POWER SYSTEMS ANALYSIS
ON
PROGRAMMABLE CALCULATORS
by
Michael Allen Walker

Thesis submitted to the Graduate 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:

L. L. Grigsby
Dr. L. L. Grigsby

T. E. Bechert
Dr. T. E. Bechert

August, 1974
Blacksburg, Virginia
ACKNOWLEDGMENTS

The author wishes to express great appreciation to Dr. L. L. Grigsby and the Energy Research Group for the financial support provided during the past fifteen months of graduate study. Dr. Grigsby has provided the inspiration and the extra push, when needed, to enable this thesis to be written. His many suggestions, comments, and most of all, his moral support have been an immeasurable contribution in this work. To Dr. Grigsby simple thanks are not enough.

Much thanks goes to Juanita Marlett for her time and never ending patience during the typing of this thesis. Her's is a job well done.

Special thanks is extended to Ed Dove and Tony Payne; fellow roommates and above all, friends, whose many ideas, suggestions and edifications were of astronomical value in the work and writing of this thesis.

Most of all, the author wishes to express his deepest appreciation to four people, without whom this thesis would not have been possible. Their financial and moral support, patiences and understanding for the past year and a half have been a tremendous aid to the author. If some day even a tenth of what they have given can be returned, then they will get some idea of their contribution to this work. To his parents and two brothers, the author is forever indebted.
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.1</td>
<td>Generator Representation At Bus i</td>
<td>22</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Norton's Equivalent Circuit of Figure 3.2.1</td>
<td>22</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Generator No. 2 Removed From System, Back On-line At t = 0.1 sec.</td>
<td>39</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Generator No. 1 Removed From System, Back On-line At t = 0.1 sec.</td>
<td>40</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Plot of Solution By 4th Order Runge-Kutta And Solution By State Transition Using Same Δt</td>
<td>42</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Plot of Solution By 4th Order Runge-Kutta With Δt = 0.01 And Plot of Solution By State Transition With Δt = 0.0025</td>
<td>44</td>
</tr>
</tbody>
</table>
**LIST OF TABLES**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.1</td>
<td>Load Flow Comparisons Using Three Bus System of Elgerd</td>
<td>35</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Load Flow Comparisons Using Five Bus System of Stagg and El-Abiad</td>
<td>36</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Load Flow Comparisons Using Five Bus System of Stevenson</td>
<td>37</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Incremental and Total Solution Times of 4th Order Runge-Kutta and State Transition</td>
<td>43</td>
</tr>
</tbody>
</table>
# Table of Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Definition of the Problem</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Approach</td>
<td>2</td>
</tr>
<tr>
<td>II. Methods of Load Flow</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Criteria Selection</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Newton-Raphson Load Flow</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Decoupled Newton-Raphson Load Flow</td>
<td>6</td>
</tr>
<tr>
<td>2.4 Y-Bus Gauss-Seidel</td>
<td>8</td>
</tr>
<tr>
<td>2.5 Aitken's ( \Delta^2 ) Process</td>
<td>10</td>
</tr>
<tr>
<td>2.6 Z-bus Gauss-Seidel With Respect to Ground</td>
<td>12</td>
</tr>
<tr>
<td>2.7 Z-bus Gauss-Seidel With Respect to the Slack Bus</td>
<td>16</td>
</tr>
<tr>
<td>2.8 Summary</td>
<td>19</td>
</tr>
<tr>
<td>III. Transient Stability Analysis</td>
<td>20</td>
</tr>
<tr>
<td>3.1 Introduction to Stability</td>
<td>20</td>
</tr>
<tr>
<td>3.2 System Performance Equations</td>
<td>20</td>
</tr>
<tr>
<td>3.3 Swing Equations</td>
<td>24</td>
</tr>
<tr>
<td>3.4 Solution of the Swing Equations By State Transition</td>
<td>25</td>
</tr>
<tr>
<td>3.5 Solution of the Swing Equations By Fourth-Order Runge-Kutta</td>
<td>28</td>
</tr>
<tr>
<td>3.6 Summary</td>
<td>32</td>
</tr>
<tr>
<td>Chapter</td>
<td>Page</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>IV. RESULTS AND CONCLUSIONS</td>
<td>33</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>33</td>
</tr>
<tr>
<td>4.2 Comparison of Load Flow Techniques</td>
<td>33</td>
</tr>
<tr>
<td>4.3 Application of Transient Stability Analysis</td>
<td>38</td>
</tr>
<tr>
<td>4.4 Comparison of Methods of Solving the Swing Equations</td>
<td>41</td>
</tr>
<tr>
<td>4.5 Conclusions and Suggestions for Further Work</td>
<td>45</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>47</td>
</tr>
<tr>
<td>VITA</td>
<td>49</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

Of the more than 3600 utilities in the United States, about 100 are producing about 90 percent of the total power [7]. Since these few companies generate the vast majority of the electrical power and support the majority of the research work being done in this area, one could only expect that this research be large-systems oriented. Because of the size of the companies and the trend toward more interconnections, load flow and other types of analysis programs capable of handling networks of several thousand busses are necessary. These large and complicated programs make necessary the availability of a large computer, such as the IBM 370. However, not all of the analyses performed by a major utility are performed on the entire system. There are instances when only a small section of the entire layout must be analysed. Also there are departments within a big company whose work requires computer aid, but not the full capacity of a large computer. In addition, the planning and operation of the approximately 3500 small utilities requires the analysis of networks on the order of ten to fifty busses. These needs for power system analyses of modest size networks provide an impetus for a study of the applicability of minicomputers and programmable calculators. The continuing trend towards reduced price and increased capability for these small digital machines provides more motivation for a study of this type.

The subject of this theseis is the development and implementation of a power systems analysis package for use on minicomputers and programmable calculators. An attempt will be made to define a set of criteria for use
in picking the most 'optimal' approach of analysis in order to keep down program size and be capable of working a moderate size system. All of the work presented in this thesis relating to a computer was performed on a Hewlett-Packard 9830A programmable calculator.

1.2 APPROACH

The approach taken is to define a set of criteria for selecting a load flow technique. Given this set of criteria, a review and comparison of various methods of load flow will be made. Then attention will be given to a review of the stability problem and a discussion of two methods of solving the differential swing equations. Afterwards the selected load flow and stability techniques will be combined to form an analysis package. At this point conclusions will be drawn and ideas for further work in this area will be suggested.
2.1 CRITERIA SELECTION

When power systems analysis is mentioned, the first type of analysis most often brought to mind is that of load flow. The load flow study determines the voltage, current, power, and power factor at various points in an electric network. Other types of analysis, such as economic dispatch, fault analysis, and stability analysis, require the knowledge of these factors in advance. Because of the dependence of the other types of analysis on the predetermined values of voltage, current, power, etc., the load flow analysis could be considered the heart of a power systems analysis package. Therefore, the selection of criteria has for the most part been focused on load flow.

Since the work is being done on a programmable calculator, two of the most important aspects would be the size and complexity of the load flow program itself. The larger and more complex a program, the larger the amount of space required to take care of array storage and complicated logic. The more space taken by the program itself, the less the size of a system that can be solved.

Other criteria to be considered are the convergence quality of the load flow method and the speed with which results are obtained. These two criteria go hand in hand because if the iterating technique converges in only a few iterations, then the required time is going to be reasonably small. Let it be pointed out that speed on a programmable calculator and speed on a large computer are in no way comparable. What can be done on a large computer, such as the IBM 370, in a matter of
4 microseconds and seconds takes seconds and minutes on a programmable calculator. Thus, if one wants to do on-line analysis the larger computer is needed.

Since the purpose of this theseis is to develop a power systems package, a criterion which must be included is that of ease of implementation of the load flow technique into other forms of analysis. To include all other forms of analysis would require many, many hours of research and programming. As was stated in Chapter 1, only a stability analysis program will be included. Therefore, this criterion can be restated as the ease of implementation of the load flow technique into a stability analysis program.

In summary, the set of criteria to be used in determining a 'best' load flow technique for the analysis package is the following:

1. Size and complexity of the program.
2. Convergence quality of the method.
3. Relative time for solution.
4. Ease of implementation of the method into stability analysis.

Now that the criteria have been selected a review and comparison of the various load flow methods with respect to these criteria must be made.

2.2 Newton-Raphson Load Flow

The one method of load flow analysis most often used on large-scale systems is that of Newton-Raphson. Only a brief discussion of the method will be presented in this thesis. For a more detailed approach see references [13], [17], [18].

With Newton-Raphson load flow a new guess of bus voltages is found by updating the previous guess with incremental values.
\[
\begin{align*}
\mathbf{e}^{k+1} &= \mathbf{e}^k + \Delta \mathbf{e}^k \\
\mathbf{f}^{k+1} &= \mathbf{f}^k + \Delta \mathbf{f}^k \\
\begin{bmatrix}
\Delta \mathbf{e}^k \\
\Delta \mathbf{f}^k
\end{bmatrix} &= J^{-1} \begin{bmatrix}
\Delta \mathbf{P}^k \\
\Delta \mathbf{Q}^k
\end{bmatrix} 
\end{align*}
\] (2.2-1) (2.2-2)

where

- \( \mathbf{e}^k \) = vector of real bus voltages after the \( k \)th iteration
- \( \mathbf{f}^k \) = vector of imaginary bus voltages after the \( k \)th iteration
- \( \Delta \mathbf{e}^k \) = vector of incremental changes in the \( k \)th iterate of real bus voltages
- \( \Delta \mathbf{f}^k \) = vector of incremental changes in the \( k \)th iterate of imaginary bus voltages
- \( \Delta \mathbf{P}^k \) = vector of differences in scheduled bus real powers and calculated real powers for the \( k \)th iterate of bus voltages
- \( \Delta \mathbf{Q}^k \) = vector of differences in scheduled bus reactive powers and calculated reactive powers for the \( k \)th iterate of bus voltages
- \( J \) = \( 2(n-1) \times 2(n-1) \) Jacobian matrix where \( n \) is the size of the system.

This method of analysis has the characteristic of quadratic coverage and thus a solution is obtained in only a few iterations. The bus admittance matrix, which can be easily obtained, is used with this method.
The Newton-Raphson method, however, is a very complicated one, requiring difficult programming and a considerable amount of array storage. The Jacobian matrix is, as previously stated, a $2(n-1) \times 2(n-1)$ matrix for an $n$-bus system. This matrix itself requires a noticeable amount of space and logic to program, and because the entries are dependent upon the bus voltages they have to be recalculated after each iteration. Also, as can be seen by Equation (2.2-2), the inverse of the Jacobian must be found after each re-evaluation, which means solving a set of $2(n-1)$ simultaneous equations. A user programmed algorithm to solve these simultaneous equations would require much space and time. If a built-in matrix inversion routine is available, as is the case with the Hewlett-Packard 9830, the extra space would not be required, however this routine, too, takes time to solve $2(n-1)$ simultaneous equations.

Because of the complexity of the Newton-Raphson program and the many disadvantages of the Jacobian matrix due to the use of a programmable calculator, the decision was made to omit this method of load flow from any further consideration for use in an analysis package.

2.3 DECOUPLED NEWTON-RAPHSON LOAD FLOW

Work has been done in the area of decoupling the Newton-Raphson load flow problem, the most notable work being that of Brian Stott [14]. In his paper, Stott has mathematically decoupled the bus voltage angle and magnitude calculations. In its polar form, the Newton Raphson algorithm becomes

$$
\begin{bmatrix}
\Delta P^k \\
\Delta Q^k
\end{bmatrix} =
\begin{bmatrix}
H^k & N^k \\
J^k & L^k
\end{bmatrix}
\begin{bmatrix}
\Delta \theta^k \\
\Delta V^k
\end{bmatrix}
$$

(2.3-3)
where $\Delta P_k^k$ and $\Delta Q_k^k$ are previously defined and

$$\Delta f_k^k = \text{vector of incremental change in bus voltage angles after k iterations}$$

$$f_k^k = \text{vector of incremental change in bus voltage magnitudes after k iterations}$$

$H_k^k, N_k^k, L_k^k, U_k^k = \text{submatrices of the Jacobian matrix.}$

Neglecting the effects of the bus voltage magnitudes on the reactive powers and the effects of the bus voltage angles on the real powers leaves two separate sets of equations

$$\Delta P_k^k = H_k^k \Delta f_k^k. \quad (2.3-4)$$

$$\Delta Q_k^k = L_k^k \Delta f_k^k. \quad (2.3-5)$$

Such a separation would mean that (2.3-4) and (2.3-5) could be solved independently, reducing storage and computation time.

When implementing this decoupled algorithm, Stott found it to be weak in convergence and not competitive with Newton-Raphson or any of the other accepted load flow methods, therefore he replaced Equation (2.3-5) with the polar current mismatch.

$$\Delta I_k^k = D_k^k \Delta f_k^k. \quad (2.3-6)$$

This method was given considerable attention in this thesis work and every attempt was made to implement the algorithm, however, it was found to be unsuccessful on every test system to which it was applied. Stott and Alsac have since revised Stott's original work in the area of decoupling [16]. This more recent method has been programmed and tested by E. L. Dove [6]. Dove's preliminary findings are in agreement with the claims of the paper.
2.4 Y-BUS GAUSS-SEIDEL

The Gauss-Seidel iterative technique, like Newton-Raphson, most often uses the bus admittance matrix in its formulation of the load flow problem. This method takes the system power equations \( \vec{S} = \vec{V} \vec{I}^* \) and manipulates them into recursive type equations by expressing the bus currents in terms of the bus voltages and the bus admittance matrix. For a more detailed approach see Stagg & El-Abiad[13], Stevenson [14], and Elgerd [7].

The power at any bus \( k \) in the electrical network can be expressed as

\[
\vec{S}_k = P_k + jQ_k = \vec{V}_k \vec{I}_k^* \quad k = 1,2,...,n \quad (2.4-1)
\]

where \( \vec{S}_k \) is the complex power at the bus, \( P_k \) is the real power, \( Q_k \) is the reactive power, and \( \vec{V}_k \) and \( \vec{I}_k \) are the bus voltage and current, respectively. The system is also governed by the matrix equation

\[
\vec{I}_{bus} = \vec{Y}_{bus} \vec{V}_{bus} \quad (2.4-2)
\]

where

\[
\begin{align*}
\vec{I}_{bus} & = \text{vector of bus currents} \\
\vec{V}_{bus} & = \text{vector of bus voltages} \\
\vec{Y}_{bus} & = \text{system admittance matrix.}
\end{align*}
\]

Solving Equation (2.4-1) for \( \vec{I}_k \), removing the \( k \)th equation from (2.4-2) and setting the two equal gives

\[
\frac{P_k - jQ_k}{\vec{V}_k^*} = \sum_{i=1}^{n} \vec{Y}_{ki} \vec{V}_i \quad k = 1,2,...,n. \quad (2.4-3)
\]
Equation (2.4-3) can then be rearranged into a recursive equation for all buses except the slack bus, whose generated powers are unknown. The recursive equation is

\[ \hat{V}_k = \frac{1}{\hat{y}_{kk}} \left( \frac{p_k - jq_k}{\hat{V}_k^*} - \sum_{i=1}^{n} \hat{y}_{ki} \hat{V}_i \right) \]

\[ k = 2, \ldots, n \]  

where \( \hat{y}_{ki} \) is the \( k^{th} \) row, \( j^{th} \) column element of the system impedance matrix and all other variables are the same as previously defined.

The above formulation of the Y-bus Gauss-Seidel load flow problem does not include the capability of handling voltage controlled busses. At a voltage controlled bus, the voltage magnitude rather than the generated reactive power is specified. Before the real and imaginary components of voltage at such a bus are determined, a new value of reactive power at that bus is found.

\[ Q_k = -\text{Im} \left\{ \left( \hat{y}_{kk} \hat{V}_k + \sum_{i=1}^{n} \hat{y}_{ki} \hat{V}_i \right) \hat{V}_k^* \right\} \]  

(2.4-5)

where \( \text{Im} \) means 'imaginary part of' and all variables are as previously defined [14].

The Y-bus Gauss-Seidel technique is an easy one to program and therefore uses a relatively small amount of calculator space with its logic. The fact that the logic of the program is straightforward means each iteration is fast. This method of analysis is also advantageous in the fact that it uses the bus admittance matrix, which, as has been stated, is easily found.
Not everything about Y-bus Gauss-Seidel is quick and easy. The iterations of bus voltages are found quickly, but the convergence of the method is slow (around the order of 1.6), which means many iterations are needed to reach convergence. To say that the convergence of the method is on the order of 1.6 simply means that

\[
\lim_{n \to \infty} \frac{e_n^{1.6}}{e_{n+1}} = \text{constant}
\]  

(2.4-6)

where \( e_n \) is the difference between the true solution and the \( n \)th iteration value, and similarly \( e_{n+1} \) is the difference between the true solution and the \((n+1)\)th iteration value.

The criterion of ease of implementation into stability analysis has yet to be discussed. Discussion of this criterion will be considered in the next chapter when the stability problem is presented.

2.5 AITKEN'S \( \Delta^2 \) PROCESS

The convergence quality of the Y-bus Gauss-Seidel method can be improved by the use of accelerating factors. With accelerating factors the new voltage value is found by updating the previous value with an adjusted increment. That is

\[
\tilde{V}^{\text{new}} = \tilde{V}^{\text{old}} + \alpha \Delta \tilde{V}
\]  

(2.5-1)

where \( \tilde{V}^{\text{old}} \) = previous guess of bus voltage

\( \Delta \tilde{V} \) = calculated incremental change in bus voltage

\( \alpha \) = scalar (or complex) accelerating factor.

The use of accelerating factors improves the convergence and therefore reduces the number of iterations required, but unfortunately the optimal
α is different for every system. Rather than try to find the best α for each individual system studied, it would be more convenient if a process were included in the load flow program that picked the optimal acceleration factor for any given system. A method that chooses a near optimal accelerating factor is a mathematical technique called Aitken's $\Delta^2$ process.

Let $[X_n]_{n=1}^{\infty}$ be any sequence and define a new sequence $[X'_n]_{n=1}^{\infty}$ with

$$X'_n = X_n - \frac{(\Delta X_n)^2}{\Delta^2 X_n}$$

(2.5-2)

where

$$\Delta X_n = X_{n+1} - X_n$$

(2.5-3)

$$\Delta^2 X_n = \Delta X_{n+1} - \Delta X_n.$$  

(2.5-4)

If the sequence $[X_n]_{n=1}^{\infty}$ is converging to a point S such that the error after the $n^{th}$ iteration, $e_n = X_n - S$, is not equal to zero and the error after the $(n+1)^{th}$ iteration, $e_{n+1}$, can be expressed as

$$e_{n+1} = (B + \beta_n) e_n$$

(2.5-5)

where $|B| < 1$ and $\lim_{n \to \infty} \beta_n = 0$, then $X'_n$ goes to S faster than $X_n$, in the sense that

$$\lim_{n \to \infty} \frac{X'_n - S}{X_n - S} = 0.$$  

(2.5-6)

This technique is discussed in detail by Conte and deBoor [5].

Putting this accelerating technique in terms of bus voltages and simplifying the expression gives

$$V'_n = \frac{\bar{V}_n \bar{V}_{n+2} - \bar{V}^2_{n+1}}{\bar{V}_{n+2} - 2\bar{V}_{n+1} + \bar{V}_n}$$

(2.5-7)
where

\[ V_n = \text{bus voltage after } n \text{ iterations} \]
\[ V_{n+1} = \text{bus voltage after } n+1 \text{ iterations} \]
\[ V_{n+2} = \text{bus voltage after } n+2 \text{ iterations}. \]

Although this technique is well known in numerical analysis, to the author's knowledge this is its first application to load flow. The results of this testing will be discussed in a later chapter along with all other applications of methods under consideration.

2.6 **Z-BUS GAUSS-SEIDEL WITH RESPECT TO GROUND**

The Gauss-Seidel technique is most often used with the topology of the system being described by the bus admittance matrix, however the method can also be applied using the bus impedance matrix, Z-bus, to describe the system. Such an approach is called a Z-bus Gauss-Seidel load flow. The Z-bus Gauss-Seidel load flow problem can be formulated with respect to one of two reference points, these being ground and the slack bus. Both methods will be discussed in this chapter, with the formulation of the problem with respect to ground being presented in this section and the formulation with respect to the slack bus coming in the next section.

In Y-bus Gauss-Seidel, the injected bus currents are expressed in terms of the bus voltages as

\[ \vec{I}_{bus} = \vec{V}_{bus} \cdot \vec{V}_{bus} . \]  \hspace{1cm} (2.4-2)

If this equation is rearranged to give an expression for the bus voltages in terms of the bus currents, it becomes
The voltage at any bus \( k \) in the \( n \)-bus system can then be expressed as

\[
\bar{V}_k = \sum_{j=1}^{n} \bar{z}_{kj} \bar{I}_j .
\]  

(2.6-3)

To formulate a recursive equation, the bus currents need to be expressed in terms of the bus voltages by an equation other than (2.4-2). Such an expression is found from the system power equation

\[
\bar{S}_k = \bar{P}_k + j \bar{Q}_k = \bar{V}_k \bar{I}_k^* \quad k = 1, 2, \ldots, n.
\]  

(2.4-1)

Solving (2.4-1) for \( \bar{I}_k \) gives

\[
\bar{I}_k = \frac{\bar{P}_k - j \bar{Q}_k}{\bar{V}_k^*} .
\]  

(2.6-4)

Note that this equation for the injected bus current is in terms of the bus voltage and the scheduled power (real and reactive) at bus \( k \). The generated powers, and therefore the scheduled powers, at the slack bus (usually the first bus in the system) are not specified, so this equation cannot be used to find the slack bus injected current. Since the voltage at this bus is known, (2.6-3) can be used to express the slack bus current in terms of the slack bus voltage and the other bus currents, to give

\[
\bar{I}_1 = \frac{1}{\bar{z}_{11}} \left[ \bar{V}_1 - \sum_{j=2}^{n} \bar{z}_{1j} \bar{I}_j \right] .
\]  

(2.6-5)
Substituting this expression back into (2.6-3) gives

\[ \tilde{V}_k = \sum_{j=2}^{n} \tilde{z}_{kj} \bar{I}_j + \frac{\tilde{z}_{kl}}{\tilde{z}_{11}} \left[ \tilde{V}_1 - \sum_{j=2}^{n} \tilde{z}_{lj} \bar{I}_j \right] \quad k = 2,3,\ldots,n. \]  

(2.6-6)

Note that the two summations are over the same variable, therefore the expression can be simplified to give

\[ \tilde{V}_k = \sum_{j=2}^{n} \left( \tilde{A}_{kj} \bar{I}_j \right) + \frac{\tilde{z}_{kl}}{\tilde{z}_{11}} \tilde{V}_1 \quad k = 2,\ldots,n \]  

(2.6-7)

where

\[ \tilde{A}_{kj} = \tilde{z}_{kj} - \frac{\tilde{z}_{kl}}{\tilde{z}_{11}} \tilde{z}_{lj}. \]  

(2.6-8)

Now if (2.6-4) is substituted for \( \bar{I}_j \), the following recursive equation is obtained

\[ \tilde{V}^{(t+1)}_k = \sum_{j=2}^{n} \left( \tilde{A}_{kj} \frac{P_j - jQ_j}{\tilde{V}^{(t)}_j} \right) + \frac{\tilde{z}_{kl}}{\tilde{z}_{11}} \tilde{V}_1 \quad k = 2,\ldots,n \]  

(2.6-9)

When used in Gauss-Seidel this equation takes the form

\[ \tilde{V}^{(t+1)}_k = \sum_{j=2}^{n} \left( \tilde{A}_{kj} \frac{P_j - jQ_j}{\tilde{V}^{(t)}_j} \right) + \sum_{j=k}^{n} \left( \tilde{A}_{kj} \frac{P_j - jQ_j}{\tilde{V}^{(t)}_j} \right) + \frac{\tilde{z}_{kl}}{\tilde{z}_{11}} \tilde{V}_1 \quad k = 2,\ldots,n \]  

(2.6-10)

where

\[ \tilde{V}^{(t)}_k = t^{\text{th}} \text{ iterate of voltage at bus } k \]

\[ \tilde{V}^{(t+1)}_k = t+1^{\text{th}} \text{ iterate of voltage at bus } k. \]

In the above formulation, specification of all real and reactive bus powers (excluding the slack bus) was assumed. If the system under consideration contains a voltage controlled bus, such an assumption is
invalid. At a voltage controlled bus the voltage magnitude and the real power are given and the reactive generation is left unspecified. To perform a load flow analysis on a system with a voltage controlled bus, an equation for the scheduled reactive power in terms of what is known in the system, must be found.

To obtain an expression for the reactive power at a voltage controlled bus (2.6-7) is used. Suppose that bus k is a voltage controlled bus;

Solving (2.6-7 for $I_k$ gives

\[ \bar{I}_k = \frac{1}{\bar{A}_{kk}} \left\{ \bar{V}_k - \left[ \frac{\bar{z}_{kl}}{\bar{Z}_{11}} \bar{V}_l + \sum_{j=2}^{n} \bar{A}_{kj} \bar{I}_j \right] \right\}. \quad (2.6-11) \]

From (2.6-4)

\[ Q_k = -\text{Im} \left[ \bar{V}_k^* \bar{I}_k \right], \quad (2.6-12) \]

therefore

\[ Q_k = -\text{Im} \left\{ \frac{\bar{V}_k^*}{\bar{A}_{kk}} \left[ \bar{V}_k - \left( \frac{\bar{z}_{kl}}{\bar{Z}_{11}} \bar{V}_l + \sum_{j=2}^{n} \bar{A}_{kj} \bar{I}_j \right) \right] \right\}. \quad (2.6-13) \]

where \text{Im} is defined in Section 2.4 and all variables are previously defined.

Like Y-bus Gauss-Seidel, Z-bus Gauss-Seidel is relatively easy to program, therefore keeping space required for logic close to a minimum. The time per iteration for this method is longer than that of Y-bus Gauss-Seidel, however convergence is usually obtained in a much smaller number of iterations. The use of the bus impedance matrix makes the implementation of this method of load flow into a stability analysis an easy task, as shall be seen in the discussion of this criterion in the next chapter.
Unlike the bus admittance matrix, the bus impedance matrix is not easily obtained. A somewhat complicated algorithm is needed to calculate this matrix, and therefore much more time is required than when obtaining the bus admittance matrix. Let it be pointed out here that the time required to obtain Z-bus or Y-bus, whichever the case, is not of great importance. The matrix can be found initially and stored for further use. The only time its elements will change is when changes are made in the system topology.

One major disadvantage of Z-bus Gauss-Seidel with the impedance matrix grown with respect to grown is that a simplified system that neglects shunt admittances, such as Stevenson's 5-bus system [14], cannot be worked by such a method because there are no lines connecting the system with the reference point. A further discussion of this problem will be carried out in Chapter 4, when example systems shall be listed along with the disadvantages and advantages of each method of load flow.

2.7 Z-BUS GAUSS-SEIDEL WITH RESPECT TO THE SLACK BUS

As was pointed out in Section 2.6, the impedance matrix approach to Gauss-Seidel load flow can be formulated with respect to one of two references. Ground can be used as reference to solve the load flow problem, as was formulated in the previous section, or, if the necessary information is retained separately from the impedance matrix, the slack bus can be used as reference. In the latter formulation, the impedance matrix is grown using the slack bus as reference while all shunt lines from the system busses to ground are stored in a separate vector.
With the shunt connections being treated as current sources the injected current at any bus \( k \) in the system can be expressed as

\[
\bar{I}_k = \frac{P_k - jQ_k}{\tilde{Y}_k} - \tilde{Y}_k \bar{V}_k
\]  \hspace{1cm} (2.7-1)

where

\[
\tilde{Y}_k = \text{shunt admittance at bus } k
\]

and all other variables are as previously defined. A new estimate of bus voltages can be obtained from the network equation

\[
\tilde{V}_\text{bus} = \tilde{z}_\text{bus} \bar{I}_\text{bus} + \tilde{V}_s
\]  \hspace{1cm} (2.7-2)

where

\[
\tilde{V}_\text{bus} = \text{(n-1)-vector of bus voltages for an n-bus system with the slack bus being omitted}
\]

\[
\bar{I}_\text{bus} = \text{(n-1)-vector of injected bus currents}
\]

\[
\tilde{V}_s = \text{(n-1)-vector with all entries equal to slack bus voltage}
\]

\[
\tilde{z}_\text{bus} = \text{(n-1)x(n-1) impedance matrix grown with respect to the slack bus.}
\]

Note that the injected bus current at the slack bus is not needed because of the selection of that bus as the reference point.

If bus 1 is taken as the slack bus, the Gauss-Seidel form of (2.7-2) becomes

\[
\tilde{V}_k^{(k+1)} = \tilde{V}_1 + \sum_{j=2}^{k-1} \tilde{z}_{kj} \bar{I}_j^{(k+1)} + \sum_{j=k}^{n} \tilde{z}_{kj} \bar{I}_j^{(k)}
\]

\[
k = 2, 3, \ldots, n
\]  \hspace{1cm} (2.7-3)
where

\[
\bar{I}_k^{(k+1)} = \frac{p_k - jq_k}{V_k^*(k+1)} - \bar{V}_k \bar{V}_k^{(k+1)}.
\]

(2.7-4)

For a more clear and complete discussion of the formulation of this method see Chapter 8 of Stagg and El-Abiad [13].

As with Y-bus Gauss-Seidel and Z-bus Gauss-Seidel with respect to ground, if a voltage controlled bus is included in the system under study an expression for the reactive power must be obtained. Solving the \(k\)th equation of (2.7-2) for the current \(\bar{I}_k\) gives

\[
\bar{I}_k = \frac{1}{\bar{z}_{kk}} \left[ \bar{V}_k - \bar{V}_1 - \sum_{j=2}^{n} \bar{z}_{kj} \bar{I}_j \right].
\]

(2.7-5)

Substituting (2.7-1) for the left hand side of (2.7-5) and solving for \(q_k\) gives

\[
q_k = -\text{Im} \left\{ \bar{V}_k^* \left[ \frac{1}{\bar{z}_{kk}} \left( \bar{V}_k - \bar{V}_1 - \sum_{j=2}^{n} \bar{z}_{kj} \bar{I}_j \right) + \bar{Y}_k \bar{V}_k \right] \right\}.
\]

(2.7-6)

This equation for the reactive power at the voltage controlled bus \(k\) must be reevaluated prior to each reevaluation of the voltage at the bus.

Like Gauss-Seidel using the Z-bus matrix grown with respect to ground, this method usually converges in only a few iterations. Because this method of load flow analysis employs the bus impedance matrix grown with respect to the slack bus, any type of system can be handled, including those with no shunt admittances to ground.

The big disadvantage of using the impedance matrix grown with respect to the slack bus is found when a stability study is attempted using this
representation of the system topology. Because of the way in which Z-bus was formed, no information other than the voltage value is known about the slack bus. There is no way of knowing what happens to this bus when a fault is applied elsewhere in the system. This disadvantage will be presented more clearly in the next chapter when implementation of the load flow techniques into a stability analysis is discussed.

2.8 SUMMARY

In summary of the chapter, attention has been given to six forms of load flow analysis with two methods being dropped from further consideration in this thesis. Newton-Raphson load flow was considered to be too complicated, too large and too time consuming for use on a programmable calculator. The method of decoupling, first introduced by Stott, was found to be nonsatisfactory due to the fact that it proved unsuccessful in all attempts made. This leaves the last four techniques discussed in this chapter to be implemented and compared according to the set of criteria previously determined. Results of such a comparison will be discussed in Chapter 4 of this thesis.
CHAPTER III
TRANSIENT STABILITY ANALYSIS

3.1 INTRODUCTION TO STABILITY

Although it is preferred, a power system does not always operate at an equilibrium point. When a system is operating in an undesired mode, a transient stability analysis is needed so that necessary information can be obtained. Such an analysis "... provides information related to the capability of a power system to remain in synchronism during major disturbances resulting from either the loss of generating or transmission facilities, sudden or sustained load changes, or momentary faults." [13]

When a stability study is performed, a load flow analysis is needed in advance so as to know the condition of the system prior to the application of a fault. With the data acquired from the load flow solution and the system generator data, the network impedance or admittance matrix is updated to include equivalent load impedances (or admittances) to ground and the transient impedances of the machines. After the impedance or admittance matrix has been modified to include all of these element representations, the fault is applied. The machine swing equations are now solved by one of several methods to obtain the new positions and speeds of the rotors. Each time this is done, the fault voltages, a product of the system performance equations, have to be recalculated because of their dependence upon the equivalent currents of the network, which are dependent upon the positions of the machines rotors.

3.2 SYSTEM PERFORMANCE EQUATIONS

At any generator bus in a power system the machine can be represented as a voltage source with a series impedance as shown in Figure 3.2.1, where
\( \bar{V}_i \) = voltage at the \( i^{th} \) bus

\( \bar{E}_i \) = emf of the \( i^{th} \) generator

\( \bar{Z}_{d_i} \) = transient impedance of the \( i^{th} \) generator.

The magnitude and angle of the generator emf, \( \bar{E}_i \), is not known. However, with the transient impedance being given and the bus voltage and generated real and reactive powers at bus \( i \) being found from a load flow analysis, the emf can be calculated as

\[
\bar{E}_i = \bar{V}_i + \bar{Z}_{d_i} \frac{P_{Ti} - jQ_{Ti}}{\bar{V}_i^*},
\]

(3.2-1)

where

\( P_{Ti} \) = real power of the turbine at generator \( i \).

\( Q_{Ti} \) = reactive power of the turbine at generator \( i \).

By use of Norton's Theorem, the generator emf and transient impedance can be represented by a current source in parallel with the transient impedance. Such an equivalent circuit for Figure 3.2.1 is seen in Figure 3.2.2, with

\[
\bar{I}_{eq_i} = \frac{\bar{E}_i}{\bar{Z}_{d_i}}.
\]

(3.2-2)

If the \( i^{th} \) bus is not a generator bus then \( \bar{E}_i = 0 \) and therefore \( \bar{I}_{eq_i} = 0 \).

With all of the system generators now being treated as equivalent current sources, the bus voltages can be found by use of the system performance equation

\[
\bar{V}_{bus} = \bar{Z}_{bus \ (fault)} \bar{I}_{eq}
\]

(3.2-3)
Figure 3.2.1. Generator Representation at Bus $i$.  

Figure 3.2.2. Norton's Equivalent Circuit of Figure 3.2.1.
or

\[ I_{eq} = \bar{Z}_{bus} \text{(fault)} \bar{V}_{bus}, \quad (3.2-4) \]

where the bus impedance or admittance matrix has been updated as stated in Section 3.1.

From Equations (3.2-3) and (3.2-4) it can be seen that there is an advantage of using the bus impedance matrix rather than the bus admittance matrix in a stability analysis formulation. If \( \bar{Z}_{bus} \) is employed, finding the new bus voltages is a simple matter of matrix multiplication. However, if \( \bar{Y}_{bus} \) is used, either the inverse of the admittance matrix must be found or an iterating procedure used in order to obtain the bus voltages.

Either of the two ways of finding the voltages with \( \bar{Y}_{bus} \) being used takes up time. If an inverse is found, then only matrix multiplication is needed until the system topology is changed at which time the bus admittance matrix is updated and a new inverse obtained. If (3.2-4) is solved iteratively to find \( \bar{V}_{bus} \), the solution time will be longer than when finding an inverse, because the voltages must be updated at each increment of time, using the latest positions of the machine rotors.

When using the bus impedance matrix in the stability formulation, modification of \( \bar{Z}_{bus} \), due to changes in the network topology, are handled by use of the algorithm for growing \( \bar{Z}_{bus} \). When the impedance matrix is used for stability studies ground is usually taken as reference because all bus voltages, except at the faulted bus, change during the transient period [4]. When the slack bus is taken as reference, there is no way of knowing what happens to the slack bus when a fault is applied elsewhere in the network.
3.3 SWING EQUATIONS

The operating characteristics of the system generators are described by sets of differential equations. Depending upon the necessary details of a machine, the number of differential equations needed to describe the machine will vary. The simplest representation is with two first-order differential equations. For a further discussion and derivation of these equations see Elgerd [7], Grigsby [8] and Stagg and El-Abiad [13].

The differential equation describing the motion of the machine rotor is called the swing equation. This equation in per unit form is

\[ P_{Ti} - P_{Gi} = \frac{H_i}{\pi f^2} \frac{d^2 \delta_i}{dt^2} + D_i \frac{d \delta_i}{dt} \]  \hspace{1cm} (3.3-1)

where

- \( P_{Ti} \) = turbine power at bus \( i \)
- \( P_{Gi} \) = generator power at bus \( i \)
- \( H_i \) = per unit inertia constant of machine at bus \( i \)
- \( D_i \) = per unit damping constant of machine at bus \( i \)
- \( \delta_i \) = angle of the rotor of the machine at bus \( i \)
- \( f^0 \) = nominal operating frequency
- \( \pi = 3.14159 \.

Substituting \( \omega_i = \frac{d \delta_i}{dt} \) into (3.3-1) and solving for \( \frac{d^2 \delta_i}{dt^2} \) gives the following set of differential equations

\[ \frac{d \delta_i}{dt} = \omega_i \]  \hspace{1cm} (3.3-2a)
where

\[ \dot{\omega}_i = \frac{-D_i}{M_i} \omega_i + \frac{1}{M_i} (P_{Ti} - P_{Gi}) \quad (3.3-2b) \]

\[ Wi = \text{speed of the rotor on the machine at bus } i \]

\[ M_i = H_i/\pi f^o. \]

The solution to this set of differential equations can be found by any of several methods, such as Euler's method, state transition, predictor-corrector, or Runge-Kutta. In this thesis the methods of state transition and fourth-order Runge-Kutta have been applied. In the following sections the application of these two methods is discussed.

### 3.4 SOLUTION OF THE SWING EQUATIONS BY STATE TRANSITION

Given the state Equations (3.3-2a) and (3.3-2b), these equations can be rewritten in matrix form as

\[ \dot{x} = Ax + Bu \quad (3.4-1) \]

where

\[ x = [\delta_i, \omega_i]^T \]

\[ A = \begin{bmatrix} 0 & 1 \\ -D_i/M_i & 0 \end{bmatrix} \]

\[ B = \begin{bmatrix} 0 \\ 1/M_i \end{bmatrix}^T \]

The solution to (3.4-1) can be found using state transition. That is,
\[ X[(k+1)T] = \Phi(T) X[kT] + \left( \int_0^T \Phi(T-\tau)B(\tau)d\tau \right) U[kT] \]  

(3.4-2)

where \( \Phi(T) \) is the state transition matrix [10]. This matrix can be found by any of several methods, among which are Taylor series expansion, inverse Laplace transform, Caley-Hamilton, and transfer function.

Since the system is only of order two, the use of the inverse Laplace transform technique, to give a closed form of \( \Phi(T) \), is a rather simple task. Using this method,

\[ \Phi(T) = \mathcal{L}^{-1}\left\{[sI - A]^{-1}\right\} \bigg|_{t=T} \]  

(3.4-3)

where

- \( I \) = identity matrix of order two
- \( s \) = Laplace transform variable
- \([ ]^{-1}\) = inverse of the matrix within the brackets
- \( \mathcal{L}^{-1}\{ \} \) = inverse Laplace transform operation
- \( T \) = chosen time increment.

As can be seen in (3.4-3), before \( \Phi(T) \) can be found, \([sI - A]\) must be formed and its inverse determined.

\[
\begin{bmatrix}
\frac{1}{s} & -1 \\
0 & \frac{D_i}{M_i}
\end{bmatrix}
\]

(3.4-4)

\[
\begin{bmatrix}
\frac{1}{s} & -1 \\
0 & \frac{D_i}{M_i}
\end{bmatrix}
\]

(3.4-5)
Substituting (3.4-5) into (3.4-3) and performing the appropriate operations gives

$$\phi(T) = \begin{bmatrix} 1 & \frac{M_i}{D_i} \left( 1 - \exp \left( -\frac{D_i}{M_i} T \right) \right) \\ 0 & \exp \left( -\frac{D_i}{M_i} T \right) \end{bmatrix}.$$  

(3.4-6)

Since U(kT) is independent of X, (3.4-2) can be expressed as

$$X[(k+1)T] = \phi(T)X[kT] + \varrho(T)U[kT]$$

(3.4-7)

where

$$\varrho(T) = \int_0^T \phi(T-\tau)B(\tau)d\tau,$$  

(3.4-8)

and T is the solution time increment. Without showing the mathematical details, \(\varrho(T)\) becomes

$$\varrho(T) = \begin{bmatrix} \frac{1}{D_i} \left[ T - \frac{M_i}{D_i} \left( 1 - \exp \left( -\frac{D_i}{M_i} T \right) \right) \right] \\ \frac{1}{D_i} \left[ 1 - \exp \left( -\frac{D_i}{M_i} T \right) \right] \end{bmatrix}.$$  

(3.4-9)

If damping is neglected in a machine, (3.4-6) and (3.4-9) are not applicable, because by setting \(D_i = 0\) most of the terms of the two matrices become undefined. However, a complete new derivation is not necessary. With \(D_i = 0\), (3.4-5) becomes

$$[sI - A]^{-1} = \begin{bmatrix} 1/s & 1/s^2 \\ 0 & 1/s \end{bmatrix}.$$  

(3.4-10)
By the same procedure as before, $\dot{\phi}(T)$ and $\theta(T)$ now becomes

$$
\dot{\phi}(T) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (3.4-11)
$$

and

$$
\theta(T) = \begin{bmatrix} \frac{T^2}{2M_I} \\
\frac{T}{M_I} \\
\end{bmatrix} \quad (3.4-12)
$$

As can be seen in (3.4-11) and (3.4-12, neglect of the damping in the system machines simplifies the two matrices greatly.

The accuracy of state transition increases with a decrease in the solution time increment. With a very small time increment the solution is nearly exact, however, accuracy is not the only desired outcome for obtaining a solution. Simple programming and small storage requirements are two characteristics of this technique that also should be considered.

3.5 Solution of the Swing Equations by Fourth-Order Runge-Kutta

A well known and accepted method of solving the machine swing equations is that of Runge-Kutta. The various ways of using Runge-Kutta are all 'one step methods that do not require the evaluation of higher derivatives of the function, but they nevertheless produce solutions accurate to the same order of magnitude as those produced by using a corresponding number of terms in a Taylor-series expansion." [11]

Solving the swing equations by fourth-order Runge-Kutta is equivalent to using the first five terms of the Taylor series. This means the formulas have a truncation error proportional to $\Delta x^5$. For a complete discussion
of this method see Ley [11] and Stagg and El-Abiad [13].

When fourth-order Runge-Kutta is applied to the machine swing equations the solution takes the form

$$\delta_i(n+1) = \delta_i(n) + \frac{1}{6} (K_i + 2 K_{i+m} + 2 K_{i+2m} + K_{i+3m})$$

(3.5-1)

and

$$\omega_i(n+1) = \omega_i(n) + \frac{1}{6} (L_i + 2 L_{i+m} + 2 L_{i+2m} + L_{i+3m})$$

(3.5-2)

where

$$\delta_i(n) = \text{rotor angle of machine } i \text{ after } n \text{ time increments}$$

$$\omega_i(n) = \text{rotor speed of machine } i \text{ after } n \text{ time increments}$$

and the K's and L's are found in the following manner.

Evaluation of the K and L variables is accomplished by the following procedure.

1. $$U_i = \frac{P_{T_i}}{T_i} - \frac{|E_i|}{|z_{d_i}|} \left[ c_i \sin(\delta_i) - d_i \cos(\delta_i) \right]$$

2. $$K_i = \omega_i \Delta t$$

$$L_i = \frac{\pi f^o}{H_i} U_i \Delta t$$

3. $$c_i^{(1)} + j d_i^{(1)} = \sum_{j=1}^{m} \frac{z_{ij}}{|z_{d_i}|} \left[ \frac{|E_i|}{|z_{d_i}|} \angle (\delta_i + \frac{K_i}{2}) \right]$$

4. $$U_i^{(1)} = \frac{P_{T_i}}{T_i} - \frac{|E_i|}{|z_{d_i}|} \left[ c_i^{(1)} \sin(\delta_i + \frac{K_i}{2}) - d_i^{(1)} \cos(\delta_i + \frac{K_i}{2}) \right]$$
5. \( K_{i+m} = (\omega_i + \frac{L_i}{2}) \Delta t \)

\( L_{i+m} = \frac{\pi f^o}{H_i} u_i^{(1)} \Delta t \)

6. \( c_i^{(2)} + d_i^{(2)} = \sum_{j=1}^{m} \frac{z_{ij}}{z_{d_i}} \left[ \frac{|E_i|}{z_{d_i}} \angle \left( \delta_i + \frac{K_{i+m}}{2} \right) \right] \)

7. \( u_i^{(2)} = \frac{P_{T_i}}{|z_{d_i}'|} \left[ c_i^{(2)} \sin(\delta_i + \frac{K_{i+m}}{2}) - d_i^{(2)} \cos(\delta_i + \frac{K_{i+m}}{2}) \right] \)

8. \( K_{i+2m} = (\omega_i + \frac{L_{i+2m}}{2}) \Delta t \)

\( L_{i+2m} = \frac{\pi f^o}{H_i} u_i^{(2)} \Delta t \)

9. \( c_i^{(3)} + j d_i^{(3)} = \sum_{j=1}^{m} \frac{z_{ij}}{z_{d_i}} \left[ \frac{|E_i|}{z_{d_i}} \angle (\delta_i + K_{i+2m}) \right] \)

10. \( u_i^{(3)} = \frac{P_{T_i}}{|z_{d_i}'|} \left[ c_i^{(3)} \sin(\delta_i + K_{i+2m}) - d_i^{(3)} \cos(\delta_i + K_{i+2m}) \right] \)

11. \( K_{i+3m} = (\omega_i + L_{i+2m}) \Delta t \)

\( L_{i+3m} = \frac{\pi f^o}{H_i} u_i^{(3)} \Delta t \)

where

\( P_{T_i} = \) turbine real power at bus \( i \)

\(|E_i| = \) magnitude of the emf of machine at bus \( i \)

\( z_{d_i}' = \) transient impedance of machine at bus \( i \).
\( c_i \) = real voltage at bus \( i \)
\( d_i \) = imaginary voltage at bus \( i \)
\( \delta_i \) = rotor angle of machine at bus \( i \)
\( \omega_i \) = rotor speed of machine at bus \( i \)
\( H_i \) = per unit inertia constant of machine at bus \( i \)
\( f^o \) = nominal operating frequency
\( \pi \) = 3.14159
\( m \) = number of generator busses in the system
\( \Delta t \) = solution time increment
\( z_{ij} \) = the \( i \)th row, \( j \)th column entry of the bus impedance matrix grown with respect to ground.

As can clearly be seen, fourth-order Runge-Kutta is a difficult method to program. The logic involved in performing the eleven steps is complicated and requires a significant amount of computer space. Also, obtaining a solution for the swing equations over a somewhat lengthy time interval is a very slow process because the eleven steps, listed previously, have to be performed after each incremental time increase.

All of the above presentation and discussion gives the picture of fourth-order Runge-Kutta being an extremely troublesome method to use when solving the swing equations. However, the complicated logic and the long solution time can be tolerated if an accurate solution is desired. The method is very accurate, as previously stated, and thus can be solved with an incremental solution time of magnitude much larger than one used in a method such as state transition to obtain as accurate results.
3.6 **SUMMARY**

In this chapter, the concept of transient stability has been presented with the advantages and disadvantages of using the system impedance matrix and the system admittance matrix being pointed out. Also a discussion of solving the machine swing equations by state transition and by fourth-order Runge-Kutta has been presented. A comparison of the accuracy and solution times of these two methods will be included in the next chapter.
CHAPTER IV
RESULTS AND CONCLUSIONS

4.1 INTRODUCTION

This chapter is a compilation and comparison of results obtained in the implementation of the load flow methods, the transient stability program, and the two methods of solving the machine swing equations. These results will be used to point out the characteristics of the various techniques. Finally, conclusions will be drawn, where applicable, and ideas for further work in this area will be suggested.

4.2 COMPARISON OF LOAD FLOW TECHNIQUES

The load flow techniques were tested on Elgerd's three bus system [7], Stagg and El-Abiad's five bus system [13], and Stevenson's five bus system [14]. Tables 4.2.1, 4.2.2, and 4.2.3 are compilations of the number of iterations required, the maximum errors, approximate solution times, and appropriate core usages for each of the four methods of load flow under consideration. Since the HP9830A programmable calculator contains no clock for measuring solution times, exact times could not be obtained. The times listed in the tables were obtained by observing a wristwatch to measure the time intervals between inputs and outputs. Because of this crude method, answers are approximations to the nearest second. The maximum error for each method is defined as

\[ e_{\text{max}}^i = \max \left| V_i^{(\ell)} - V_i^{(\ell-1)} \right| \]  

(4.2-1)

where convergence is obtained after \( \ell \) iterations.
The normal Y-bus Gauss-Seidel load flow technique required the greatest number of iterations in all test cases, and for the majority of the cases, this was also the slowest method. When Aitken's $\Delta^2$ process was applied to the Y-bus Gauss-Seidel technique, the results were not uniform. When convergence was obtained, the process proved effective in decreasing the required number of iterations. However, convergence was not always achieved with the process included in the load flow algorithm, as can be seen in Table 4.2.3.

In all of the test cases, Z-bus Gauss-Seidel load flow, using the bus impedance matrix with the slack bus as its reference, was the fastest method. In this formulation, the bus impedance matrix is a $(n-1)$ x$(n-1)$ matrix for an n bus system, which is a contributing factor to the method also requiring the least amount of core. If only a load flow analysis were needed, this method would be very appealing; however, the selection of the slack bus as reference hinders any attempt to perform a transient stability analysis. The reasons for this were presented in Chapters II and III.

The Z-bus Gauss-Seidel load flow technique, employing Z-bus with ground as its reference, was in no case the fastest method used. In all cases the technique was second to the other impedance matrix load flow formulation in the minimum amount of required core. The method has the disadvantage of not being able to handle a system without shunt elements. In his example system, Stevenson [14] has assumed a short line model neglecting all shunt elements. Therefore, this example cannot be analysed by this algorithm. Use of this formulation is very advantageous in the
<table>
<thead>
<tr>
<th></th>
<th>error tolerance</th>
<th></th>
<th>error tolerance</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.001</td>
<td></td>
<td>0.0001</td>
<td></td>
</tr>
<tr>
<td></td>
<td>k</td>
<td>e&lt;sub&gt;max&lt;/sub&gt;</td>
<td>time</td>
<td>core</td>
</tr>
<tr>
<td>Y-bus Gauss-Seidel</td>
<td>5</td>
<td>0.00012</td>
<td>9</td>
<td>1205</td>
</tr>
<tr>
<td>Y-bus Gauss-Seidel with Aitken's Δ²</td>
<td>4</td>
<td>0.00011</td>
<td>9</td>
<td>1338</td>
</tr>
<tr>
<td>Z-bus Gauss-Seidel with slack bus reference</td>
<td>3</td>
<td>0.00044</td>
<td>5</td>
<td>872</td>
</tr>
<tr>
<td>Z-bus Gauss-Seidel with ground reference</td>
<td>3</td>
<td>0.00034</td>
<td>9</td>
<td>1110</td>
</tr>
</tbody>
</table>

k = number of iterations  
core = number of 16 bit, 2 byte words

Table 4.2.1. Load Flow Comparisons Using Three Bus System of Elgerd.
<table>
<thead>
<tr>
<th>Error Tolerance 0.001</th>
<th>Error Tolerance 0.0001</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>$e_{\text{max}}$</td>
</tr>
<tr>
<td>---</td>
<td>-----------------</td>
</tr>
<tr>
<td>Y-bus Gauss-Seidel</td>
<td>14</td>
</tr>
<tr>
<td>Y-bus Gauss-Seidel with Aitken's $\Delta^2$</td>
<td>7</td>
</tr>
<tr>
<td>Z-bus Gauss-Seidel with slack bus reference</td>
<td>4</td>
</tr>
<tr>
<td>Z-bus Gauss-Seidel with ground reference</td>
<td>4</td>
</tr>
</tbody>
</table>

$k = \text{number of iterations}$

$core = \text{number of 16 bit, 2 byte words}$

Table 4.2.2. Load Flow Comparisons Using Five Bus System of Stagg and El-Abiad.
Table 4.2.3. Load Flow Comparisons Using Five Bus System of Stevenson.

<table>
<thead>
<tr>
<th></th>
<th>error tolerance 0.001</th>
<th>error tolerance 0.0001</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>k</td>
<td>e_{max}</td>
</tr>
<tr>
<td>Y-bus Gauss-Seidel</td>
<td>7</td>
<td>0.00041</td>
</tr>
<tr>
<td>Y-bus Gauss-Seidel with Aitken's $\Delta^2$</td>
<td>(27)</td>
<td>(0.131)</td>
</tr>
<tr>
<td>Z-bus Gauss-Seidel with slack bus reference</td>
<td>4</td>
<td>0.00044</td>
</tr>
<tr>
<td>Z-bus Gauss-Seidel with ground reference</td>
<td>(not applicable)</td>
<td></td>
</tr>
</tbody>
</table>

k = number of iterations  
core = number of 16 bit, 2 byte words
implementation of the load flow technique into a stability study. These advantages have been pointed out in the discussion in Chapter III. As stated in this previous discussion of the matter, if the impedance matrix, with ground as reference, is to be used for a stability study, then it is usually used in the load flow program.

4.3 APPLICATION OF TRANSIENT STABILITY ANALYSIS

When performing a stability analysis, the system performance equations can be represented in impedance form, as in (3.2-3), or in admittance form as in (3.2-4). A decision to use the bus impedance matrix in the stability analysis was made as a result of the previous discussion of the matter in Section 2 of Chapter III and Section 2 of this chapter. A stability program has been written which has the capability of handling a three phase fault, or a generator removal from any bus in the system. The five bus system of Stagg and El-Abiad [13] was used as a test system for the program.

Plots of the machines responses, for various faults on the system are included in this thesis as proof of the existence of an analysis program. Figure 4.3.1 is a plot of the machines' performances with the generator at bus two being removed from the network and put back on-line after a 0.1 second duration. Figure 4.3.2 shows the machines' performances with the generator at bus one being removed and put back on line after a 0.1 second duration. The system's performance due to a three phase fault being applied at bus two and lasting for 0.09 seconds is shown in Figures 4.4.1 and 4.4.2.
Figure 4.3.1. Generator No. 2 Removed From System, Back On-line At $t = 0.1$ sec.
Figure 4.3.2. Generator No. 1 Removed From System, Back On-line At $t = 0.1$ sec.
4.4 COMPARISON OF METHODS OF SOLVING THE SWING EQUATIONS

The two methods used for solving the machine swing equations have been presented in detail in Chapter III, therefore, only a comparison of results and appropriate conclusions will be made in the section. As can be seen in the previous discussion of the methods, the state transition approach to solving these differential equations is much less complicated and easier to program than the fourth order Runge-Kutta technique. The system performance equations are solved four times for each time increment in fourth order Runge-Kutta, compared to once for state transition. Therefore, a solution time for Runge-Kutta on the order of four times that of state transition would be expected when the same solution time increment is used in both methods.

Using the same basic stability program and inserting each of the two methods as a subroutine, plots and solution times were obtained. Figure 4.4.1 is a comparison of the methods with a solution time increment for both methods of 0.01 seconds. Because of the numerous re-evaluations of the system performance equations, the method of Runge-Kutta takes much longer to obtain a solution, than does the state transition technique. Total solution times, including plotting time, are given in Table 4.4.1.

The solution time increment of the state transition method was decreased by a factor of four (Δt = 0.0025 seconds) and the corresponding solution curve was compared to a Runge-Kutta solution using Δt = 0.01 seconds. This comparison is shown in Figure 4.4.2. As can be seen by the curves in the figure, this decrease in the state transition solution
Figure 4.4.1. Plot of Solution By 4th Order Runge-Kutta And Solution By State Transition Using Same \( \Delta t \).
\[ \Delta t \quad \text{Total time} \]

<table>
<thead>
<tr>
<th>Method</th>
<th>( \Delta t )</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runge-Kutta</td>
<td>0.01</td>
<td>31.5</td>
</tr>
<tr>
<td>State Transition</td>
<td>0.01</td>
<td>9.0</td>
</tr>
<tr>
<td>State Transition</td>
<td>0.0025</td>
<td>30.0</td>
</tr>
</tbody>
</table>

\( \Delta t \) is in seconds
Total time is in minutes

Table 4.4.1
Incremental and Total Solution Times of 4th Order Runge-Kutta and State Transition.
Figure 4.4.2. Plot of Solution By 4th Order Runge-Kutta With $\Delta t = 0.01$
And Plot of Solution By State Transition With $\Delta t = 0.0025$. 

$\delta_i$ (degrees)
time increment brings the responses of the two methods closer together. From Table 4.4.1 it can be seen that even with the much smaller solution time increment, state transition is still the faster of the two methods.

4.5 CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

Four methods for solving the load flow problem and two methods for solving the machine swing equations have been developed and implemented. The results of the implementation of the load flow methods show that the two impedance matrix approaches yield a solution in the least amount of time and use the least amount of storage to perform the analysis. For a stability program, Z-bus Gauss-Seidel using the slack bus as reference cannot be applied. Because of the ease of solving the system performance equations, Gauss-Seidel employing the impedance matrix with ground as reference is the most comfortable type of load flow for use in a stability analysis. To solve the swing equations by state transition required a smaller solution time increment than Runge-Kutta, however, the logic is less complicated and the total time for solution is less.

The conclusion that Z-bus Gauss-Seidel with ground as reference is the better load flow method to use and the state transition approach is the better technique for solving the swing equations should not be considered final. These conclusions were made from the work that has been done; however, the number of approaches to the problem are in no way exhausted. Because of its speed and simplified algorithm, Stott and Alsac's Fast Decoupled Load Flow [16] looks very appealing for use on minicomputers and programmable calculators. Also, a very detailed examination of the application of Aitken's \( \Delta^2 \) process to the Y-bus Gauss-Seidel load flow technique should be performed. It is the author's
opinion that this process may be very promising for use in reducing the

time and iterations for the load flow analysis.

Work should be done to determine an upper bound on the size of a

system that can be worked on the HP9830A programmable calculator. The

AEP fourteen bus test system [1] was analyzed on the calculator by the

Y-bus Gauss-Seidel load flow program; however, due to data errors, com­

plications were encountered when analysis was attempted by the other

load flow methods. The calculator has the storage capability to handle

a system of this size on larger, depending upon the load flow method

used for analysis.

The power systems package developed in this work contains only a

load flow program and a stability program. Further work could be done

to include other types of analysis in the package.


VITA

Michael Allen Walker was born on March 2, 1951, in Danville Virginia. After receiving his primary and secondary education in Pittsylvania County, he entered Virginia Polytechnic Institute and State University in the Fall of 1969 and was graduate in June 1973 with a B.S. degree in Electrical Engineering.

Michael Allen Walker
POWER SYSTEMS ANALYSIS
ON
PROGRAMMABLE CALCULATORS
by
Michael Allen Walker

(ABSTRACT)

The objective of this thesis is to develop and implement a power systems analysis package for use on minicomputers and programmable calculators. Algorithms for four different load flow techniques are developed and tested on the HP9830A programmable calculator. The transient stability analysis problem is reviewed, with special attention being given to the solution of the system performance equations by either the bus impedance matrix approach or the bus admittance matrix approach. Also attention is focused on the solution of the machine swing equations by the state transition method and by the technique of fourth order Runge-Kutta. Comparisons are made between the different load flow methods to possibly determine the 'best' method to be used in the analysis package. Comparisons are also made between the methods of solving the swing equations in order to select a preferred technique for use in the stability program.