", *Electrical Power and Energy Systems*, Vol. 11, No. 4, October 1989, pp. 239-247.

[24] IEEE Committee Report, "Computer Representation of Excitation Systems", *IEEE Transactions on Power Apparatus and Systems,* Vol. PAS-87, June 1968, pp. 1460-1463.

# APPENDIX A

# EQUATIONS FOR BILINEAR TRANSFORMATION

## A.1. Conversion for First Order Equations From z to s Domain

The technique identifies the z-domain transfer function

$$\frac{a_0 + a_1 \mathrm{z}^{-1}}{1 + b_1 \mathrm{z}^{-1}}, \tag{A.1}$$

which is converted to the s-domain transfer function, yielding

$$\frac{m_1 \mathrm{s} + m_0}{\mathrm{s} + n_0}. \tag{A.2}$$

Using the bilinear relationship of $\mathrm{z}^{-1} = \dfrac{2 - \mathrm{s}T}{2 + \mathrm{s}T}$, the relationship (A.1) becomes

$$\frac{a_0 + a_1 \dfrac{2 - \mathrm{s}T}{2 + \mathrm{s}T}}{1 + b_1 \dfrac{2 - \mathrm{s}T}{2 + \mathrm{s}T}}. \tag{A.3}$$

The latter reduces to

$$\frac{a_0(2 + \mathrm{s}T) + a_1(2 - \mathrm{s}T)}{2 + \mathrm{s}T + b_1(2 - \mathrm{s}T)} = \frac{sT(a_0 - a_1) + 2(a_0 + a_1)}{sT(1 - b_1) + 2 + 2b_1}. \tag{A.4}$$

Therefore, the values of the coefficients of the s-domain transfer function are given by

$$n_o = \frac{2+2b_1}{T(1-b_1)} \quad ,$$

$$m_1 = \frac{(a_0 - a_1)}{(1-b_1)} \quad ,$$

$$m_o = \frac{2(a_0 + a_1)}{T(1-b_1)} \quad . \tag{A.5}$$

## A.2. Conversion Equations for Bilinear Transformation of Sixth Order Equation (A.8)

The identified model is given in the z-domain as

$$\frac{\sum_{i=0}^{i=6} a_i \, \mathrm{z}^{-i}}{1+\sum_{i=1}^{i=6} b_i \, \mathrm{z}^{-i}} \quad , \tag{A.6}$$

which corresponds to the desired s-domain transfer function

$$\frac{m_1 \, \mathrm{s} + m_0}{\mathrm{s} + n_0} \quad . \tag{A.7}$$

Using the bilinear relationship $\mathrm{z}^{-1} = \frac{2-\mathrm{s}T}{2+\mathrm{s}T}$ in (A.6) and then reducing the obtained equations and grouping the coefficients of s together yields

$$m_6 = T^6(a_o - a_1 + a_2 - a_3 + a_4 - a_5 + a_6)/D \quad ,$$

$$m_5 = T^5(12a_o - 8a_1 + 4a_2 - 4a_4 + 8a_5 - 12a_6)/D \quad ,$$

$$m_4 = T^4(60a_o - 20a_1 - 4a_2 + 12a_3 - 4a_4 - 20a_5 + 60a_6)/D \quad ,$$

$$m_3 = T^3(160a_o - 32a_2 + 32a_4 - 160a_6)/D \quad ,$$

$$m_2 = T^2(240a_o + 80a_1 - 16a_2 - 48a_3 - 16a_4 + 80a_5 + 240a_6)/D \quad ,$$

$$m_1 = T(192a_o + 128a_1 + 64a_2 - 64a_4 - 128a_5 - 192a_6)/D \quad ,$$

$$m_0 = 64(a_o + a_1 + a_2 + a_3 + a_4 + a_5 + a_6)/D \quad ,$$

$$n_6 = T^6(b_o - b_1 + b_2 - b_3 + b_4 - b_5 + b_6)/D \quad ,$$

$$n_5 = T^5(12 - 8b_1 + 4b_2 - 4b_4 + 8b_5 - 12b_6)/D \quad ,$$

$$n_4 = T^4(60 - 20b_1 - 4b_2 + 12b_3 - 4b_4 - 20b_5 + 60b_6)/D \quad ,$$

$$n_3 = T^3(160 - 32b_2 + 32b_4 - 160b_6)/D \quad ,$$

$$n_2 = T^2(240 + 80b_1 - 16b_2 - 48b_3 - 16b_4 + 80b_5 + 240b_6)/D \quad ,$$

$$n_1 = T(192 + 128b_1 + 64b_2 - 64b_4 - 128b_5 - 192b_6)/D \quad ,$$

$$n_0 = 64(1 + b_1 + b_2 + b_3 + b_4 + b_5 + b_6)/D \quad , \tag{4.8}$$

where

$$D = T^6(1 - b_1 + b_2 - b_3 + b_4 - b_5 + b_6) \quad .$$

# APPENDIX B

# EQUATIONS RELATING THE PARAMETERS WITH THE S-DOMAIN COEFFICIENTS

The relationship between the transfer functions of the generator and exciter in terms of the parameters of the machines are determined through

$$\underline{\mathbf{H}} = \underline{\mathbf{C}}(s\underline{\mathbf{I}} - \underline{\mathbf{A}})^{-1}\underline{\mathbf{B}} \quad , \tag{B.1}$$

where the state matrices, A, B, and C are given by (4.5-4.7).

## B.1. The q-axis Equations

The q-axis equations reduce to the first order transfer function

$$E_d^{'} = \frac{(X_q - X_q^{'})}{1 + sT_{qo}^{'}} I_q = \frac{(X_q - X_q^{'}) / T_{qo}^{'}}{1 / T_{qo}^{'} + s} I_q \, . \tag{B.2}$$

The resulting relationship between the coefficients and the parameters is apparent in the equation itself. So it is not given separately.

## B.2. The Shaft Dynamics Equation

The resulting relationship between the rotor speed and the d and q -axis currents is of seventh order and, hence, is very difficult to identify. But since the voltage phasor and the current phasor are measured, the real power output of the machine can be calculated. The resulting transfer function is given as

$$\omega = \frac{1}{D + sM}\left(P_m - P_e\right) = \frac{1/M}{D/M + s}\left(P_m - P_e\right) \ . \tag{B.3}$$

Again the coefficient - parameter relationship is obvious and is not further detailed.

## B.3. The d-axis Equations

The d-axis equation is a sixth order transfer function expressed as

$$E_q^{'} = \frac{\sum_{i=0}^{5} a_i s^i}{s^6 + \sum_{i=0}^{5} b_i s^i} I_d + \frac{\sum_{i=0}^{2} c_i s^i}{s^6 + \sum_{i=0}^{5} b_i s^i} I_q \ . \tag{B.4}$$

The resulting relationships between the coefficients $a_i$, $b_i$, and $c_i$ and the parameters are written as

$$a_5 = \frac{-X_{df}}{T_d}, \; where \; X_{df} = X_d - X_d^{'} \quad ,$$

$$a_4 = \frac{-X_{df}}{T_d}\left(\frac{1}{T_A} + \frac{K_E}{T_E} + \frac{1}{T_F} + \frac{1}{T_q} + \frac{1}{T_R}\right) \quad ,$$

$$a_3 = \frac{-X_{df}}{T_d}\left(\begin{array}{l} \frac{K_E}{T_A T_E} + \frac{K_E}{T_E T_F} + \frac{1}{T_F T_A} + \frac{1}{T_q T_A} + \frac{1}{T_R T_A} + \frac{1}{T_q T_R} + \frac{1}{T_R T_F} + \frac{1}{T_F T_q} \\ + \frac{K_E}{T_R T_E} + \frac{K_E}{T_E T_q} + \frac{K_A K_F}{T_A T_E T_F} \end{array}\right) \quad ,$$

$$a_2 = \frac{-C3\,K_A}{T_A T_E T_R T_d} - \frac{X_{df}}{T_d}\left(\begin{array}{l} \frac{K_E}{T_A T_E T_F} + \frac{K_E}{T_A T_E T_q} + \frac{K_E}{T_E T_F T_q} + \frac{K_E}{T_A T_E T_R} + \frac{K_E}{T_E T_F T_R} \\ + \frac{K_E}{T_E T_q T_R} + \frac{K_A K_F}{T_A T_E T_F T_q} + \frac{K_A K_F}{T_A T_E T_F T_R} + \frac{1}{T_q T_A T_F} + \frac{1}{T_F T_A T_R} \\ + \frac{1}{T_q T_A T_R} + \frac{1}{T_q T_F T_R} \end{array}\right) \quad ,$$

$$a_1 = \frac{-C3\,K_A}{T_A T_E T_R T_d}\left(\frac{1}{T_q} + \frac{1}{T_F}\right) - \frac{X_{df}}{T_d}\left(\begin{array}{l} \frac{K_E}{T_A T_E T_F T_q} + \frac{K_E}{T_A T_E T_F T_R} + \frac{K_E}{T_E T_A T_q T_R} + \frac{K_E}{T_F T_E T_q T_R} \\ + \frac{1}{T_A T_q T_F T_R} + \frac{K_A K_F}{T_A T_E T_F T_q T_R} \end{array}\right) \quad ,$$

$$a_0 = \frac{-C3\,K_A}{T_A T_E T_R T_d T_q T_R} - \frac{X_{df} K_E}{T_A T_E T_R T_d T_q T_R} \quad ,$$

$$b_5 = \frac{1}{T_A} + \frac{1}{T_d} + \frac{K_E}{T_E} + \frac{1}{T_F} + \frac{1}{T_q} + \frac{1}{T_R} \quad ,$$

$$b_4 = \begin{pmatrix} \frac{1}{T_A T_d} + \frac{K_E}{T_A T_E} + \frac{1}{T_A T_F} + \frac{1}{T_A T_q} + \frac{1}{T_A T_R} + \frac{K_E}{T_d T_E} + \frac{1}{T_d T_F} + \frac{1}{T_d T_q} + \frac{1}{T_d T_R} \\ + \frac{K_E}{T_E T_F} + \frac{K_E}{T_E T_q} + \frac{K_E}{T_E T_R} + \frac{1}{T_F T_q} + \frac{1}{T_F T_R} + \frac{1}{T_q T_R} + \frac{K_A K_F}{T_A T_E T_F} \end{pmatrix} \quad ,$$

$$b_3 = \begin{pmatrix} \frac{K_E}{T_A T_d T_E} + \frac{1}{T_A T_d T_F} + \frac{1}{T_A T_d T_q} + \frac{1}{T_A T_d T_R} + \frac{K_E}{T_A T_E T_F} + \frac{K_E}{T_A T_E T_q} + \frac{K_E}{T_A T_E T_R} \\ + \frac{1}{T_A T_F T_q} + \frac{1}{T_A T_F T_R} + \frac{1}{T_A T_q T_R} + \frac{K_E}{T_d T_E T_F} + \frac{K_E}{T_d T_E T_q} + \frac{K_E}{T_d T_E T_R} + \frac{1}{T_d T_F T_q} \\ + \frac{1}{T_d T_F T_R} + \frac{1}{T_d T_q T_R} + \frac{K_E}{T_E T_F T_q} + \frac{K_E}{T_E T_F T_R} + \frac{K_E}{T_E T_q T_R} + \frac{1}{T_F T_q T_R} \\ + \frac{K_A K_F}{T_A T_E T_F} \left( \frac{1}{T_d} + \frac{1}{T_q} + \frac{1}{T_R} \right) \end{pmatrix} \quad ,$$

$$b_2 = \begin{pmatrix} \frac{K_E}{T_A T_d T_E T_F} + \frac{K_E}{T_A T_d T_E T_q} + \frac{K_E}{T_A T_d T_E T_R} + \frac{1}{T_A T_d T_F T_q} + \frac{1}{T_A T_d T_F T_R} + \\ \frac{1}{T_A T_d T_q T_R} + \frac{K_E}{T_A T_E T_F T_q} + \frac{K_E}{T_A T_E T_F T_R} + \frac{K_E}{T_A T_E T_q T_R} + \frac{1}{T_A T_F T_q T_R} + \\ \frac{K_E}{T_d T_E T_F T_q} + \frac{K_E}{T_d T_E T_F T_R} + \frac{K_E}{T_d T_E T_q T_R} + \frac{1}{T_d T_F T_q T_R} + \frac{K_E}{T_E T_F T_q T_R} + \\ + \frac{K_A V_{qn}}{T_A T_d T_E T_R} + \frac{K_A K_F}{T_A T_E T_F} \left( \frac{1}{T_d T_q} + \frac{1}{T_q T_R} + \frac{1}{T_d T_R} \right) \end{pmatrix} \quad ,$$

$$b_1 = \left( \begin{array}{l} \dfrac{K_E}{T_A T_d T_E T_F T_q} + \dfrac{K_E}{T_A T_d T_E T_F T_R} + \dfrac{K_E}{T_A T_d T_E T_q T_R} + \dfrac{1}{T_A T_d T_F T_q T_R} + \\ \dfrac{K_E}{T_A T_E T_F T_q T_R} + \dfrac{K_E}{T_d T_E T_F T_q T_R} + \dfrac{K_A K_F}{T_A T_E T_F T_d T_q T_R} + \\ \dfrac{K_A V_{qn}}{T_A T_d T_E T_R} \left( \dfrac{1}{T_F} + \dfrac{1}{T_q} \right) \end{array} \right) ,$$

$$b_0 = \frac{K_E}{T_A T_d T_E T_F T_q T_R} + \frac{K_A V_{qn}}{T_A T_d T_E T_F T_q T_R} ,$$

$$c_2 = \frac{-C4\, K_A}{T_A T_d T_E T_R} ,$$

$$c_1 = \frac{-C4\, K_A}{T_A T_d T_E T_R} \left( \frac{1}{T_F} + \frac{1}{T_q} \right) - \frac{K_A V_{dn} X_{qf}}{T_A T_d T_E T_q T_R} ,$$

$$c_o = \frac{-C4\, K_A}{T_A T_d T_E T_R T_F T_q} - \frac{K_A V_{dn} X_{qf}}{T_A T_d T_E T_F T_q T_R} ,$$

where $V_{dn} = \dfrac{V_{do}}{V_{to}}$ and $X_{qf} = X_q - X_q^{'}$ .

# APPENDIX C

# PROGRAM LISTING

## C.1. Main Program

```
%=====================================================
% System Identification of a generator and excitation system.
%
% Set up and declare initial variables
%=====================================================
rand('normal');          % set random generator to gaussian normal
innerloop=3              % number of iterations per time step.
for tttt=1:100            % loop for number of data runs
tttt                      % display data run number
%=====================================================
% Initialize matrices for the identification of the machine parameters
%  1 = Ed axis equation   2 = Eq axis equation   3 = Shaft dynamics equation
%=====================================================
P1=P3=zeros(3,3);
P2=zeros(20,20);
Q1=Q3=eye(3,3);
Q2=eye(20,20);
%=====================================================
% Initialize matrices for the identification of the error parameters
%=====================================================
R1=R2=R3=zeros(num_s,num_s);
RR1=RR2=RR3=eye(num_s,num_s);
%=====================================================
% Declare and clear theta, phi, psi and gk
%=====================================================
theta1=theta3=zeros(3,1);
theta2=zeros(20,1);
phi1=phi3=zeros(3,1);
phi2=zeros(20,1);
```

```
psi1=psi2=psi3=zeros(num_s,1);
gk1=gk2=gk3=zeros(num_s,1);
%=================================================
% set flags used in routine
%=================================================
flag1=flag2=flag3=flagr1=flagr2=flagr3=0;
%=================================================
% input the data and convert to Ed,Eq,Id,Iq, omega, and initial conditions
%=================================================
if tttt==1
  [TT,U]=loaddata(2);    % Load in all data at during first pass through algorithm
  ss=size(TT);
  si=ss(1,1);
  sii=ss(1,2);
end
%=================================================
% Add error to the measurements
%=================================================
rrr=.01*rand(si,sii);        % Create random error matrix
T=TT.*(1+rrr);
Eq=T(:,1);
Ed=T(:,2);
Id=T(:,3);
Iq=T(:,4);
omega=T(:,5);
Pm=T(:,6);
%==================================================
% Assign values to constants
%==================================================
c3=U(1,1);
c4=U(2,1);
vdn=U(3,1);
vqn=U(4,1);
xqf=U(5,1);
xdf=U(6,1);
delta=1/12;      % Declare time step of one twelth of a second
% ================================================
% set initial values for the coefficients
%=================================================
[theta1,theta2,theta3,init,CONST]=initval(U);
Uqo=CONST(1,1);
UE=CONST(2,1);
%=================================================
```

```
% Determine the Jacobian and the multipliers
%=====================================================
J=zeros(15,7);    % initialize Jacobian with zeros
II=eye(7);
dif=1e-2;
F = coeff(init,Uqo,UE,xdf,xqf,c3,c4,vdn,vqn);
for ll=1:7                                    % Perform Numerical differentiation
J(1:15,ll) = (coeff(init+dif*II(:,ll),Uqo,UE,xdf,xqf,c3,c4,vdn,vqn)-F)./dif;
end
%=====================================================
% initialize counters
%=====================================================
k1=k2=k3=kr1=kr2=kr3=0;
%=====================================================
% generalized least squares
%=====================================================
psi1t=psi1;
psi2t=psi2;
psi3t=psi3;
num_s=3;          % size of autoregressive error model
%
for k=1:145       % loop through number of data points
%
for kk=1:innerloop;      % inner loop
  %=================================================
  % filter input and output values
  %=================================================
  phi1=bphi1(Iq,Ed,psi1t,k,num_s);
  phi2=bphi2(Id,Iq,Eq,psi2t,k,num_s);
  phi3=bphi3(Pm,omega,psi3t,k,num_s);
  temp1=bout(Ed,psi1t,k,num_s);
  temp2=bout(Eq,psi2t,k,num_s);
  temp3=bout(omega,psi3t,k,num_s);
  %==================================================
  % Identify coefficients in transfer function for Ed vs Iq
  %==================================================
   if (flag1==0)
    [P1t,Q1t,theta1t,inc]=minnorm(P1,Q1,theta1,phi1,temp1);
    if kk==innerloop
      k1=k1+inc;
      P1=P1t;
      Q1=Q1t;
      theta1=theta1t;
```

```
    end;
  else
    [P1t,theta1t]=gls(P1,theta1,phi1,temp1);
    if kk==innerloop
      P1=P1t;
      theta1=theta1t;
    end
  end
  %==================================================
  % Determine value of error and update gk1
  %==================================================
  ek1=Ed(k+1,1)-theta1t(1,1)*Iq(k+1,1)-theta1t(2,1)*Iq(k,1)+theta1t(3,1)*Ed(k,1);
  for rr=1:num_s
  if k-rr>0
  gk1(rr,1)=Ed(k+1-rr,1)-theta1t(1,1)*Iq(k+1-rr,1)-theta1t(2,1)*Iq(k-rr,1)
                                                  +theta1t(3,1)*Ed(k-rr,1);
  gk1(rr,1)=-1*gk1(rr,1);
  elseif k-rr+1>0
  gk1(rr,1)=Ed(k+1-rr,1)-theta1t(1,1)*Iq(k+1-rr,1);
  gk1(rr,1)=-1*gk1(rr,1);
  else
  gk1(rr,1)=0;
  end
  end
  %==================================================
  % Determine parameter of error model
  %==================================================
  if (flagr1==0)
  [R1t,RR1t,psi1t,inc]=minnorm(R1,RR1,psi1,gk1,ek1);
  if kk==innerloop
  kr1=kr1+inc;
  R1=R1t;
  RR1=RR1t;
  psi1=psi1t;
  end
else
  [R1t,psi1t]=gls(R1,psi1,gk1,ek1);
  if kk==innerloop
  R1=R1t;
  psi1=psi1t;
  end
end
    %==================================================
```

```
    % Identify coefficients of equations relating Eq to Id and Iq
    %==================================================
if (flag2==0)
  [P2t,Q2t,theta2t,inc]=minnorm(P2,Q2,theta2,phi2,temp2);
  if kk==innerloop
  k2=k2+inc;
  P2=P2t;
  Q2=Q2t;
  theta2=theta2t;
  end
else
  [P2t,theta2t]=gls(P2,theta2,phi2,temp2);
  if kk==innerloop
  P2=P2t;
  theta2=theta2t;
  end
end
  %==================================================
  % Identify resulting error value
  %==================================================
  sum=theta2t(1,1)*Id(k+1,1)+theta2t(8,1)*Iq(k+1,1);
  if k>5
  for l=1:6
   sum=sum+theta2t(l+1,1)*Id(k+1-l,1)+theta2t(l+8,1)*Iq(k+1-l,1)
                                    -theta2t(l+14,1)*Eq(k+1-l,1);
  end
  else
  for l=1:k
   sum=sum+theta2t(l+1,1)*Id(k+1-l,1)+theta2t(l+8,1)*Iq(k+1-l,1)
                                    -theta2t(l+14,1)*Eq(k+1-l,1);
  end
  end
  ek2=Eq(k+1,1)-sum;
  for rr=1:num_s
  sum=0;
  if (k-rr+1)>0
  sum=theta2t(1,1)*Id(k+1,1)+theta2t(8,1)*Iq(k+1,1);
  if (k-rr)>0
  if k-rr>5
  for l=1:6
   sum=sum+theta2t(l+1,1)*Id(k+1-l,1)+theta2t(l+8,1)*Iq(k+1-l,1)
                                    -theta2t(l+14,1)*Eq(k+1-l,1);
  end
```

```
   else
   for l=1:k-rr
     sum=sum+theta2t(l+1,1)*Id(k+1-l,1)+theta2t(l+8,1)*Iq(k+1-l,1)-
theta2t(l+14,1)*Eq(k+1-l,1);
   end
   end
   end
   gk2(rr,1)=Eq(k+1-rr,1)-sum;
   else
   gk2(rr,1)=0;
   end
   end
   gk2=-1*gk2;
   %=====================================================
   % Identify error parameters
   %=====================================================
if (flagr2==0)
   [R2t,RR2t,psi2t,inc]=minnorm(R2,RR2,psi2,gk2,ek2);
   if kk==innerloop
   kr2=kr2+inc;
   R2=R2t;
   RR2=RR2t;
   psi2=psi2t;
   end
else
   [R2t,psi2t]=gls(R2,psi2,gk2,ek2);
   if kk==innerloop
   R2=R2t;
   psi2=psi2t;
   end
end
%=====================================================
% Identify coefficients of transfer function relating omega to Pm
%=====================================================
if (flag3==0)
   [P3t,Q3t,theta3t,inc]=minnorm(P3,Q3,theta3,phi3,temp3);
   if kk==innerloop
   k3=k3+inc;
   P3=P3t;
   Q3=Q3t;
   theta3=theta3t;
   end
else
```

```
   [P3t,theta3t]=gls(P3,theta3,phi3,temp3);
   if kk==innerloop
   P3=P3t;
   theta3=theta3t;
   end
end
   %=====================================================
   % Determine resulting error value
   %=====================================================
    ek3=omega(k+1,1)-theta3t(1,1)*Pm(k+1,1)-theta3t(2,1)*Pm(k,1)
                                          +theta3t(3,1)*omega(k,1);
   for rr=1:num_s
   if k-rr>0
   gk3(rr,1)=omega(k+1-rr,1)-theta3t(1,1)*Pm(k+1-rr,1)-theta3t(2,1)*Pm(k-rr,1)
                                          +theta3t(3,1)*omega(k-rr,1);
   gk3(rr,1)=-1*gk3(rr,1);
   elseif k-rr+1>0
   gk3(rr,1)=omega(k+1-rr,1)-theta3t(1,1)*Pm(k+1-rr,1);
   gk3(rr,1)=-1*gk3(rr,1);
   else
   gk3(rr,1)=0;
  end
  end
   %=====================================================
   % Identify error parameters
   %=====================================================
if (flagr3==0)
   [R3t,RR3t,psi3t,inc]=minnorm(R3,RR3,psi3,gk3,ek3);
   if kk==innerloop
   kr3=kr3+inc;
   R3=R3t;
   RR3=RR3t;
   psi3=psi3t;
   end
else
   [R3t,psi3t]=gls(R3,psi3,gk3,ek3);
   if kk==innerloop
   R3=R3t;
   psi3=psi3t;
   end
end
%=====================================================
% Check if enough data has been taken to switch from minnorm to regular
```

```
% generalized least squares solution
%=====================================================
if (k1==3)
  flag1=1;
end
if (kr1==3)
 flagr1=1;
end
if (kr2==3)
 flagr2=1;
end
if (k2==20)
 flag2=1;
end
if (k3==3)
 flag3=1;
end
if (kr3==3)
 flagr3=1;
end
%=====================================================
% Transform estimated coefficients from the z to the s domain
% using the bilinear approximation
%=====================================================
%=====================================================
% Ed versus Iq
%=====================================================
fd0=(1+theta1(3,1))/(1-theta1(3,1))*2/delta;
fn1=(theta1(1,1)-theta1(2,1))/(1-theta1(3,1));
fn0=(theta1(1,1)+theta1(2,1))/(1-theta1(3,1))*2/delta;
% ====================================================
% Omega versus Pm
% ====================================================
td0=(1+theta3(3,1))/(1-theta3(3,1))*2/delta;
tn1=(theta3(1,1)-theta3(2,1))/(1-theta3(3,1));
tn0=(theta3(1,1)+theta3(2,1))/(1-theta3(3,1))*2/delta;
% ====================================================
% Eq versus Id and Iq
% ====================================================
a=theta2(15,1);
b=theta2(16,1);
c=theta2(17,1);
d=theta2(18,1);
```

```
e=theta2(19,1);
f=theta2(20,1);
tmult=(1-a+b-c+d-e+f)*delta^6;
d5=(12-8*a+4*b-4*d+8*e-12*f)*delta^5/tmult;
d4=(60-20*a-4*b+12*c-4*d-20*e+60*f)*delta^4/tmult;
d3=(160-32*b+32*d-160*f)*delta^3/tmult;
d2=(240+80*a-16*b-48*c-16*d+80*e+240*f)*delta^2/tmult;
d1=(192+128*a+64*b-64*d-128*e-192*f)*delta/tmult;
d0=(1+a+b+c+d+e+f)*64/tmult;
g=theta2(1,1);
a=theta2(2,1);
b=theta2(3,1);
c=theta2(4,1);
d=theta2(5,1);
e=theta2(6,1);
f=theta2(7,1);
n16=(g-a+b-c+d-e+f)*delta^6/tmult;
n15=(12*g-8*a+4*b-4*d+8*e-12*f)*delta^5/tmult;
n14=(60*g-20*a-4*b+12*c-4*d-20*e+60*f)*delta^4/tmult;
n13=(160*g-32*b+32*d-160*f)*delta^3/tmult;
n12=(240*g+80*a-16*b-48*c-16*d+80*e+240*f)*delta^2/tmult;
n11=(192*g+128*a+64*b-64*d-128*e-192*f)*delta/tmult;
n10=(g+a+b+c+d+e+f)*64/tmult;
g=theta2(8,1);
a=theta2(9,1);
b=theta2(10,1);
c=theta2(11,1);
d=theta2(12,1);
e=theta2(13,1);
f=theta2(14,1);
n26=(g-a+b-c+d-e+f)*delta^6/tmult;
n25=(12*g-8*a+4*b-4*d+8*e-12*f)*delta^5/tmult;
n24=(60*g-20*a-4*b+12*c-4*d-20*e+60*f)*delta^4/tmult;
n23=(160*g-32*b+32*d-160*f)*delta^3/tmult;
n22=(240*g+80*a-16*b-48*c-16*d+80*e+240*f)*delta^2/tmult;
n21=(192*g+128*a+64*b-64*d-128*e-192*f)*delta/tmult;
n20=(g+a+b+c+d+e+f)*64/tmult;
output=[n15 n14 n13 n12 n11 n10 n22 n21 n20 d5 d4 d3 d2 d1 d0]';
%=====================================================
% Solve for new parameters.  J (delta init) = (delta z)
%=====================================================
initt=init+J\(output-F);
% ====================================================
```

```
% output the results
% ==================================================
end
end
fprintf('output.dat','------------- iteration %g ---------------------\n',120);
fprintf('output.dat','Tqo - %g \n',1/fd0);
fprintf('output.dat','Tdo - %g \n',1/initt(1,1));
fprintf('output.dat','TE  - %g \n',1/UE);
fprintf('output.dat','TF  - %g \n',1/initt(2,1));
fprintf('output.dat','TA  - %g \n',1/initt(3,1));
fprintf('output.dat','TR  - %g \n',1/initt(4,1));
fprintf('output.dat','KE  - %g \n',initt(5,1));
fprintf('output.dat','KF  - %g \n',initt(6,1));
fprintf('output.dat','KA  - %g \n',initt(7,1));
fprintf('output.dat','Tj  - %g \n',-1/tn0);
fprintf('output.dat','D   - %g \n\n',-td0/tn0);
final=[final;num_s,tttt,1/fd0,1/initt(1,1),1/UE,1/initt(2,1),1/initt(3,1),1/initt(4,1),initt(5,1),
                initt(6,1),initt(7,1), -1/tn0,-td0/tn0];
end
end
save final3.dat final /ascii /double;
```

## C.2. Subroutine to Load the Data

```
function [T,U]=loaddata(temp)      % start of routine, with T being the data values and U
                                   % as the constants vector
load voltmag.dat                   % read voltage magnitude data file
load voltang.dat                   % read voltage angle data file
load curmag.dat                    % read current magnitude data file
load curang.dat                    % read current angle data file
load speed.dat                     % read omega data file
load constpar.dat                  % read machine constants data file
load ang.dat                       % read rotor angle data file
%==================================================
% Assign machine constants
%==================================================
res=constpar(1,1);
xds=constpar(2,1);
xqs=constpar(3,1);
```

```
xdds=constpar(4,1);
xqqs=constpar(5,1);
xqf=xqs-xqqs;
xdf=xds-xdds;

%=====================================================
% Determine d and q axis voltage and current values for initial data point
%=====================================================
vm=voltmag(1,1);
va=voltang(1,1);
cm=curmag(1,1);
ca=curang(1,1);
deltanew=ang(1,1);
vdo=vm*sin((deltanew-va)*pi/180.0);
vqo=vm*cos((deltanew-va)*pi/180.0);
vto=vm;
ido=cm*sin((deltanew-ca)*pi/180.0);
iqo=cm*cos((deltanew-ca)*pi/180.0);
speedo=speed(1,1);
omega(1,1)=0;
eqo=vqo+iqo*res+ido*xdds;
edo=vdo+ido*res-iqo*xqqs;
pmo=vm*cm*cos((va-ca)*pi/180);
% ========================================================
% Determine incremental changes for the rest of the data values
% ========================================================
for k=2:151
vm=voltmag(k,1);
va=voltang(k,1);
cm=curmag(k,1);
ca=curang(k,1);
deltanew=ang(k,1);
dvd=vm*sin((deltanew-va)*pi/180.0)-vdo;
dvq=vm*cos((deltanew-va)*pi/180.0)-vqo;
Id(k-1,1)=cm*sin((deltanew-ca)*pi/180.0)-ido;
Iq(k-1,1)=cm*cos((deltanew-ca)*pi/180.0)-iqo;
Eq(k-1,1)=dvq+Iq(k-1,1)*res+Id(k-1,1)*xdds;
Ed(k-1,1)=dvd+Id(k-1,1)*res-Iq(k-1,1)*xqqs;
Pm(k-1,1)=vm*cm*cos((va-ca)*pi/180)-pmo;
omega(k-1,1)=(speed(k,1)-speedo)*360;
end
vdn=vdo/vto;
vqn=vqo/vto;
```

```
c3=-vdn*res-vqn*xdds;
c4=-vqn*res+vdn*xqqs;
%==============================================
% initialize return values before leaving routine
%==============================================
T=[Eq Ed Id Iq omega Pm];
U=[c3;c4;vdn;vqn;xqf;xdf];
% End of Routine
```

## C.3. Subroutine for Calculation of Initial Values of Parameter Vectors:

```
function [t1,t2,t3,init,CONST]=initval(U)   % start of routine
c3=U(1,1);      % Extract constants
c4=U(2,1);
vdn=U(3,1);
vqn=U(4,1);
xqf=U(5,1);
xdf=U(6,1);
delta=1/12;
init(5,1)=1.05;  % KE   % initialize Parameters with a slight error away from actual
init(6,1)=.085;  % KF
init(7,1)=26;  % KA
init(1,1)=1/4.8;   % 1/Td
CONST(2,1)=1/.05;   % 1/TE
init(2,1)=1/.38;   % 1/TF
init(3,1)=1/.18;   % 1/TA
init(4,1)=1/.015;   % 1/TR
Uqo=1/.2;          % 1/Tq
CONST(1,1)=Uqo;
%==================================================================
% Determine Coefficients s-domain transfer functions
%==================================================================
F = coeff(init,Uqo,CONST(2,1),xdf,xqf,c3,c4,vdn,vqn);
Tj=4643.7/60/180;
D=.6;
%==================================================================
% Transform s-domain coefficients to z-domain coefficients with bilinear transformation
%==================================================================
Tran=[1 1 1 1 1 1 1;-6 -4 -2 0 2 4 6;15 5 -1 -3 -1 5 15; -20 0 4 0 -4 0 20];
```

```
Tran=[Tran;15 -5 -1 3 -1 -5 15;-6 4 -2 0 2 -4 6; 1 -1 1 -1 1 -1 1];
tdel=2/delta;
mat1=[1*tdel^6;
F(10,1)*tdel^5;F(11,1)*tdel^4;F(12,1)*tdel^3;F(13,1)*tdel^2;F(14,1)*tdel;F(15,1)];
mat2=[0; F(1,1)*tdel^5;F(2,1)*tdel^4;F(3,1)*tdel^3;F(4,1)*tdel^2;F(5,1)*tdel;F(6,1)];
mat3=[0; 0;0;0;F(7,1)*tdel^2;F(8,1)*tdel;F(9,1)];
tmat1=Tran*mat1;
tmat2=Tran*mat2;
tmat3=Tran*mat3;
tmat2=tmat2/tmat1(1,1);
tmat3=tmat3/tmat1(1,1);
tmat1=tmat1/tmat1(1,1);
%=========================================================
% Setup vectors for returning to the main routine
%=========================================================
t2=[tmat2;tmat3;tmat1(2:7,1)];
t1(1,1)=xqf/(Uqo+tdel)*Uqo;
t1(2,1)=t1(1,1);
t1(3,1)=(Uqo-tdel)/(Uqo+tdel);
t3(1,1)=-1/(D/Tj+tdel)/Tj;
t3(2,1)=t3(1,1);
t3(3,1)=(D/Tj-tdel)/(D/Tj+tdel);

% End of Routine
```

## C.4. Subroutine to Determine s-domain Coefficients for q-axis Equation

```
function F = coeff(init,Uqo,UE,XDF,XQF,c3,c4,Vdn,Vqn)
a=Uqo;    % initialize constants and parameters
b=XDF;
c=XQF;
d=c3;
e=c4;
f=init(5,1);  % KE
g=init(6,1);  % KF
h=init(7,1);  % KA
i=init(1,1);   % 1/Td
j=UE;   % 1/TE
k=init(2,1);   % 1/TF
```

```
l=init(3,1);   % 1/TA
m=init(4,1);   % 1/TR
n=Vqn;
o=Vdn;
%==========================================================
% Calculate coefficients from equations found from B(sI-A)^-1C
%==========================================================
F(1,1)=-b*i;
F(2,1)=-a*b*i-b*f*i*j-b*i*k-b*i*l-b*i*m;
F(3,1)=-a*b*f*i*j-a*b*i*k-b*f*i*j*k-a*b*i*l-b*f*i*j*l-b*i*k*l-b*g*h*i*j*k*l-a*b*i*m
       -b*f*i*j*m-b*i*k*m-b*i*l*m;
F(4,1)=-a*b*f*i*j*k-a*b*f*i*j*l-a*b*i*k*l-b*f*i*j*k*l-a*b*g*h*i*j*k*l-a*b*f*i*j*m
       -a*b*i*k*m-b*f*i*j*k*m-a*b*i*l*m-b*f*i*j*l*m-d*h*i*j*l*m-b*i*k*l*m
       -b*g*h*i*j*k*l*m;
F(5,1)=-a*b*f*i*j*k*l-a*b*f*i*j*k*m-a*b*f*i*j*l*m-a*d*h*i*j*l*m-a*b*i*k*l*m
       -b*f*i*j*k*l*m-d*h*i*j*k*l*m-a*b*g*h*i*j*k*l*m;
F(6,1)=-a*b*f*i*j*k*l*m-a*d*h*i*j*k*l*m;
F(7,1)=-e*h*i*j*l*m;
F(8,1)=-a*e*h*i*j*l*m-e*h*i*j*k*l*m-a*c*h*i*j*l*m*o;
F(9,1)=-a*e*h*i*j*k*l*m-a*c*h*i*j*k*l*m*o;
F(10,1)=a+i+f*j+k+l+m;
F(11,1)=a*i+a*f*j+f*i*j+a*k+i*k+f*j*k+a*l+i*l+f*j*l+k*l+g*h*j*k*l+a*m+i*m+f*j*m
        +k*m+l*m;
F(12,1)=a*f*i*j+a*i*k+a*f*j*k+f*i*j*k+a*i*l+a*f*j*l+f*i*j*l+a*k*l+i*k*l+f*j*k*l
        +a*g*h*j*k*l+g*h*i*j*k*l+a*i*m+a*f*j*m+f*i*j*m+a*k*m+i*k*m+f*j*k*m
        +a*l*m+i*l*m+f*j*l*m+k*l*m+g*h*j*k*l*m;
F(13,1)=a*f*i*j*k+a*f*i*j*l+a*i*k*l+a*f*j*k*l+f*i*j*k*l+a*g*h*i*j*k*l+a*f*i*j*m
        +a*i*k*m+a*f*j*k*m+f*i*j*k*m+a*i*l*m+a*f*j*l*m+f*i*j*l*m+a*k*l*m
        +i*k*l*m+f*j*k*l*m+a*g*h*j*k*l*m+g*h*i*j*k*l*m+h*i*j*l*m*n;
F(14,1)=a*f*i*j*k*l+a*f*i*j*k*m+a*f*i*j*l*m+a*i*k*l*m+a*f*j*k*l*m+f*i*j*k*l*m
        +a*g*h*i*j*k*l*m+a*h*i*j*l*m*n+h*i*j*k*l*m*n;
F(15,1)=a*f*i*j*k*l*m+a*h*i*j*k*l*m*n;
```

## C.5. Subroutine for Filtering Input for Ed Equation

```
function phi = bphi1(Iq,Ed,psi1,k,s)
phi(1,1)=Iq(k+1,1);
for i=1:s
if (k+1-i)>0
```

```
phi(1,1)=phi(1,1)+psi1(i,1)*Iq(k+1-i,1);
end
end
if (k>0)
phi(2,1)=Iq(k,1);
phi(3,1)=-1*Ed(k,1);
for i=1:s
if (k-i)>0
phi(2,1)=phi(2,1)+psi1(i,1)*Iq(k-i,1);
phi(3,1)=phi(3,1)-psi1(i,1)*Ed(k-i,1);
end
end
end
```

## C.6. Subroutine for Filtering Eq Axis Inputs

```
function phi = bphi2(Id,Iq,Eq,psi2,k,s)
phi=zeros(20,1);
phi(1,1)=Id(k+1,1);
phi(8,1)=Iq(k+1,1);
for i=1:s
if (k+1-i)>0
phi(1,1)=phi(1,1)+psi2(i,1)*Id(k+1-i,1);
phi(8,1)=phi(8,1)+psi2(i,1)*Iq(k+1-i,1);
end
end
for j=1:6
if (k+1-j>0)
phi(2-1+j,1)=Id(k+1-j,1);
phi(9-1+j,1)=Iq(k+1-j,1);
phi(15-1+j,1)=-Eq(k+1-j,1);
for i=1:s
if (k+1-j-i)>0
phi(2-1+j,1)=phi(2-1+j,1)+psi2(i,1)*Id(k+1-j-i,1);
phi(9-1+j,1)=phi(9-1+j,1)+psi2(i,1)*Iq(k+1-j-i,1);
phi(15-1+j,1)=phi(15-1+j,1)-psi2(i,1)*Eq(k+1-j-i,1);
end
end
end
```

```
end
```

## C.7. Subroutine for Filtering Inputs for Shaft Dynamics Equation

```
function phi = bphi3(Pm,omega,psi3,k,s)
phi(1,1)=Pm(k+1,1);
for i=1:s
if (k+1-i)>0
phi(1,1)=phi(1,1)+psi3(i,1)*Pm(k+1-i,1);
end
end
if (k>0)
phi(2,1)=Pm(k,1);
phi(3,1)=-1*omega(k,1);
for i=1:s
if (k-i)>0
phi(2,1)=phi(2,1)+psi3(i,1)*Pm(k-i,1);
phi(3,1)=phi(3,1)-psi3(i,1)*omega(k-i,1);
end
end
end
```

## C.8. Subroutine for Filtering Outputs for the Three Equations

```
function t=bout(out,psi,k,s)
t=out(k+1,1);
for i=1:s
if (k+1-i)>0
t=t+psi(i,1)*out(k+1-i);
end
end
```

## C.9. Subroutine for Minnorm Identification

```
%==========================================================
%
function [PP,QQ,THE,inc]=minnorm(p,q,th,phi,y)
tmp1=q*phi;
tmp2=p*phi;
temp=(phi'*tmp1);
if temp~=0
inc=1;
THE=th+tmp1*(y-phi'*th)/temp;
QQ=q-tmp1*tmp1'/temp;
PP=p-(tmp2*tmp1'+tmp1*tmp2')/temp+(tmp1*tmp1'*(1+phi'*tmp2))/temp/temp;
else
inc=0;
PP=p;
QQ=q;
THE=th;
end
```

## C.10. Subroutine to Carry Out Generalized Least Squares Equations

```
function [PP,THE]=gls(p,th,phi,y)
tmp1=p*phi;
temp=1+phi'*tmp1;
THE=th+tmp1*(y-phi'*th)/temp;
PP=p-tmp1*tmp1'/temp;
```

## C.11. Datafile Containing Constant Parameters Used for the Machine

```
2e-5        % resistance
1.463       % Xd
1.407       % Xq
.222        % Xd'
.349        % Xq'
```

# Vita

David Barber was born on May 28, 1969 in Fairfax County, Virginia. He graduated high school in 1987 after having spent his senior year at the Thomas Jefferson High School for Science and Technology. He graduated summa cum laude from Virginia Polytechnic Institute and State University(VPI) with a Bachelor of Science degree in Electrical Engineering in December 1991. While in undergraduate, he participated in the cooperative education program where he worked three semesters at General Electric Drive Systems in Salem, Virginia. In Jan 1992, he started a masters program in Electrical Engineering at VPI. He worked as a graduate teaching assistant for one semester and then as a graduate research assistant in the area of power systems engineering for a year. He completed his master's course work in May 1993. He started work with General Public Utilities Service Corporation as an engineer in June 1993.

David C Barber