VLAHUTIN, Paul Andrew, 1944-
POWER SYSTEM TRANSIENT STABILITY ASSESSMENT
BY PATTERN RECOGNITION.

Case Western Reserve University, Ph.D., 1971
Engineering, electrical

University Microfilms, A XEROX Company, Ann Arbor, Michigan

POWER SYSTEM TRANSIENT STABILITY ASSESSMENT
BY PATTERN RECOGNITION

by

PAUL ANDREW VLAHUTIN

Submitted in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy

Thesis Advisor: S. K. Mitter, Ph.D.

School of Engineering
CASE WESTERN RESERVE UNIVERSITY
June 1971

CASE WESTERN RESERVE UNIVERSITY

GRADUATE STUDIES

We hereby approve the thesis of

Paul Andrew Vlahutin

candidate for the Ph.D.

degree.

Signed: A Ritter
(Chairman)

R. Schoeffler

Date 3/12/71

POWER SYSTEM TRANSIENT STABILITY ASSESSMENT BY PATTERN RECOGNITION

Abstract

by

PAUL ANDREW VLAHUTIN

A technique for approximating the transient stability boundary for electric power systems is presented. The power system is modeled as a set of synchronous machines connected by a passive network. Each machine is represented as a constant voltage behind a transient reactance. The problem is formulated as a dichotomous pattern recognition problem using a quadratic decision surface. The patterns are the postfault initial conditions and the classes are the stable and unstable ones. An automated search procedure in the system state-space is given which locates the boundary along a set of rays emanating from a postfault equilibrium point. The stability of each point is evaluated via a simulation. The resulting data set is separated by a least-squares pattern recognition algorithm.

A variant of the technique is also presented which attempts to obtain a boundary approximation for a given amount of total generation. Different allocations of generation are assigned and several postfault networks are chosen. Each point is then simulated for each allocation-network pair and is deemed stable if all solutions are stable. The rest of the technique remains as previously given.

Several examples are worked. A single-machine system boundary is approximated and several clearing times evaluated. Slightly conservative estimates were obtained. A three-machine system boundary is also generated for near capacity generation on each machine. For the fault used to generate the boundary, the clearing time estimate was exact. This example was also used to evaluate the multiple allocation procedure. The boundary approximation was generated using four allocations and three postfault networks. In this case the estimates of the clearing times were good for eleven of the twelve combinations.

ACKNOWLEDGEMENTS

During the course of my work at Case Western Reserve University I have become indebted to many people and organizations. I wish to express my sincere thanks to Dr. Sanjoy K. Mitter under whose expert direction this research was performed. I am grateful to Dr. Tomas Dy-Liacco of the Cleveland Electric Illuminating Company for his stimulating discussions and helpful advice. Many thanks are also due the Illuminating Company for the use of their computer and to K. Ramarao and the members of the System Protection Group for their help with many programming details. I am also indebted to the Control of Complex Systems Group of Case Western Reserve University for providing excellent facilities and much-needed financial support. I also express my gratitude to my fellow graduate students for their aid. Finally, I wish to acknowledge the immeasurable support of my wife and family which is a debt that can never be fully repaid.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE PAGE

FIGURE PAGE

LIST OF TABLES

TABLE PAGE

INTRODUCTION

The increase in size and complexity of electric power utilities has forced interest in the automatic control of power systems. Recently, several works have appeared on various aspects of this control problem [1-5]. In particular, Tomas Dy-Liacco [4] has formulated a multi-level scheme for on-line computer control of power systems. He has defined the security control aspect of this problem and has shown the applicability of pattern recognition to steady-state security assessment.

Figure A has been borrowed from [4] and depicts the multi-level control strategy. Easily evident is the integration of security assessment into the over-all control scheme. At several points in the drawing a security check is made of the present or some future state. The basis for this test is a vector of coefficients, α, which are obtained from a pattern classifier which is trained off-line (see the upper right-hand corner). This thesis purports to explore the possibility of using similar techniques to assess transient stability.

During the normal operation of a power system, the operator has little but his experience with which to assess whether the present operating state is secure with respect to common transient disturbances. If he is fearful that such a disturbance will cause one or more machines to lose their synchronous operation, he has no means at present with which to check this out. Perhaps there is another allocation of generation which will minimize the possibility of such an instability at slightly less economic benefit. If a fault does occur, knowledge as to whether the system will recover of itself is not normally present until the event has taken place. If this assessment could be made, perhaps some corrective action could be made so as to minimize the disturbance to the system. In this thesis a technique is developed which, if realized, will enable such decisions to be made. If the operator desires to test a contingency, he will be able to make the test by simulating the system up to the clearing time of the fault being tested. If a fast-time simulation is available, evaluation of security can similarly be made automatically upon detection of a fault.

It was this power system application which motivated this work. For the reader familiar with power system transient stability analysis, Chapter I may be bypassed. It contains some basic concepts, definitions, and assumptions; the well-known classical model of a power system is given there. Chapter II contains a discussion of some recent work on transient stability analysis using Liapunov fuctions and presents a new technique for assessing stability. In Chapter III several numerical examples are given; estimates of critical clearing times are shown to be accurate. Finally, Chapter IV discusses some more general uses for the technique.

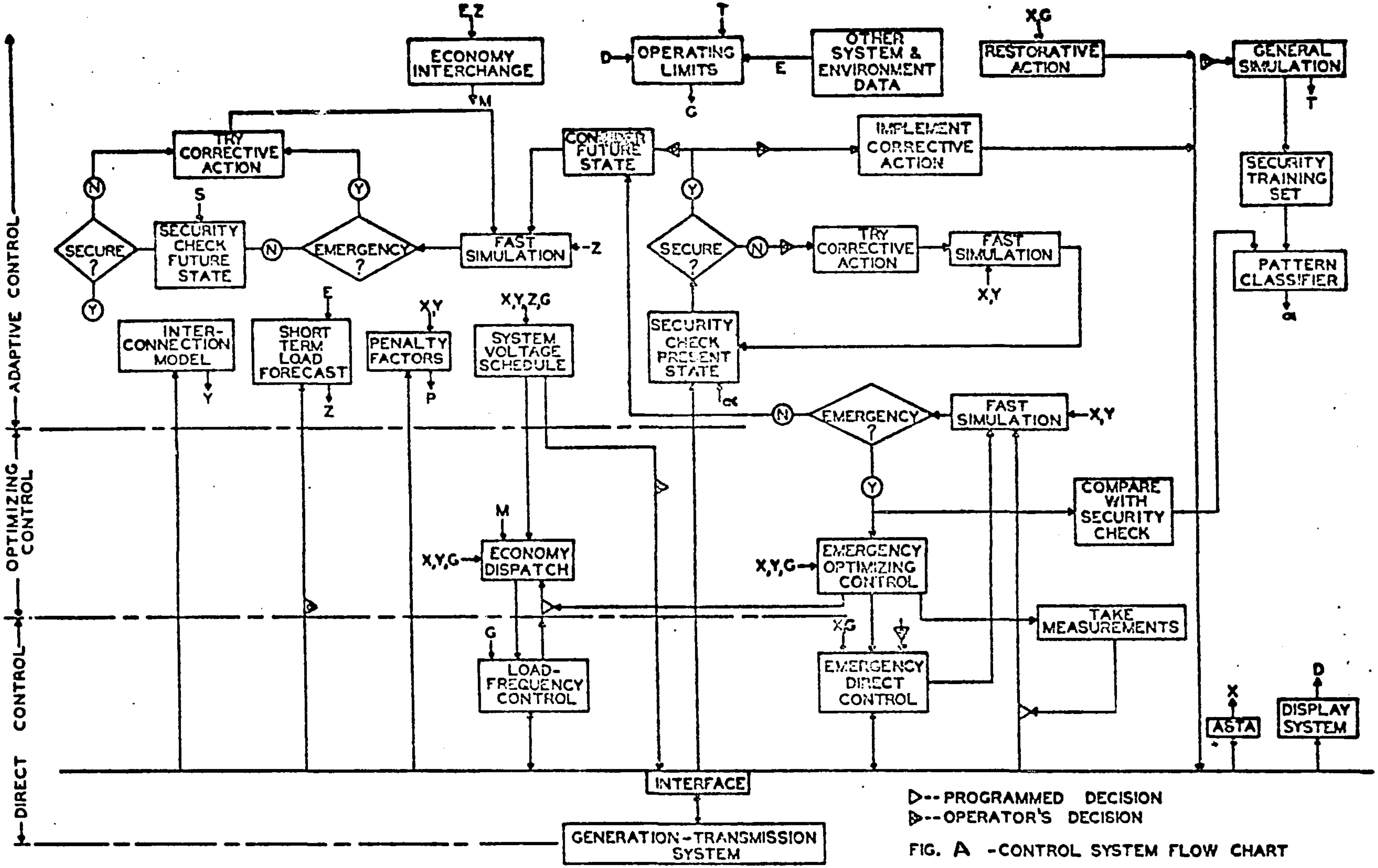


FIG. A -CONTROL SYSTEM FLOW CHART

CHAPTER I

PRELIMINARIES

1.1 Power System Transient Stability

Power system stability is the property of alternating current power systems which insures that the system will remain in operating equilibrium through both normal and abnormal operating conditions. When used in reference to interconnected synchronous machines, operating equilibrium refers to the synchronous, or common-frequency, operation of all machines in the system. Loss of this synchronous behavior will be caused by a disturbance. The disturbance may be slight and considered normal in terms of frequency of occurance during operation or it may be severe and unusual.

The traditional approach is to divide power system stability analyses into two classes based upon the type of disturbance; these are steady-state stability analysis and transient stability analysis. Steady-state analysis is concerned with slow and gradual changes that occur during operation such as governor response to slight changes in speed, system response to small changes in load, and the action of voltage regulation equipment to normal fluctuations. Transient analysis is concerned with severe disturbances such as sudden, large load changes and short circuits. Since short circuits, or faults as they are called, constitute the most severe disturbance to a power system, their effects must be determined in nearly all stability studies.

When a fault occurs on a power system, the faulted portion of the network is usually removed or isolated from the system by protective relaying equipment. The impedances between the various machines in the system change abruptly when the fault occurs and again when clearing is accomplished. Often, attempts at reinstating the faulted line are made by reclosing the relay breakers after a fixed time. This serves to perturb the system yet again. The power balance between mechanical input and electrical output in each machine in the system is upset by these disturbances. The question to be answered by a transient stability analysis is whether or not the system will find a new stable equilibrium at synchronous speed after the fault is isolated from the system. The answer to this question depends upon several factors, the type, location and duration of the fault, and the power level of each machine at the time of the fault.

There are several types of faults that can occur in a three-phase power distribution system. These are three-phase, two-line-to-ground, line-to-line, and one-line-to-ground short circuits. The symmetrical three-phase fault represents the most severe disturbance to the system. Such a fault results in the loss of all power transmission along the faulted line and presents a zero shunt impedance to ground. The remaining faults appear in the above list in decreasing order of severity. The single-line-to-ground fault is typically of most frequent occurance and the three-phase fault is the most rare [6]. The distribution of the remaining types depends upon the system considered. At the risk of being conservative, and for the sake of computational simplicity, only symmetrical three-phase faults will be considered in this thesis.

The location and duration of the three-phase fault also determines the severity of the disturbance to the system. The location determines the impedances seen by the machines in the network both during and after the fault. As a consequence, it influences the power flows during these periods. The duration governs the time during which the fault impedance is on the system and hence affects stability. One assumption will be made that may restrict as to location the type of faults to be considered in this thesis.

Assumption 1.1

All networks resulting from the removal of a faulted line will be steady-state stable.

This assumption accounts for the possibility of a system recovering from a fault temporarily and then losing synchronism due to the resulting network. This effect can be classified as a steady-state stability problem. Transient stability analysis based upon this assumption is termed "First Swing Transient Stability Analysis" [7].

The final factor affecting transient stability that was mentioned previously is the generation distribution. For a given disturbance the operating condition of a machine, i.e. the power output, will affect the amount of power unbalance that occurs during a fault condition. The resulting acceleration or deceleration of the machine will be determined from this unbalance. For this work, the total generation will be assumed known, and the effect of different allocations between the machines will be investigated.

This discussion of the transient stability problem has laid down restrictions upon the disturbances which will be allowed. Only three-phase faults will be considered; steady-state unstable postfault networks will be neglected; and the total amount of generation will be given information. The following discussion will formulate the mathematical model of a power system that will be used within these restrictions.

1.2 Power System Model

The equation of motion for the rotor of a synchronous machine in the transient state is given by

$$M \ddot{\delta} = P_i - P_u \tag{1.1}$$

where M is the angular momentum of the shaft, δ is the shaft angle referred to a synchronously rotating reference frame, P_i

is the mechanical power input to the shaft corrected for rotational losses, and P_u is the electrical power output corrected for electrical losses. Equation 1.1 is the usual "Swing Equation" [6,8].

The mathematical form of P_i and P_u depend upon the desired detail of representation. For this thesis, the "Classical Machine Model", i.e. a constant voltage behind a transient reactance, will be used. The use of this model requires making certain assumptions. In order to simplify calculations further, certain network simplifications will also be made. These are listed as follows:

1. The input power to a machine remains constant during the entire transient period.
2. All damping or asynchronous torques are negligible.
3. Each machine in the system may be represented by a constant voltage source in series with a constant reactance.
4. The angular momentum of a machine is constant during a transient period.
5. The mechanical angle of each machine rotor coincides with the electrical phase of the voltage source behind the reactance.
6. The resistance of all transmission lines may

be neglected.

7. Synchronous power may be calculated from a steady-state solution of the network to which the machines are connected.

A detailed discussion of these assumptions is not warranted here. Such detail may be found in [6-12]. However, several brief comments should be made.

When a machine is in steady state as it is prior to a fault, the mechanical input is equal to the electrical output plus losses. When a disturbance occurs, the output is abruptly changed. The input, however, is controlled by the speed governor of the machine's prime mover. The governor has a deadzone in which it will not act and has a large time constant associated with input changes when compared against the period of a transient swing. As a result, governor action usually does not take place during the interval of interest for first swing analysis. To be precise, governor action may not typically take place for a full second following a fault.

The angular momentum, M, does vary with speed. However, since percentage speed changes are small until synchronism is actually lost, assuming that M is constant introduces little error and significantly simplifies the form of 1.1. Consequently, assumption 4. is usually made and M is assigned its value at synchronous speed.

Due to the neglect of governor action and the assumption of constant angular momentum, the validity of the resultant model must be restricted to a time interval of less than one second following a fault. After this time interval, governor action will definitely have an effect upon the motion of the rotor. This time period is sufficient for first swing analysis.

The above restrictions and assumptions have been made in many previous studies, [6,7,16-20], and although the resulting model is not exact, it produces reasonable, but conservative results, [7]. It will be adequate for this thesis.

Consider the case where the power system contains N machines connected together by a linear, passive network [1]. Then under the assumptions given, the system is a multiport network driven by constant voltage sources. Denote the complex voltages by

$$\hat{V}_i = V_i \angle \delta_i \quad i = 1,2,\ldots,N \qquad , (1.2)$$

and the terminal impedances of the network by

$$\hat{Z}_{ij} = Z_{ij} \angle \theta_{ij} \quad i,j = 1,2,\ldots,N \qquad . (1.3)$$

The real power flow out of each source can be expressed as

$$P_u^i = \sum_{j=1}^{N} V_i V_j / Z_{ij} \ \ \mathrm{SIN}(\delta_i - \delta_j + \theta_{ij}) \quad i = 1,2,\ldots N \qquad . (1.4)$$

Then, substituting into 1.1, the motion of the rotors of the machines is described by

[1] This will be the case throughout this thesis.

$$M_k\ddot{\delta}_k = P_i^k - \sum_{j=1}^{N} V_kV_j/Z_{kj} \quad \mathrm{SIN}(\delta_k-\delta_j+\theta_{kj}) \quad k = 1,2,\ldots N \ . \quad (1.5)$$

Equations 1.5 are coupled nonlinear differential equations. The steady-state solution is obtained by solving

$$P_i^k = \sum_{j=1}^{N} V_kV_j/Z_{kj} \quad \mathrm{SIN}(\delta_k-\delta_j+\theta_{kj}) \quad k = 1,2,\ldots,N \quad . \ (1.6)$$

Note that since only difference angles are involved, the solution to 1.6 is determined up to an arbitrary constant. Hence, one of the machines, termed the swing machine, may be chosen as a reference and given an arbitrary value. The equilibria found in 1.6 will be periodic with a period of 2π radians due to the presence of the sine function. Even if attention is restricted to one principal interval of 2π radians, say [-π,π], there will in general be multiple equilibria. Some of these equilibria will be stable and others will be unstable. For most analyses these must be located.

Now, as mentioned previously, the question of loss of synchronism is concerned with whether the system will find a postfault equilibrium at synchronous speed. Since 1.6 contains only difference angles, it is not necessary to attach significance to the absolute value of the angles but only to the differences between them. Hence, one of the angles in 1.5 may be assigned an arbitrary initial value and the difference angles calculated from the solutions to the N equations; alternatively, one of the equations may be subtracted from all the others and the system modelled as N-1 equations in the difference angles. Thus, if a system has N machines, there are N-1 independent difference angles and the phase space consists of these angles and their derivatives and hence is of 2N-2 dimensions.

Loss of synchronism is reflected in the rotor difference angles. If θ_{kj} are small, then when one of the machine angles is greater than π radians away from the others, the sign of the corresponding sine function terms changes and power flows reverse tending to accelerate the machine further away from the others and no equilibrium can be found. For this work, θ_{kj} will be assumed small and attention will be focused on the region defined by

$$|\delta_k - \delta_j| \leq 180^\circ \qquad \begin{array}{l} k = 1,2,\ldots,N \quad k \neq j \\ j \text{ - Swing Machine} \end{array} \qquad . \quad (1.7)$$

In summary, then, equations 1.5 constitute a simplified model of an electric power system consistent with the given assumptions. These equations are used for simplicity. Results obtained with them will be realistic and will serve the purpose of illustrating the technique that will subsequently be proposed.

1.3 Definitions

In section 1.1 the factors affecting transient stability were discussed and some restrictions were placed upon these factors. In section 1.2 the system model was presented and the relationship between synchronous behavior and machine rotor angles discussed. These ideas can be condensed into the following formal definition [21] :

> A power system is said to be transient stable for a given location and duration of a symmetrical three-phase fault if after clearance of the fault or faulted portion within the given time, all synchronous machines connected in the system regain their synchronous operation and adjust their rotor angles to either their original or some new steady-state value. If any one or more machines tend to accelerate or retard continuously, then the system is considered unstable for the particular type and duration of the fault.

For the model that we have chosen, bounded periodic motion is synchronous operation.

One value that is of considerable interest to power system engineers is the critical clearing time. This is the maximum time period over which a fault can be tolerated in a system such that removal of the fault within this time interval will not produce instability. This time is obviously dependent upon the fault considered.

From the discussion in Section 1.2 concerning the handling of faults on a system, three modes of operation can be distinguished. The steady-state operating condition of the power system before any fault occurs is termed the prefault operating condition. The operation while a fault is on the system is called the operation during a fault. The operating condition after a fault has been cleared from the system is the postfault operating condition.

CHAPTER II

TRANSIENT STABILITY BOUNDARY GENERATION BY PATTERN RECOGNITION

2.1 Transient Stability Regions and Liapunov Functions

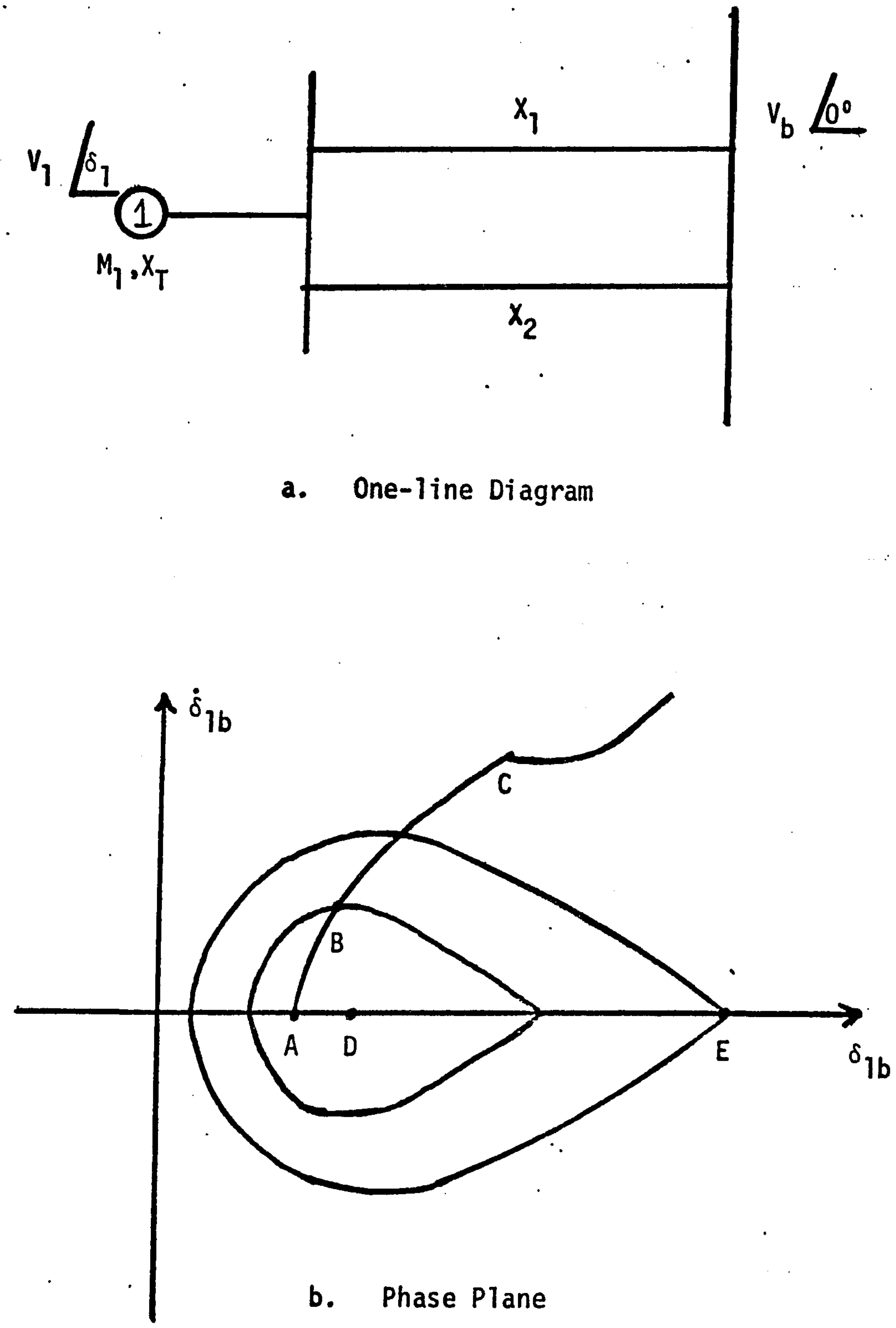
The nonlinear form of equations 1.5 preclude any direct solution. Consequently, transient stability of power systems has usually been studied via simulation. Prior to the advent of the digital computer, simulations were performed on a special purpose analog computer known as the "AC Calculating Board" [6]. Systems were represented as in Section 1.2 and various fault conditions were tried. Critical clearing times were determined by trial and error procedures. When the use of digital computers became widespread, a similar method was followed using them. It was possible, however, to represent the system in greater detail by including the effects of various control devices. Special purpose programs were written, [13-15], and used for systems analysis and design. Such programs require a large number of computations, but provide the designer with valuable information concerning system parameters.

Analytical stability techniques do exist for the transient stability problem, but they are only applicable to the one or two machine cases. These simple cases can be analyzed via the equal-area-criterion [6]. Recently, several methods based upon energy considerations and Liapunov functions have appeared in the literature [16-32]. These techniques show promise of being applicable to multimachine problems. All of these works attempt to locate a so-called "transient stability region" in the system state-space and once this is obtained, can estimate the critical clearing time with a single simulation. The transient stability region concept is fundamental to this thesis, and hence, it must be considered in some detail.

Consider a single-machine system delivering power through several parallel transmission lines to an infinite bus, Figure 2.1. The equation of motion describing this system in the transient state is, from Section 1.2,

$$M_1\ddot{\delta}_{1b} = P_{in}-V_1V_b/X_{1b}\ \mathrm{SIN}(\delta_{1b}) \qquad (2.1)$$

where the subscript b refers to the infinite bus. The operation of this system is best explained in the phase plane, Figure 2.1. In a prefault state, the system would be operating at point A, the steady-state operating point. When a fault occurs on one of the transmission lines, the impedance between the machine and the bus, X_{1b}, is increased and the machine begins to accelerate. If the fault is cleared, solutions for the postfault system can be stable, indicated by switching at point B, or unstable, indicated by switching at point C. A stable solution will be bounded and periodic about some stable postfault equilibrium, point D. As switching occurs later along the fault trajectory, the amplitude of these oscillations become larger until the solution passes through the unstable equilibrium at point E. After this



a. One-line Diagram

b. Phase Plane

FIGURE 2.1 Single-machine System

switching point, postfault solutions will be unstable. An equation for the solution passing through point E can be written for this case and the region enclosed by this equation is termed the transient stability region for the given power level and fault. This region is characterized by the fact that if the state of the system lies within it when the fault is cleared, the resulting postfault trajectory will be stable.

For the single-machine case, the system may be analyzed in the phase-plane as above; an exact equation for the switching boundary can be written and the fault trajectory can be integrated once up the boundary to obtain the critical clearing time. For the multimachine case, or even for a more detailed representation of the single machine, the exact form of the actual boundary is either not easily mathematically expressable or is unknown. As a consequence, researchers have gone to the theory of nonlinear differential equations and used energy functions or Liapunov functions to obtain estimates of transient stability regions.

Magnusson and Aylet, [16.17], wrote energy equations as a function of rotor difference angles using the model given in Equations 1.5. Both methods require the location of the equilibrium points, Equations 1.6. Both are used to solve three-machine examples. These techniques yield quantitative measures of relative stability for the comparison of similar systems, but do require "large expenditures of effort",[17]. Dharma Rao, [18], and Dharma Rao and Ramachandra Rao, [19], tie the energy concept to Phase-plane analysis for the single-machine case. [19] treats this case in various degrees of detail as far as machine models are concerned. The transition to the use of Liapunov functions for estimating the transient stability region began with Gless, [20], and El-Abiad and Nagappan, [22]. Since then, [23-30] have been concerned with improving the form of the function by one method or another. Siddiqee, [24], considers the one-machine problem in some detail and includes the effects of flux-decay, governor action, and voltage regulation in separate formulations. Willems, [25], considers the single machine also and includes governor action, pole saliency, and damping torques. [16,17,20, 22,26] solve multimachine problems with the largest example being that of El-Abiad and Nagappan consisting of a four-machine system with two load centers. Several other methods have been explored. In [31] generalized nonlinear network analysis is applied to the problem and in [32] the Volterra Series is used to approximate the stability boundary. In both cases only single-machine examples are considered.

The Liapunov Function Method tries to locate the domain of attraction of the system about a stable postfault equilibrium. This is accomplished in three steps:

1.) Write a Liapunov function, V(x), for the given model.

2.) Locate the nearest saddle point, $\hat{x}$, to the stable postfault equilibrium.

3.) $V(x) = V(\hat{x})$ is the boundary estimate.

Since the Liapunov Theorem, [33], is invoked, the region described by $V(x) = V(\hat{x})$ provides only a sufficient condition for postfault stability. The advantages of this technique are that the critical clearing time may be determined with one simulation, and relative stability information may be obtained, [16, 17, 34].

There are several disadvantages to this method. First, obtaining the Liapunov functions is not always straightforward. Second, considerable effort must be spent obtaining the critical points, and as exemplified by [35], finding the nearest one to the equilibrium can sometimes be difficult. Third, the Liapunov Stability Theorem provides only a sufficient condition for stability and hence, the boundary thus obtained might be very conservative. Fourth, although some extrapolation is possible, the generation procedure must in general be performed for each postfault condition and each power level of machine operation. Finally, the difficulty in applying the technique grows rapidly with system size and with the complexity of representation of each machine.

However, the information supplied by the transient stability boundary estimate is desirable. It provides valuable design information for choosing system parameters and can give information to the system operator for a given fault. Since ignorance of impending instability is undesirable from a reliability standpoint, the conservative aspect of the Liapunov function is good; however, one would like to include as much of the stable volume as is possible.

In the next section, an alternative method of approximating the boundary is presented which overcomes some of the disadvantages listed above. This new method tries to obtain the same information.

2.2 Transient Stability Regions and Pattern Recognition

Consider again the phase portrait of Figure 2.1. As mentioned previously, the curve shown represents the transient stability boundary for the single-machine case with the given model. Now, the equation for this curve is nonlinear and contains trancendental functions. Thus, it is difficult to use. Weissenburger, [36], among others, has explored the possibility of finding some sort of best polynomial Liapunov function for certain systems. This idea has appeal due to the linearity of such a function in its coefficients and due to the ease of handling polynomials. Consider the possibility of not writing such a Liapunov function, but rather of using the polynomial to directly approximate the region of Figure 2.1. From inspection it seems that an ellipse may approximate it rather well.

Such an approximation may be formulated within the field of Pattern Recognition rather easily. If after the clearing of a given fault, the state of the system is within the transient stability region, then the system will be postfault stable; if it is outside, it will be unstable. Thus, the principle region of interest in the state space is divided into two sets, one containing unstable postfault initial conditions, and one containing stable postfault initial conditions. If we view as samples from these sets the initial conditions for the given postfault system and choose as a decision function a quadratic such as

$$\begin{aligned}&\alpha_1 x_1^2+\alpha_2 x_2^2+\ldots+\alpha_N x_N^2+\alpha_{N+1} x_1 x_2+\ldots+\alpha_p x_{N-1} x_N\\&+\alpha_{p+1} x_1+\ldots+\alpha_{d-1} x_N+\alpha_d = 0\end{aligned} \tag{2.2}$$

where if N is the dimension of the state space,

$$p = N + \binom{N}{2} \quad ,$$

and

$$d = 2N + 1 + \binom{N}{2} \quad ,$$

then we have described a pattern recognition problem [37]. Such a formulation is attractive since given the samples, there are many available techniques for finding the coefficients, [38,39]. One straightforward way of gathering such samples is by system simulation, and hence, the problem can be formulated as a combination of a sample gathering and a pattern separation procedure.

To reiterate, consider simulating the postfault system with initial conditions at point A of Figure 2.1. All the information that this could infer about the stability boundary is contained in the location of point A and in the stability or instability of this initial condition. Thus, if we define two classes, one containing the unstable initial conditions and one containing the stable ones, and a quadratic decision surface such as that given in Equation 2.2, a dichotomous pattern recognition problem is described. If indeed an ellipse is a good approximation to the boundary, the pattern separation would yield a vector of coefficients for a quadratic function which would define the transient stability region for the system from which the simulated data was extracted.

In contrast to the disadvantaqes of the Liapunov method, this technique has definite appeal. Conceptually, there would be no increase in difficulty by added system complexity; all such information would be contained in the simulation results. As system size increases, the space in which the separation is performed increases in dimension and hence more simulations would be required. This would cause an increase in computation time but conceptually the technique would be the same as for small systems. Also, no equilibrium points need be found. By appropriate choice of an algorithm the separating ellipse will be as close an approximation to the boundary as the data and theory will allow. As will be seen in Section 2.3, some algorithms can be modified to penalize the inclusion of unstable points and thereby offer some control over the conservativeness of the approximation.

As explained above, the approximation obtained from the outlined technique is still only applicable for a single postfault system and a single distribution of generation among the machines. This can be overcome, however, by choosing a set of postfault conditions, and a set of distributions for a fixed amount of total generation and then redefining stable and unstable points. We do this as follows:

Definition 2.1

Given a set of postfault networks, and a set of generation distributions for a fixed amount of total generation, an initial condition for the system is said to be stable if the solution for this condition is stable for all combinations of the given sets. If the solution is unstable for one allocation-network pair, then the initial condition is said to be an unstable one.

Thus, if samples are gathered using Definition 2.1 and a quadratic separation is performed, the boundary approximation would possibly be conservative, but would be independent of the fault and would reflect the stability property of the system for the operating condition, namely the amount of total generation.

In summary, the proposed technique would proceed in the following manner:

1.) For the system to be studied, select an amount of total generation.

2.) Choose a set of representative faults for the network and their resultant postfault networks consistent with Assumption 1.1.

3.) Choose a set of distributions of the total generation among the machines of the system.

4.) Gather sample patterns by simulating the post-fault system for various initial conditions using Definition 2.1 to determine stability.

5.) Separate the pattern sets by a suitable algorithm using a quadratic surface.

In practice the set of faults would be chosen by experience, but each line could be systematically removed from the network if desired. Care would be taken to represent both high and low power levels for the machines in the system in carrying out step 3 . Also, step 5 could be accomplished with a recursive algorithm and would be performed at the same time as the data is gathered. Finally, in reality the entire procedure would be performed for several amounts of generation in order to fully represent the system.

Several interesting points can be made concerning an ellipse generated in the above manner. First, the postfault initial conditions are tested irrespective of how the system got to them. Some of the tested points may not be possible to achieve from the given network by any fault whatsoever. In any event, the ellipse reflects postfault behavior by knowledge of the state at the instant of switching. Second, by testing each point for a set of postfault networks and a set of allocations, the figure can also be made to reflect postfault stability of the system for the given amount of total generation. The stability evaluation for any given fault may be conservative, but the safe switching region is reflected by this boundary for these combinations of system parameters.

In this section a pattern recognition technique has been presented which will generate an approximation to the transient stability boundary. Its advantages over present techniques has been given. In the next section, a method of implementing the given procedure is presented and discussed.

2.3 Method of Implementation

The problem as previously given is concerned with power system transient stability. The given information necessary to begin the problem is listed as follows:

1.) Synchronous machine constants, namely maximum power ratings, angular momenta at synchronous speed, and transient reactances.

2.) Network configuration and impedances.

3.) Amount of total generation.

4.) Steady-state conditions from which the voltages behind the transient reactances will be calculated.[1]

Using this information, a set of faults will be chosen which in turn determine a set of postfault networks, and a set of generation distributions will be assigned.

From this point, the proposed procedure is seen to consist of two phases: a data-gathering phase in which initial conditions are tested and a pattern-separation phase in which the boundary approximation is actually generated. In this section each of these phases will be discussed and methods of implementing them will be given.

2.3.1 Data Gathering

Consider the region for the single-machine problem, Figure 2.1. It is only reasonable to say that points conveying the most information about the boundary are those near to it. Points well to the interior or exterior will be accounted for by boundary points. If in the multimachine case the boundary has a similar shape, then boundary points will again suffice as data

[1] These conditions are usually in the form of known bus voltages.

points. If it is agreed that the significant data lies near the boundary, the problem then consists in finding such data when there is little or no knowledge as to the boundary location.

For the two-dimensional, single-machine problem one solution is obvious; choose a constant value of angular displacement and search along it in both the positive and negative velocity directions until the boundary is located using simulations to determine stability. Then, choose as data a stable and an unstable point within some distance of each other. These points locate the curve within some error along the line on which they are gathered. If this is done for various values of angle, a grid such as that shown in Figure 2.2 is obtained and a fair representation of the boundary can be drawn.

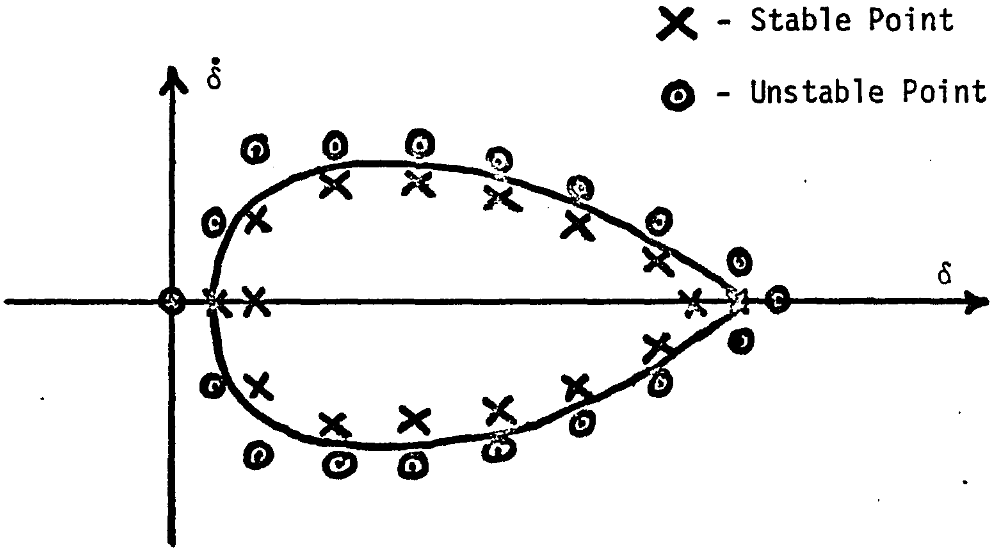


Figure 2.2 Samples for a Two-dimensional Problem

This procedure was followed with a single-machine system[1] and a separation based on such data is indeed easy to obtain and is very representative of the true boundary.

Consider performing this type of a data-gathering procedure on a multimachine system. The number of points in such a grid will increase as the power of the dimension of the space if the same distance between points is maintained. One can easily see that since a simulation is required at each point, the computation time will become prohibitive. Usually, such a large number of points is not necessary to define the boundary and an alternative procedure can be suggested.

Let us assume that it is desired to approximate all portions of the boundary with equal accuracy. Then, the data should be gathered uniformly over the boundary. The following procedure is one method of accomplishing this:

1.) Choose a point interior to the region.

2.) Scatter points uniformly over the surface of a unit sphere about the point.

3.) For each point on the surface of the unit sphere, locate the boundary of the transient stability region using system simulations along the direction determined by the line passing through the

[1] See Chapter III.

point on the surface and the point chosen in 1.

4.) Use as sample patterns each of the tested initial conditions.

The point which is interior to the region can be chosen to be a steady-state equilibrium point for some prefault condition under consideration. Care must be taken to insure that this point is a stable equilibrium point and that it is a stable postfault initial condition for all other power allocations being used.

There are several ways of scattering points on the unit sphere. This can be done in a Monte Carlo fashion using a random number generator or an actual grid can be placed upon the sphere. The random method allows one to stop gathering data after any direction, while every point on the grid should be used for consistency. Uniformity is not a necessity but care should be taken to adequately represent the surface that is being approximated. If there are any preferred directions, the search should be concentrated along them. In any event, a set of rays is obtained.

The problem, now, is to locate the boundary along each direction in the above set of rays. This entails a linear search and can be accomplished in many ways. The procedure decided upon here consists of a dual-mode search. The first mode consists of taking fixed steps of a given distance along a ray while simulating the postfault system for each allocation-network pair at

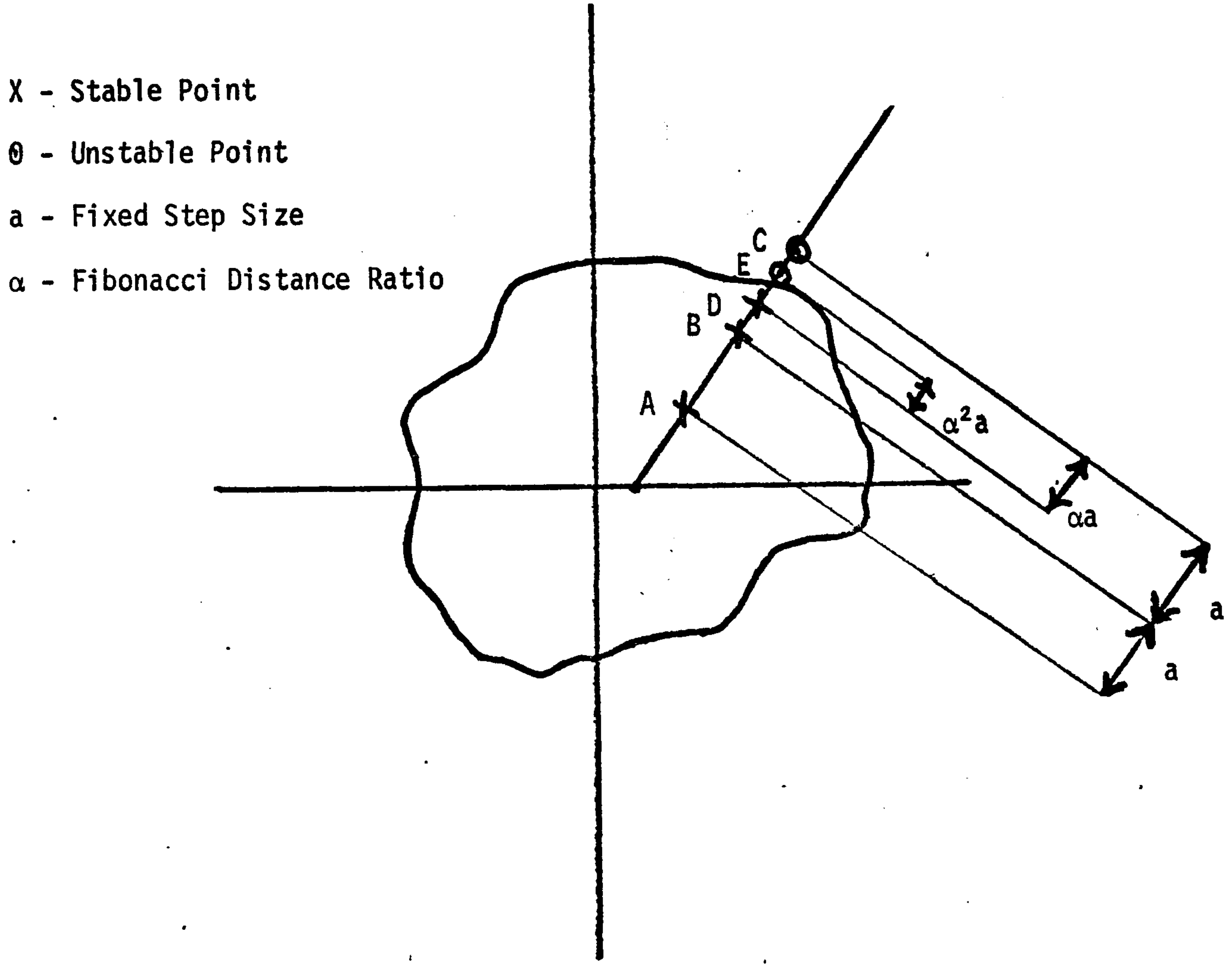


FIGURE 2.3 Search Method

each point. When a point which is unstable is first encountered, the first mode is terminated. This mode is exemplified by points A, B, and C in Figure 2.3. At this time in the search the boundary has been located as being somewhere between the last stable point and the unstable one. Generally, the fixed step size will be large so as to get to the neighborhood of the boundary rapidly. Hence, the second mode is provided to narrow the location down with greater accuracy. This is accomplished using a Fibonacci search [41]. It is used to narrow the interval between a stable and an unstable pair until they are within some fixed distance of each other. To provide an example, after mode one is completed, the interval of uncertainty along the ray in Figure 2.3 is between points B and C. The next point, D, is chosen a distance α times the distance between B and C from C towards point B. This point tested stable and hence now the interval is between points C and D. The next point, E, is chosen towards C from D a distance α times the distance between these two points. This new point tests unstable and since the distance between D and E is sufficiently small, the search is terminated along this direction.

One question of immediate interest is how many directions must be searched ? If the random procedure is used, then naturally the convergence of the number generator must be considered. Usually, this fact will insure more than a sufficient number of points. If a grid technique is used, the number of points should reflect the desired accuracy of approximation. The distance between data on the actual boundary is a function of the volume of the actual stable region. In either case the decision is somewhat heuristic. However, there is a method of testing the accuracy of the approximation which can be used as an aid when confronted with this decision. The method is discussed in Section 2.3.2 and further discussion is deferred to that point.

In summary, then, a logical data-gathering procedure has been outlined in this section. A set of rays is used to determine directions of search and the location of the boundary is found along them. In the next section, the second phase of the proposed technique is outlined. The pattern recognition problem is formulated and several algorithms are presented.

2.3.2 Pattern Recognition Formulation

There are three main parts to a dichotomous pattern recognition problem - characterization, abstraction, and generalization. The characterization problem consists of a mapping of the pattern set into a set of real variables, x, which represent the patterns and which can easily be processed by a digital computer. The abstraction problem consists in finding a decision function of these variables, F(x), such that

$$F(x) \begin{cases} > 0 & x \varepsilon A \\ \leq 0 & x \notin A \end{cases} \qquad (2.3)$$

where A is the pattern class. Finally, this function is applied to new samples and its ability to correctly classify them is analyzed.

The key to the entire process lies in the characterization of the patterns. The proper choice can make the rest of the procedure obvious and easy to perform. Unfortunately, there are no standard methods to follow and theoretical results are few.

The abstraction problem is usually approached in one of two ways - deterministically or statistically. In the former, a function, F(x), which perfectly separates the two sets according to Equation 2.3 is assumed to exist. The form of this function is further assumed known up to a set of parameters, α. Results with this approach depend upon how well the assumed form fits the actual separating surface and whether the sets overlap at all. Statistical Hypothesis Testing forms the basis for the second method of approach. The two classes are assumed to have certain probability densities, $p(x/A)$ and $p(x/\bar{A})$, and certain probabilities of occurance, $P(A)$ and $P(\bar{A})$. The decision function is then given by the conditional probability function, $P(A/x)$, which may be found by Bayes Rule,

$$P(A/x) = \frac{P(A)\ p(x/A)}{P(A)\ p(x/A)+P(\bar{A})\ p(x/\bar{A})} \quad . \quad (2.4)$$

The usual approach is to assume the form of $P(A/x)$ either directly or indirectly through assumptions on $p(x/A)$, and $P(A)$. Results with this technique depend again upon how well the assumed forms fit the actual functions.

The transient stability problem that is of concern here fits very well into the pattern recognition scheme and into either abstraction approach. A concrete formulation and several algorithms given in the following sections.

A. Characterization

Consider a set of known postfault initial conditions for a power system,

$$x(i) = \{x_1(i), x_2(i), \ldots, x_k(i)\} \quad i = 1,2,\ldots,M \quad , \tag{2.5}$$

where the classification, postfault stable or unstable, of each is known. Denote this set of patterns by

$$\chi = \{x(1), x(2), \ldots, x(M)\} \quad . \tag{2.6}$$

Define the two classes by

$$x(i) \ \varepsilon \begin{cases} A & x(i) \text{ unstable} \\ \bar{A} & x(i) \text{ stable} \end{cases} \quad . \tag{2.7}$$

Since it is thought that an ellipse would divide these classes well, choose a decision function of the form

$$f(x) = \alpha_1 x_1^2 + \ldots + \alpha_k x_k^2 + \alpha_{k+1} x_1 x_2 + \ldots + \alpha_p x_{k-1} x_k \tag{2.8}$$
$$+ \alpha_{p+1} x_1 + \ldots + \alpha_{d-1} x_k + \alpha_d \ .$$

This is a general quadratic function and can be rewritten as

$$f(x) = \alpha^T \phi(x) \tag{2.9}$$

where

$$\phi(x) = \{x_1^2, \ldots, x_k^2, x_1 x_2, \ldots, x_{k-1} x_k, x_1, \ldots, x_k, 1\} \quad ,$$
$$\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_d\} \quad ,$$

and with

$$d = 2k+1+\binom{k}{2} = (k+1)(k+2)/2 \quad .$$

Thus, the set defined by

$$\Phi = \{\phi[x(i)]\} \tag{2.10}$$

will be used to discriminate between the two pattern sets and ϕ is the characterization mapping for this problem. Note that the mapping is one-to-one from the state-space into the space defined by Φ and that $\alpha^T\phi(x)$ is linear in the coefficients, α. This is the well known ϕ-machine concept [37].

B. Deterministic Abstraction

For a deterministic approach to our problem, a vector, α, is desired such that

$$\alpha^T\phi(x) \begin{cases} > 0 & x \epsilon A \\ \leq 0 & x \epsilon \bar{A} \end{cases} \quad . \tag{2.11}$$

Redefine the characterized patterns by

$$\phi[x(i)] \begin{cases} -\phi[x(i)] & x(i) \quad \text{stable} \\ \phi[x(i)] & x(i) \quad \text{unstable} \end{cases} \quad . \tag{2.12}$$

Then a vector is desired such that

$$\alpha^T\phi[x(i)] \geq 0 \quad i=1,\ldots,M \quad . \tag{2.13}$$

The abstraction problem then becomes to find a vector, α, such that 2.13 holds for all M transformed pattern vectors.

Algorithms for accomplishing this deterministically are generally derived in a common framework [38]. First, specify a criterion function, $J(\alpha)$, such that the minimum corresponds to a solution of the inequalities 2.13. Then, using a gradient descent procedure in the parameter space, find a minimum of J. In recursive form, such a procedure is given by

$$\alpha_{N+1} = \alpha_N - \rho\left[\partial J/\partial\alpha\Big|_{\alpha_N, x(N)}\right] \tag{2.14}$$

where ρ is a positive constant and x(N) is the pattern presented to the algorithm at the N-th iteration. Note that it is not expected that a perfect separation will be achieved for a significant number of patterns taken from a real power system. This must be kept in mind when choosing the criterion function, J.

There are many existing deterministic algorithms. In what follows only one of these will be described. For a discussion of others, see [38].

The deterministic algorithm which has recieved use in this thesis was first proposed by Ho and Kashyap[40]. It is a weighted minimum-squared-error adaptation scheme which uses all patterns at each iteration, and hence converges very rapidly. The discussion contained here closely follows that given by Blaydon[38].

Label the M patterns, x(1),x(2),...,x(M) where each is a k-dimensional vector. Construct matrices X and β such that

$$X = \begin{bmatrix} x^T(1) \\ x^T(2) \\ \cdot \\ \cdot \\ \cdot \\ x^T(M) \end{bmatrix} \qquad \beta = \begin{bmatrix} b(1) \\ b(2) \\ \cdot \\ \cdot \\ \cdot \\ b(M) \end{bmatrix} \tag{2.15}$$

where b(1),b(2),...,b(M) are positive scalars. Define a criterion to be minimized as

$$J = 1/2 \sum_{i=1}^{M} \{\alpha^T\phi[x(i)]-b(i)\}^2 \qquad . \ (2.16)$$

This technique attempts to cluster the values $\alpha^T\phi[x(i)]$ about the positive scalars b(i). Note that if there exists a set of such scalars and a vector α such that

$$\alpha^T\phi[x(i)] = b(i) \quad i=1,2,\ldots,M \qquad , \ (2.17)$$

the separation is accomplished and 2.16 indeed has achieved its minimum value.

Now, instead of minimizing 2.16 with respect to α, consider minimizing it with respect to both α and β with β constrained to be positive. The associated gradients are

$$\partial J/\partial\alpha = -X^T(\beta-X\alpha) \qquad (2.18)$$

and

$$\partial J/\partial\beta = (\beta-X\alpha) \qquad . \ (2.19)$$

Since α is unconstrained the gradient 2.18 can be set to zero, and the result is

$$\alpha^* = (X^TX)^{-1}X^T\beta \qquad . \ (2.20)$$

Since β is constrained to be greater than zero, it should be changed only in a manner that will guarantee this. Thus, following 2.14, let

$$\beta(j+1) = \beta(j)+\delta\beta(j) \qquad (2.21)$$

where

$$\delta\beta_i(j) = \begin{cases} 2\rho[X\alpha(j)-\beta(j)]_i & \text{if} \quad [X\alpha(j)-\beta(j)]_i \geq 0 \\ 0 & \text{if} \quad [X\alpha(j)-\beta(j)]_i < 0 \end{cases} ,$$

$$0 < \rho < 2 \qquad .$$

and where j is the iteration index and i denotes vector components. The over-all algorithm can be written as

$$\alpha(j+1) = \alpha(j)+(X^TX)^{-1}\delta\beta(j) \qquad . \ (2.22)$$

Convergence of this algorithm has been proven by Ho and Kashyap [40] if a solution to 2.17 exists. The algorithm will terminate even if a solution does not exist. Note that due to the utilization of the information from all M patterns at each stage, rapid convergence is to be expected.

In 2.22 only positive increments on b(i) are allowed. This is overly restrictive since if b(i) > 0, it can be decreased without violating the constraint that it must remain greater than zero. The algorithm can be modified to allow such changes in β. Redefine $\delta\beta(j)$ as

$$\delta\beta_i(j) = \rho \begin{cases} 2[X\alpha(j)-\beta(j)]_i & \text{if} \quad [X\alpha(j)-\beta(j)]_i \geq 0 \\ \max[-\beta_i(j)+\varepsilon, 2[X\alpha(j)-\beta(j)]_i] & \text{otherwise} \end{cases} \qquad (2.23)$$

where ε is a vector of positive scalers such that $\varepsilon \leq \beta(0)$. The need for a minimum allowable value on b(i) is that a noninformative minimum of $J(\alpha)$ exists at $\alpha = \beta = 0$ and this minimum can now be located by the algorithm unless the b(i)'s are so restricted. Convergence of this modified version is proved by Blaydon [38].

As mentioned previously, it is not expected that a perfect separation will be achieved for a real problem. Hence, when the above algorithm terminates, some patterns will be incorrectly classified; stable patterns will lie outside the separating surface and unstable patterns will lie inside it. The latter case is undesirable from a practical point of view; in power system control, ignorance of an impending instability,which this situation provides,can be disasterous. Hence, it would be better to be conservative and try and force a separation which will not include any unstable points. This would obviously be at the expense of excluding some stable volume from the approximating region.

The basic idea behind this modification is to penalize the criterion function for the inclusion of such points. If a pattern is correctly classified, its contribution to the sum in 2.16 is zero; it is only when it is misclassified that it contributes. Now, to force the exclusion of unstable points, simply penalize the terms in 2.16 which are due to these points. Thus, define the penalty factors by

$$\lambda_i = \begin{cases} a & x(i) \text{ unstable} \\ b & x(i) \text{ stable} \end{cases} \qquad a,b>0 \qquad (2.24)$$

where i is a pattern index. We shall refer to the ratio b/a henceforth as the penalty ratio . Then, the criterion function can be rewritten as

$$J = 1/2 \sum_{i=1}^{M} \lambda_i \{\alpha^T \phi[x(i)]-b(i)\}^2 \qquad . (2.25)$$

Define a diagonal matrix Λ by

$$\Lambda = \begin{bmatrix} \lambda_1 & & O \\ & \ddots & \\ O & & \lambda_M \end{bmatrix} \qquad . (2.26)$$

Then following the same procedure as before, we have

$$\partial J/\partial\alpha = -X^T\Lambda(\beta - X\alpha) \quad , (2.27)$$

$$\partial J/\partial\beta = \Lambda(\beta - X\alpha) \quad , (2.28)$$

$$\alpha^* = (X^T\Lambda X)^{-1}X^T\Lambda\beta \quad , (2.29)$$

and

$$\delta\beta_i(j) = \begin{cases} 2\rho\lambda_i[X\alpha(j)-\beta(j)]_i & \text{if } [X\alpha(j)-\beta(j)]_i \geq 0 \\ \max[-\beta_i(j)+\varepsilon, 2\rho\lambda_i[X\alpha(j)-\beta(j)]_i & \text{otherwise} \end{cases} \quad . (2.30)$$

with the same restrictions on ε and $\beta(0)$. Note that α^* in 2.29 is identical with α^* in 2.20. Hence, Equations 2.29 and 2.30 are the same as Equations 2.20 and 2.23 with $\rho\lambda_i$ substituted for ρ. Thus, the proof of convergence for the algorithm given by 2.29 and 2.30 is a trivial extension of that for the previous form [38] and the algorithm converges for $0 < \rho\lambda_i < 2$, $i = 1,2,\ldots,M$.

The various forms of the Ho-Kashyap algorithm that have been given will be used extensively. Comparison will be made with the results obtained with them and the stochastic abstraction algorithm to be considered next.

C. Stochastic Abstraction

In the deterministic problem there are M fixed pattern vectors and the abstraction problem is defined only with respect to these vectors. In the stochastic problem the patterns are viewed as samples from a larger population about which information is to be obtained. In the deterministic case each pattern is assumed to belong to one and only one class. In the stochastic formulation the classes may overlap and membership in a class is probabilistic.

As mentioned previously, the information necessary to classify patterns in the stochastic case is the conditional probability, $P(A/x)$. Yet,when the patterns are presented, class membership is given and not values of this probability. Hence, define an indicator variable as follows:

$$z(x) = \begin{cases} 1 & x \varepsilon A \\ 0 & x \varepsilon \bar{A} \end{cases} \quad . \quad (2.31)$$

Note that $E_z[z(x)] = P(A/x) \cdot 1 + P(\bar{A}/x) \cdot 0 = P(A/x)$. Thus, the pair $\{z(x),x\}$ contains all of the information available. The stochastic approach then assumes a form for $P(A/x)$, such as $\alpha^T \phi(x)$, and tries to choose α such that a criterion is minimized. The criterion function is now a regression function and the minimization is carried out by a stochastic approximation procedure [42] which is the analog of the gradient descent procedure given previously. For example, if

$$J = E\{y(\alpha)\} \quad (2.32)$$

where $y(\alpha)$ is a random function , the approximation algorithm is of the form

$$\alpha(j+1) = \alpha(j) - \rho(j) \left[\partial y / \partial \alpha |_{\alpha=\alpha(j)} \right] \quad , \quad (2.33)$$

with

$$\lim_{j \to \infty} \rho(j) = 0 \quad . \quad (2.34)$$

The conditions needed to guarantee convergence are:[1]

(i) $|E\{y'(\alpha)\}| < A|\alpha-\alpha^*| + B < \infty$ where $E\{y'(\alpha^*)\}=0$

(ii) $E\{y'(\alpha)-E[y'(\alpha)]\}^2 \leq \sigma_{y'}^2 < \infty$

(iii) $\sum_{j=1}^{\infty} \rho(j) = \infty \quad , \quad \sum_{j=1}^{\infty} \rho^2(j) < \infty$.

We shall not dwell upon the derivation of this procedure. A complete discussion may be found in [38] or [41]. Rather, an example will be given of a procedure similar to the Ho-Kashyap algorithm discussed previously.

Consider the regression function

$$J(j) = 1/j \sum_{i=1}^{j} \{z[x(i)]-\alpha^T\phi[x(i)]\}^2 \quad . \quad (2.35)$$

Define

$$Z(j) = \begin{bmatrix} z[x(1)] \\ z[x(2)] \\ \vdots \\ z[x(M)] \end{bmatrix} \qquad \Phi(j) = \begin{bmatrix} \phi^T[x(1)] \\ \phi^T[x(2)] \\ \vdots \\ \phi^T[x(M)] \end{bmatrix} \quad . \quad (2.36)$$

where x(1),x(2),...,x(M) are independent samples with the same distribution. The vector which minimizes J(j) is given by the least-squares fit and is

$$\alpha(j) = [\Phi^T(j)\Phi(j)]^{-1}\Phi^T(j)Z(j) \quad . \quad (2.37)$$

[1] Note $y'(\nu) \equiv \partial y/\partial\alpha|_{\alpha=\nu}$

This can also be written recursively as

$$\alpha(j+1) = \alpha(j) + R^{-1}(j)\phi[x(j)]\{z[x(j)]-\alpha^T(j)\phi[x(j)]\} \qquad (2.38)$$

with

$$R(j) = R(j-1) + \phi[x(j)]\phi^T[x(j)] \qquad (2.39)$$

where 2.37 and 2.38 are equivalent for large j for an arbitrary, positive definite R(0). R^{-1} can also be computed recursively as

$$R^{-1}(j) = R^{-1}(j-1) - R^{-1}(j-1)\phi(j)[\phi^T(j)R^{-1}(j-1)\phi(j)+1]\cdot \phi^T(j)R^{-1}(j-1) \quad . \; (2.40)$$

The Strong Law of Large Numbers can be invoked to assert that

$$\lim_{j\to\infty} J(j) = J(\alpha) = E_{x,z}\{z(x)-\alpha^T\phi(x)\}^2 \qquad (2.41)$$

with probability one. It is also possible to assert that α given by 2.38 converges with probability one to the vector α^* which minimizes 2.35, where

$$\alpha^* = [E_x\{\phi(x)\phi^T(x)\}]^{-1}\, E_x\{\phi(x)P(A/x)\} \quad . \; (2.42)$$

The conditions which insure convergence are:

(i) $E_x\{\phi(x)\phi^T(x)\}$ exists and is positive definite,

(ii) $E_x\{\phi(x)\phi^T(x)\phi(x)\phi^T(x)\}$ and $E_x\{\phi(x)P(A/x)\}$ exist,

(iii) R(0) exists and is positive definite,

(iv) $\alpha(0)$ arbitrary.

The convergence proofs and a more detailed discussion of this algorithm are given in [38].

This algorithm has been presented assuming that the pattern vectors are random. However, for the transient stability problem being considered here, the patterns are not random. In such a case we cannot specify a criterion function of the form 2.35. Instead the problem becomes one of identifying α_0 in the regression function

$$E_z\{z(x)\} = P(A/x) = \alpha_0^T\phi(x) \qquad (2.44)$$

where the assumption is that α_0 exists. Class membership is not truely random but is given a probabilistic description. In this case the algorithm given in 2.38 and 2.39 can be shown to satisfy a convergence theorem due to Dvoretzky[43]. Mathematical convergence does not really concern us here, however, since for the transient stability boundary, an α_0 does not exist; convergence will not be achieved due to approximation errors. However, as mentioned previously, there is reason to believe that an ellipse will achieve a satisfactory approximation and hence we expect the coefficients to approach some α' which is the best approximation to the transient stability boundary with the errors accounted for by the probabilistic description.

The conservative aspect is easily preserved with this algorithm. The normal discriminant is

$$\alpha^T\phi(x) = 0.5 \qquad . \qquad (2.45)$$

which weights the patterns from each set equally. To penalize the inclusion of one set or another, it is only necessary to raise or lower the threshold appropriately.

D. Generalization

The generalization property of pattern recognition problems is the ability to classify patterns not used in the training procedure. It is the ability of the recognizer to extrapolate to as yet unseen patterns and to classify them correctly. The usual way of measuring this property is to divide the set of available patterns using a portion for abstraction and the remainder for testing the obtained recognizer. For the problem that we have defined, the data set is essentially unlimited since we can gather as many patterns as we desire. Hence, the testing of the separation is not data limited as in most problems.

Consider scattering points in a uniform random manner within a volume containing both the transient stability region and its quadratic approximation. If each point is tested, a Monte Carlo estimate of the actual stable volume included in the approximation and of the volume of error can be obtained using the average number of correctly classified stable points and the average number of incorrect predictions respectively [45]. In the language of Probability Theory, a series of independent Bernoulli Trials has been outlined and it is desired to estimate the actual probability of success.

For this problem the volume of search is arbitrary since it will be a known quantity, but it would be desirable that it be just slightly larger than the union of the stability region and the approximation so as not to waste computation time testing a large number of correctly classified unstable points. Furthermore, this volume can be chosen using the results of the data-gathering procedure.

An interesting feature of this procedure is that an accurate estimate of the volumes can be made. The number of points necessary to obtain such an estimate can be obtained from the law of large numbers. To be precise, define a sequence of independent Bernoulli Trials as follows:

$$x_k = \begin{cases} 1 & \text{if the k-th prediction is correct} \\ 0 & \text{if the k-th prediction is incorrect} \end{cases} \quad . \quad (2.46)$$

Then, if V denotes the volume in which points are being taken,

$$1/N \sum_{k=1}^{N} x_k \qquad (2.47)$$

is an estimate of the probability of a successful prediction and

$$V \cdot 1/N \sum_{k=1}^{N} x_k \qquad (2.48)$$

is an estimate of the volume of correct prediction.

Let p be the actual probability of success. Then, suppose it is desired to find N such that

$$P\{\bigcap_{n \geq N} \{|\sum_{k=1}^{n} x_k/n-p|>\varepsilon\}\} < \delta \quad . \quad (2.49)$$

Invoking the Strong Law of Large Numbers, the following is obtained [44];

$$N \sim 32/6\ p(1-p)/\varepsilon^2\delta \qquad (2.50)$$

where $\sim$ denotes asymptotic equality.

Also consider the weaker condition that

$$P\{|\sum_{k=1}^{n} x_k/n-p|>\varepsilon\} < \delta \qquad n \geq N \qquad . \quad (2.51)$$

The result for this, again using the law of large numbers, is

$$N \geq p(1-p)/\varepsilon^2\delta \qquad . \quad (2.52)$$

Often estimates based on the strong law are overly pessimistic. Hence, recomputing N for Equation 2.51 using the Weak Law of Large Numbers, the result is

$$N \sim a^2p(1-p)/\varepsilon^2 \qquad (2.53)$$

where $a = \Phi^{-1}(1-\delta)$ with Φ denoting the normal probability distribution.

No matter which estimate of N is used, the actual probability, p, is unknown for our problem. Hence, it must either be estimated or the experiment must be performed until a value of N is used such that

$$|1/N \sum_{k=1}^{N} x_k - p| < \varepsilon \qquad (2.54)$$

with p given by 2.50,2.52, or 2.53 depending upon which estimate is used.

Once this experiment has been performed, the quality of the approximation to the transient stability region can be evaluated. This will yield a quantitative confidence estimate and can be used to verify if an adequate number of directions was used in the data-gathering phase.

2.3 Computer Programs

A FORTRAN IV program has been written which performs the computation necessary to realize the technique that has been described in this chapter. The individual tasks are divided up among a main program and three subroutines. The main program performs the necessary input-output, calls the other routines, and calculates the coordinates of the points to be simulated. The subroutines are responsible for choosing the direction to be searched, simulating an initial condition, and updating the ellipse parameters. The describing equations and flow charts, where appropriate, appear in Appendix I.

A search direction is specified by its direction cosines in the system state-space. If a uniform grid is placed on a unit sphere about the postfault equilibrium in order to specify the directions, the cosines must be calculated prior to run-time, and entered as input as each direction is searched. If the random direction method is to be used, the routine will calculate the directions. The technique used is given by [46] and makes use of the fact that for a multivariate normal distribution, surfaces of constant probability are elliptical. A uniform random point on an N-dimensional unit sphere can be generated using the standard uniform random number generator available on most operating systems. This is accomplished as follows:

1. Generate five uniform random numbers in the interval $[0,1]$, $\alpha_1,\alpha_2,\ldots,\alpha_5$.
2. Form the normal deviate ξ_i according to
$$\xi_i = 12/5(\sum_{k=1}^{5}\alpha_k) - 6 \quad .$$
3. Generate N such numbers, $\xi_1,\xi_2,\ldots,\xi_N$, by repeating steps 1 and 2.
4. The random direction cosines specifying the uniform random point are given by
$$x_i = \xi_i/\sqrt{\sum_{k=1}^{N}\xi_k^2} \quad i=1,2,\ldots,N \quad .$$

Step 2 is from a theorem by Kintchine and is a commonly used method of generating random numbers with a normal distribution [45]. For completeness, this entire procedure is also outlined in Appendix I-C.

Once the direction of search has been specified, the dual-mode procedure given in Section 2.3.1 is used to locate the boundary along it. The parameters specified by the program user are the initial step size, the terminating step size, and the distance ratio of the Fibonacci search. Typical values used were 100.0,10.0, and 5/8 respectively. The search is carried out within the main program and a flow chart appears in Appendix I-A.

A simulation of the system using Equations 1.5 is accomplished by one of the subroutines. A fourth-order Runge-Kutta technique is used to start the simulation and Milne's Predictor-Corrector is used for continuation. The parameters that must be specified for the simulation other than the initial conditions are the integration step size and the permissable error for the predictor-corrector procedure. For the problems worked in this thesis, the step size was 0.01 seconds and the error was 0.005 degrees per second. The governing equations are listed in Appendix I-B.

One approximation was inserted in order to conserve computer time. Since the majority of the computation is involved with simulating Equations 1.5, a large number of sine functions must be evaluated. Conventionally, this function is evaluated by summing a number of terms of the infinite series for sin(x). The number of terms may be fixed or variable based upon the incremental contribution of each succeeding term. In either case, many exponentiations and multiplications must be performed. To circumvent these, a table-look-up procedure with linear interpolation between points was used. For this thesis the table size was fixed at ten points.

For most of the examples worked, the ellipse parameters were updated using the recursive form of the stochastic separation algorithm, Equations 2.38 and 2.40. The routine necessary to accomplish this is a straightforward implementation of these equations and was called after each direction using the points tested along it. The initial R-matrix was chosen to be a constant times the identity matrix and the initial α-vector used was a sphere of radius one about the equilibrium point from which the search emanated. When the Ho-Kashyap Algorithm was used in the forms given previously, the stability data was punched on cards and fed separately to a program implementing the appropriate equations. The initial β-vector was usually chosen to be a vector of appropriate dimension with $b(i)=1.0$. If Equation 2.23 was used, ε was arbitrarily chosen at 0.001.

In the above programs, the procedure is implemented digitally. If a hybrid computer installation is available, the data-gathering procedure could be performed on it with some decided advantages. The simulation could be implemented on the analog portion and possibly run in a fast-time mode. Hence, savings in time and accuracy could be made. The digital portion of the computer would have to calculate the next initial conditions and/or direction to be searched. Since these could be performed simultaneously with the simulation, another savings in computation time could be realized. Depending upon the actual timing sequences, the digital computer could possibly calculate two sets of initial conditions; the correct set could then be given to the analog computer depending upon the stability quality of the previous simulation. The results could be stored in bulk memory and the actual separation of data points performed at some convenient later time. In practice the speed of such an implementation would probably be limited by the analog-to digital converter. However, considerable savings in time and cost could be realized.

2.4 Summary

In this chapter a new technique for obtaining an approximation to the transient stability region for power systems has been presented. This technique has several advantages over existing Liapunov Function techniques which have been stated. The procedure is implemented via a linear search procedure using system simulation as an evaluating tool. Pattern separation is then used to obtain a quadratic surface which serves as the approximation.

A FORTRAN IV program has been written to automatically carry out the described procedure for the multimachine case. A flow chart of this program is given in Appendix I. In the next chapter several examples are worked using the technique and the results evaluated.

CHAPTER III

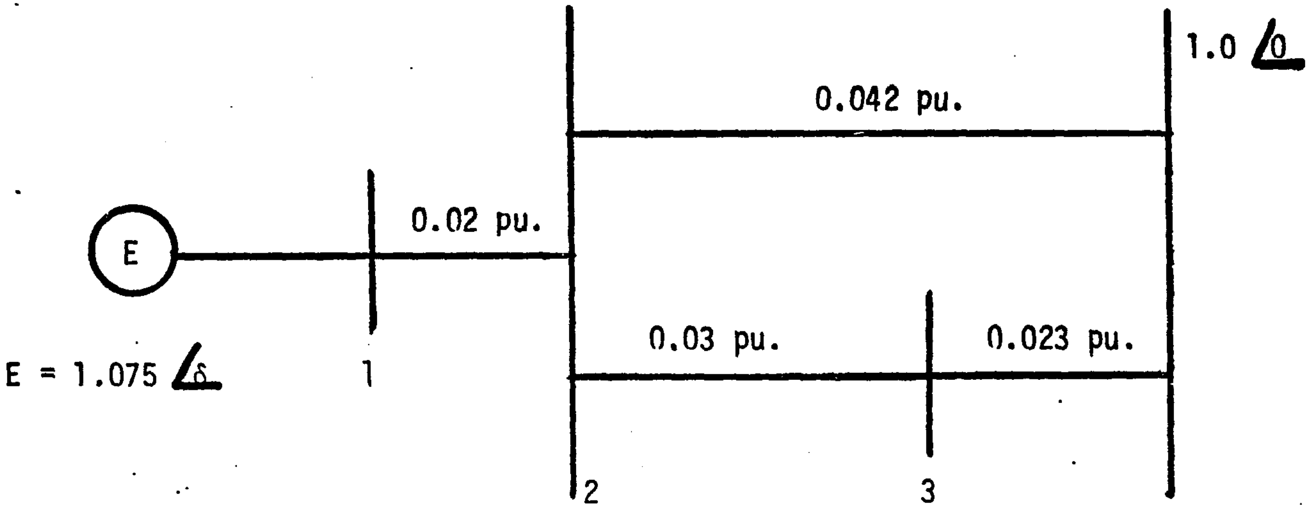
EXAMPLES

3.1 Single-machine Example

It is traditional to evaluate stability techniques on a system consisting of a single machine delivering power to an infinite bus. Since this is the simplest power system, computations are reduced. The state space for this system is of low enough dimension to make graphical interpretations possible. Besides these advantages, the study of this system affords an opportunity to gain insight into the stability problem.

Several single-machine examples are considered here also. The purpose is to take advantage of the simplicity in order to demonstrate the proposed technique and to test its feasibility. In each example the transient stability boundary is generated and several clearing times are evaluated.

The chosen system is shown in a one-line diagram in Figure 3.1. This is an equivalent system extracted from the Cleveland Electric Illuminating Company system. The machine is a round-rotor synchronous generator with a maximum power rating of six-hundred and eighty megawatts. The transmission system consists of two parallel lines between the machine and the infinite bus. The network parameters were chosen so that the resulting power-angle curve agreed with data obtained from a stability study conducted



Machine Characteristics
Inertia Constant M = 0.121 $\frac{\text{mw.-sec.}^2}{\text{mva.-rad.}}$
Transient Reactance x_t = 0.0429 pu.

Figure 3.1 Single-machine System

by simulating the entire system.

In order that the proposed technique can be used, a postfault network must be chosen. Since faulted lines are removed from the system by protective equipment, the postfault network is determined by the choice of a fault location. Certain of the possible locations can be quickly discarded; any line which,upon clearance,completely breaks the path between the machine and the infinite bus must be so treated. If such a fault occurs, the machine will be rapidly disconnected from the system and the idea of two types of postfault trajectories is meaningless. As a result, there are only two choices remaining, a fault on one of the parallel lines. Of these two, removal of the 0.042 per unit line will result in the smallest maximum power transferable to the infinite bus since the impedance of the remaining line is larger. This difference is not, however, so significant as to warrant the use of a second postfault network.

One system parameter is still unknown, the machine power level. For this example, maximum power will be used as this represents a "worst" condition.

The swing equation for the postfault system is

$$0.121\ \ddot{\delta} = 6.8 - 9.28\ \mathrm{SIN}(\delta) \qquad . \quad (3.1)$$

Solutions to this equations were generated using a digital simulation. For the first example, initial conditions were chosen according to the procedure outlined in Figure 2.2. The pattern set is depicted in Figure 3.2 along with the Liapunov Function of Gless [20] which is an exact mathematical boundary for this problem.It is noted that,consistent with our previous discussion, only points near the boundary were used. A sufficient number of these points were gathered so as to trace out the boundary.

By comparing the Liapunov boundary against the pattern set in Figure 3.2, one notices that there are stable patterns that lie outside the Liapunov boundary. This fact is demonstrated at the right end of the boundary for negative initial velocities. This is due to the fact that simulations are terminated after one second and solutions beginning in this region may not have become unstable by this time,although,if continued longer, they would have exceeded one-hundred and eighty degrees in angular value. The Liapunov function is the boundary of the mathematical domain of attraction and must classify these initial conditions as unstable. Terminating the solutions after one second defines a kind of practical stability since governor action will have changed the mechanical input to the machine by the end of this period and this fact will aid in maintaining synchronism. This difference between our definition of stability and the conventional one must be tolerated for the sake of realistic computational requirements and will manifest itself again.

The pattern separation was performed using the Ho-Kashyap algorithm. An ellipse was obtained and is shown with the pattern

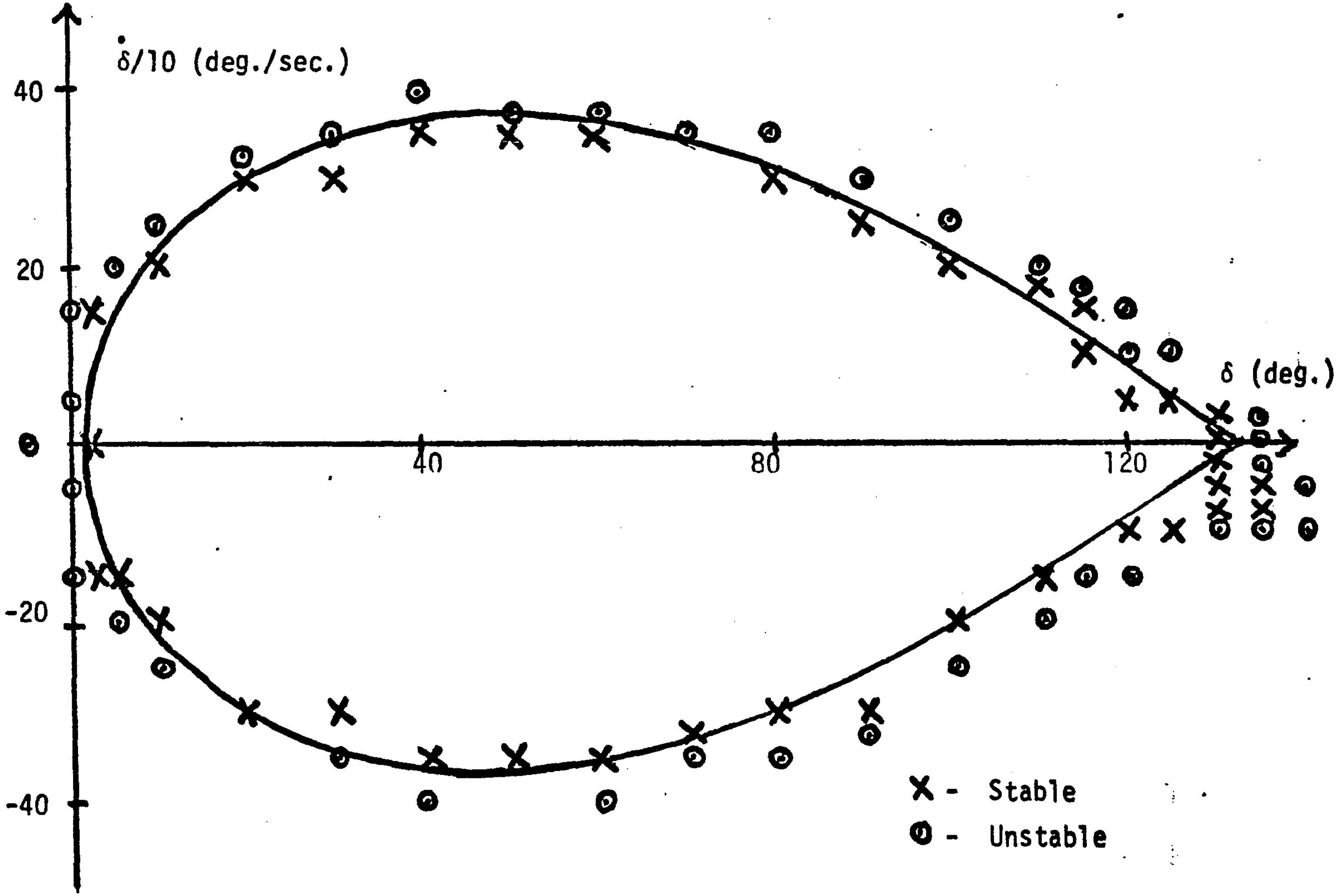


Figure 3.2 Single-machine Data

set in Figure 3.3. The ellipse correctly classifys 59.5 per cent of the training set; this recognition rate is not high, but since only boundary points were used for training, this figure does not represent a good measure of the boundary approximation.

To provide a measure of the approximation, the ellipse is shown with the Liapunov function in Figure 3.4. Qualitatively, the boundary has been approximated rather well. However, large areas with unstable solutions are included within the approximation for angular values between sixty and one hundred and thirty degrees. Now, in the operation of power systems this is undesirable. As a result the weighted version of the Ho-Kashyap algorithm was used to perform the training and its separating surface appears in Figure 3.5. It is seen that the amount of unstable area that is inclosed is reduced at the expense of being conservative along portions of the boundary. It can also be seen that the orientation of this ellipse has not substantially changed from that in Figure 3.4.

To provide an evaluation measure for the obtained ellipses, the critical clearing times were evaluated for several faults on this system. This is accomplished by simulating the system during a fault and noting the time when the trajectory crosses the boundary. The theoretical clearing times can be evaluated by the equal-area-criterion, or equivalently, by noting when the Liapunov boundary is crossed. Comparison is made, where possible, with

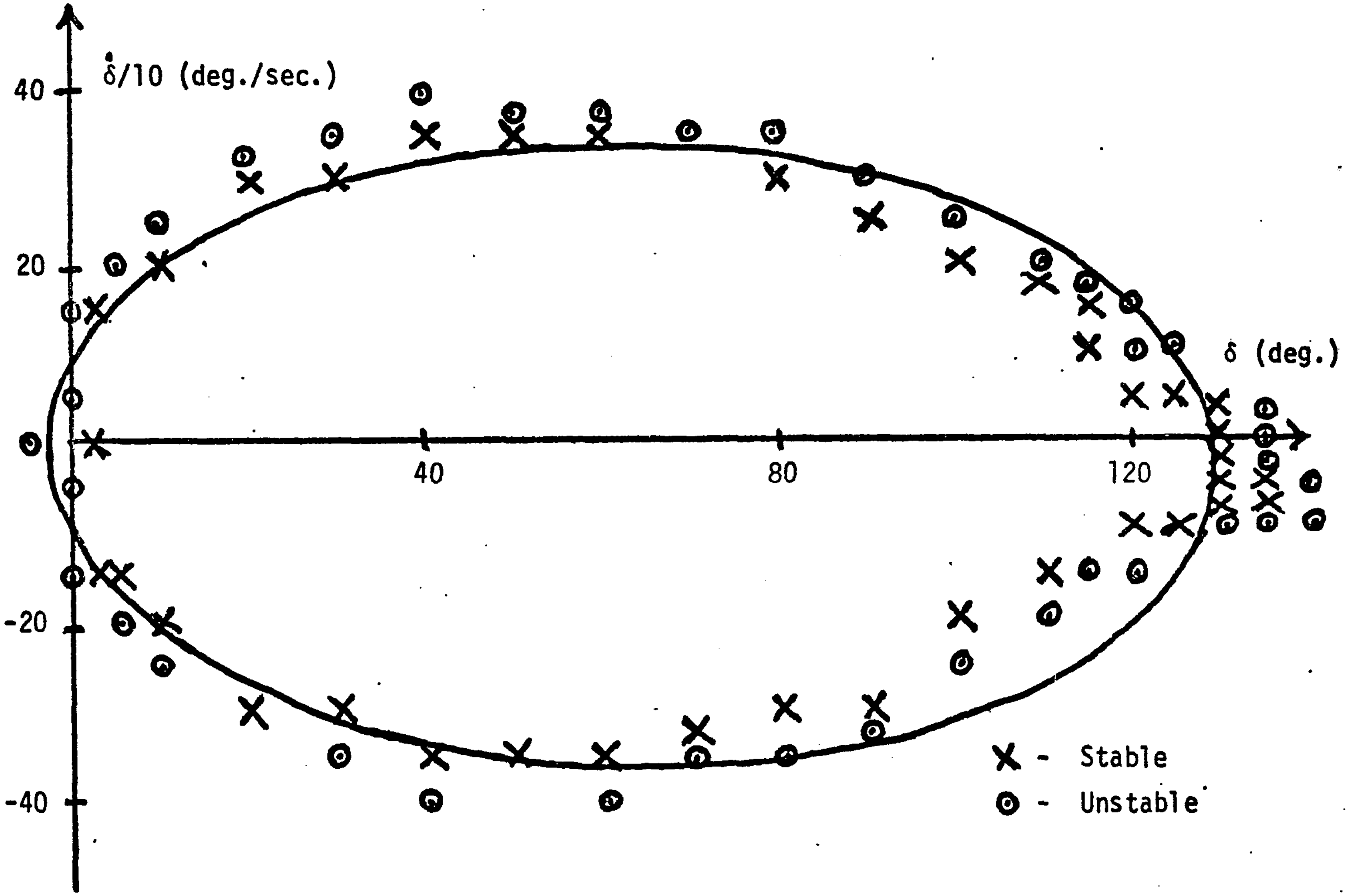


Figure 3.3 Ho-Kashyap Separation

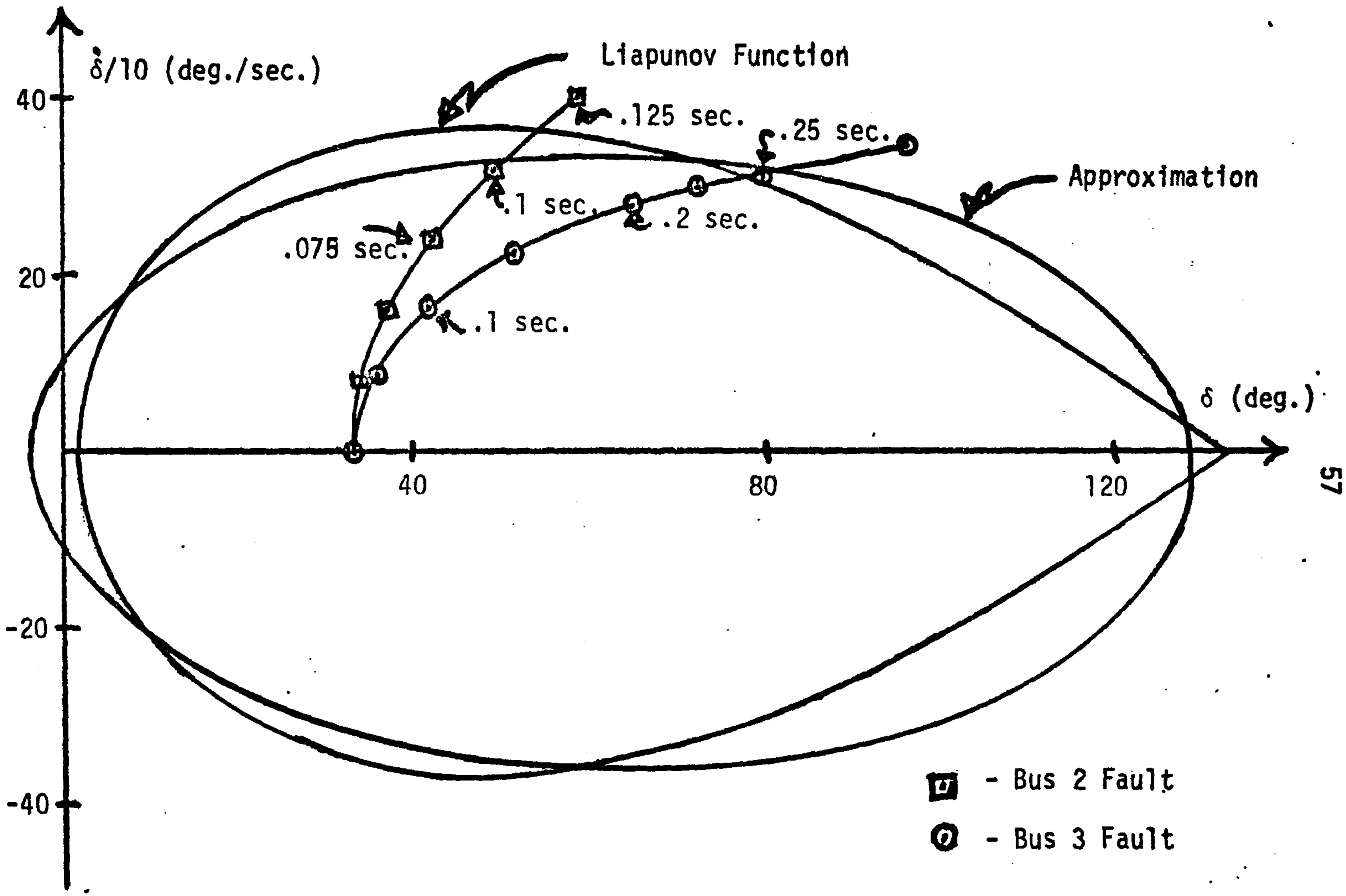


Figure 3.4 Clearing Time Estimation - Penalty Ratio = 1.0

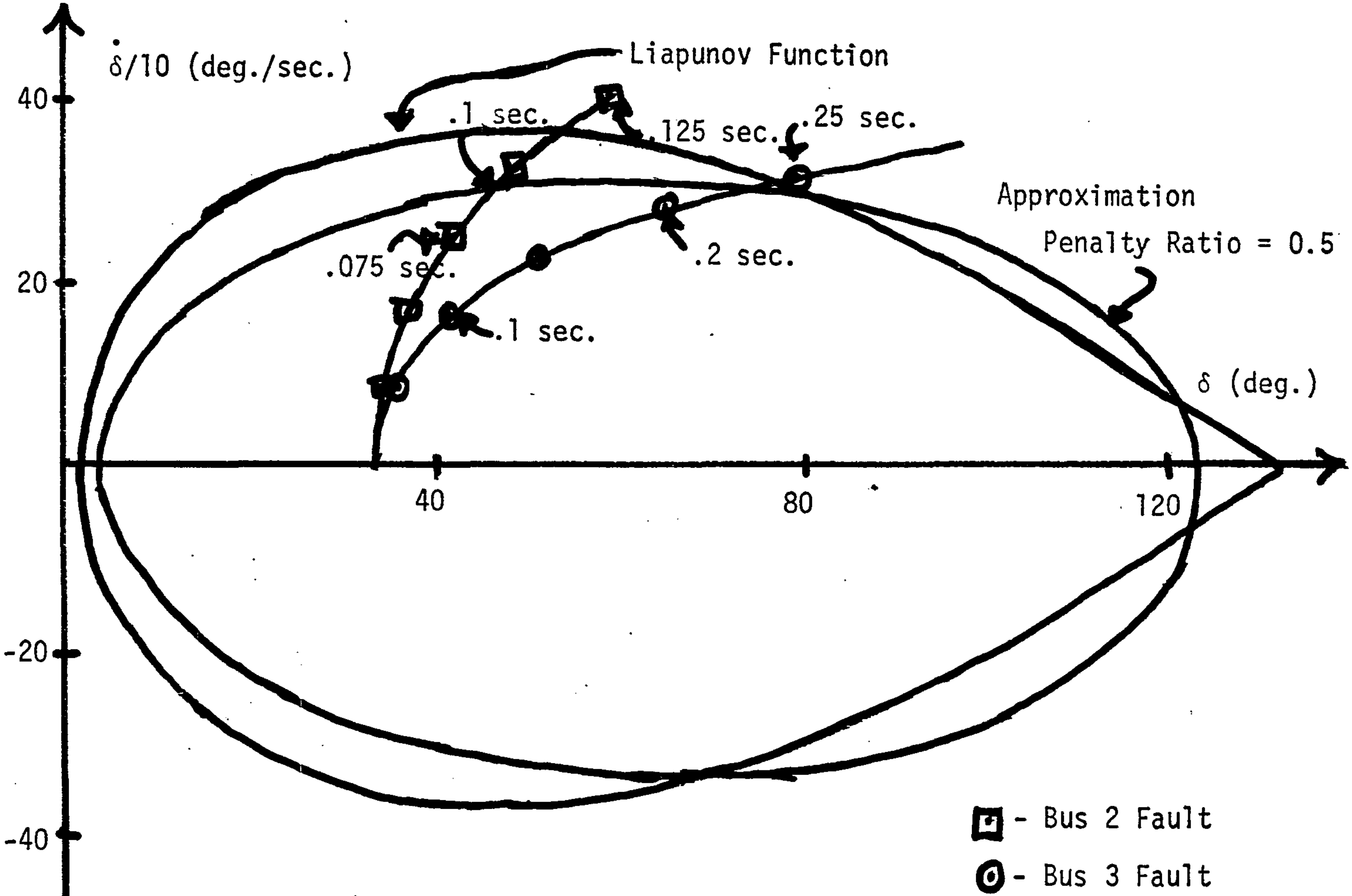


Figure 3.5 Clearing Time Estimation-
Weighted Ho-Kashyap Algorithm

actual clearing times as related by Illuminating Company engineers.

For the system under consideration, clearing times were determined for faults at Bus 2[1] resulting in the switching out of either line. A fault at Bus 3 was also investigated. The trajectories for faults on these busses appear in Figures 3.4 and 3.5. Table 3.1 gives the obtained times and compares them with the theoretical value and with known values where they are available.

The above procedure was performed to try and assess the ability of the pattern recognizer to achieve an acceptable approximation. As noted in Chapter II, the data-gathering procedure that was followed is not amenable to problems with a higher dimensional state space. A search scheme was presented in Chapter II which should overcome some of the dimensionality problems. In order to demonstrate its advantages and shortcomings, this procedure has also been used to solve the one-machine problem of Figure 3.1.

For the problem being considered, the random procedure for gathering points was used. A first-mode step size of one hundred units and a terminating step size of ten units were used. The search procedure was performed using a pseudo-random number generator. Angles were chosen uniformly in the interval $[-\pi,\pi]$ and used to determine directions of search.

[1] See Figure 3.1 for bus numbers.

Fault Point	Clearing Times (sec.)			
	Estimated		Theoretical	Actual
	Normal	Conservative		
Bus 2	0.105	0.095	0.115	0.100
Bus 3	0.258	0.227	0.242	0.225

Table 3.1 Single-machine Clearing Times

Figure 3.6 displays the directions that were searched after duplications were removed. Duplication does not imply exactly equal directions, but directions that are so close to one already shown that they do not convey much additional information. The search was terminated after 120 directions.

Note that in Figure 3.6 the directions hardly appear uniform. This is due to the fact that the $\dot{\delta}$ scale is compressed. The actual directions were uniform in the δ-$\dot{\delta}$ plane. The ordinate is compressed in the figure for ease in plotting. It does, however, bring up the point that uniformity of direction depends upon the units used and care should be taken to make them meaningful to the physical system and to assure that they allow a good representation of the boundary by the searched directions.

Figure 3.6 again exhibits the stability phenomenon mentioned previously, i. e. simulations are terminated after one second, and hence initial conditions of approximately 170 degrees and -200 to -400 degrees per second are stable despite being outside the domain of attraction of the equilibrium. This occurs because δ does not exceed 180 degrees within the alloted time period. Since in the multimachine case, the computation time will become prohibitive if this time is raised, and since it will be difficult to identify these directions, this phenomenon will be ignored.

The pattern separation was performed using the stochastic algorithm and the deterministic algorithm. Every tested point was used as data for the former and boundary points for the

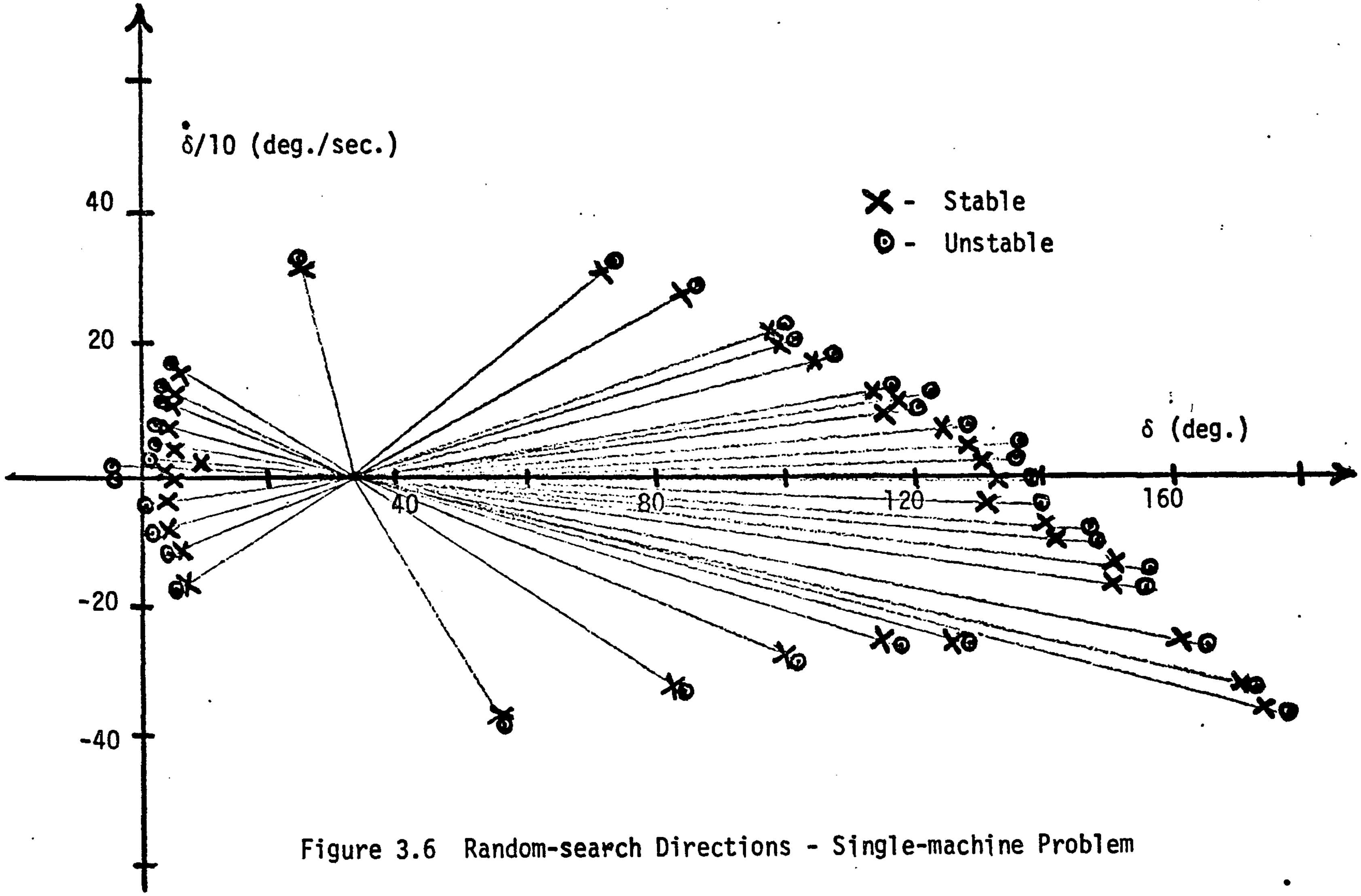


Figure 3.6 Random-search Directions - Single-machine Problem

latter. These separations are depicted along with the boundary points in Figures 3.7 and 3.8 respectively. It can be seen that the stability definition phenomenon does influence the separation by drawing the ellipse toward the points exhibiting it. To gauge the effect of these directions, they were removed from the data set and the deterministic separation performed again. Figure 3.9 shows the obtained curves, and after comparing Figures 3.8 and 3.9, the influence becomes obvious. It is noted, however, that the difference is not drastic.

Each of the algorithms performed equally well on its data set. The stochastic formulation classified 83.7 per cent of its training set correctly with about an equal number of stable and unstable patterns being misclassified. The deterministic algorithm with all directions included classified correctly 79.8 per cent of its training patterns when both types of patterns were weighted equally. With a penalty ratio of 0.5 the recognition rate dropped to 54 per cent. However, with equal weighting about one-half of those patterns which were misclassified were unstable while by penalizing the inclusion of such patterns inside the separating surface by a factor of two,produces a separation where only 4.9 per cent of those missed are unstable. The ellipses of Figure 3.9 classified 75.7 per cent and 82.5 per cent correctly for penalty ratios of 1.0 and 0.5 respectively. For a ratio of 1.0,64 per cent of those misclassified were unstable while for

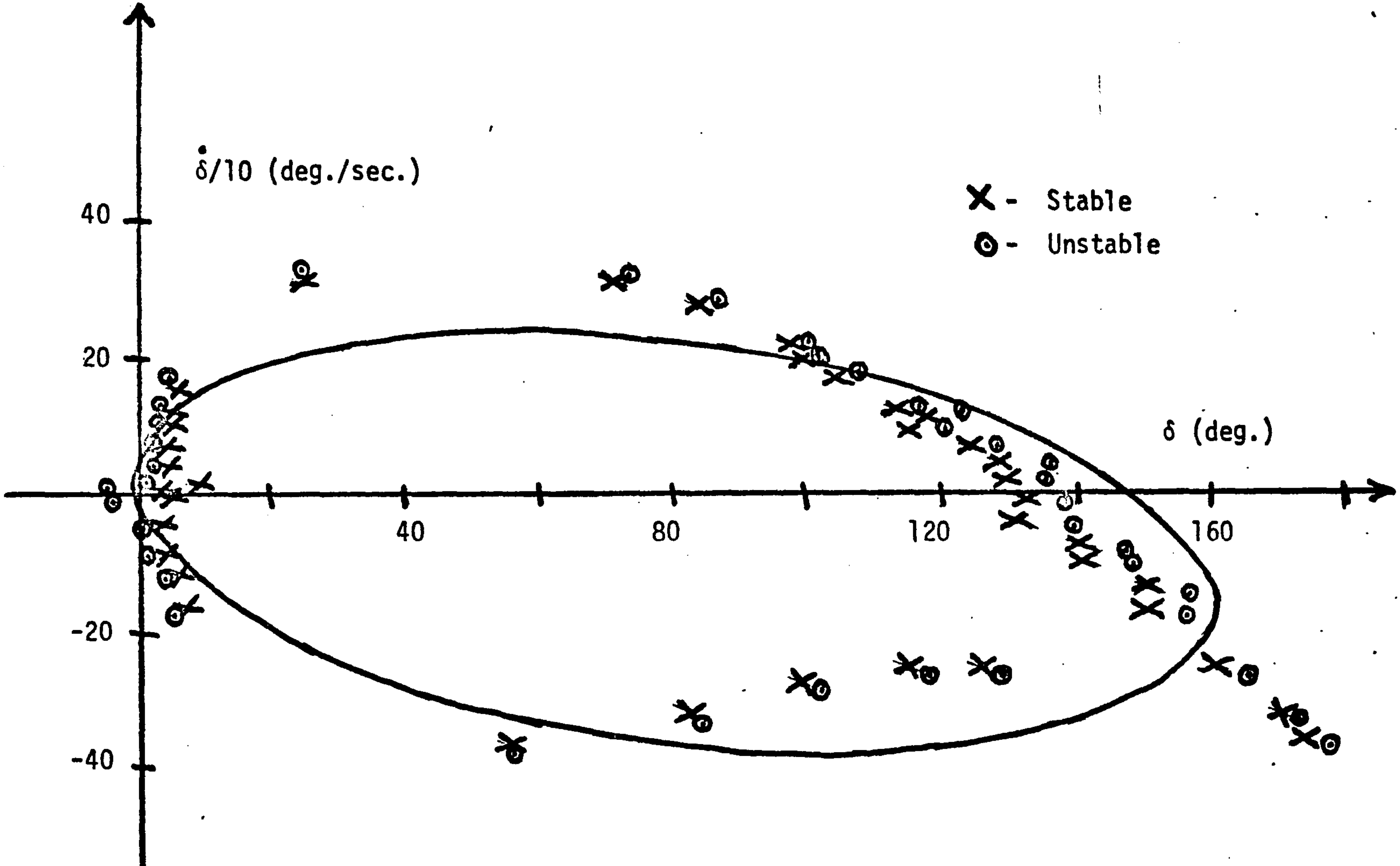


Figure 3.7 Random-search, Stochastic Separation

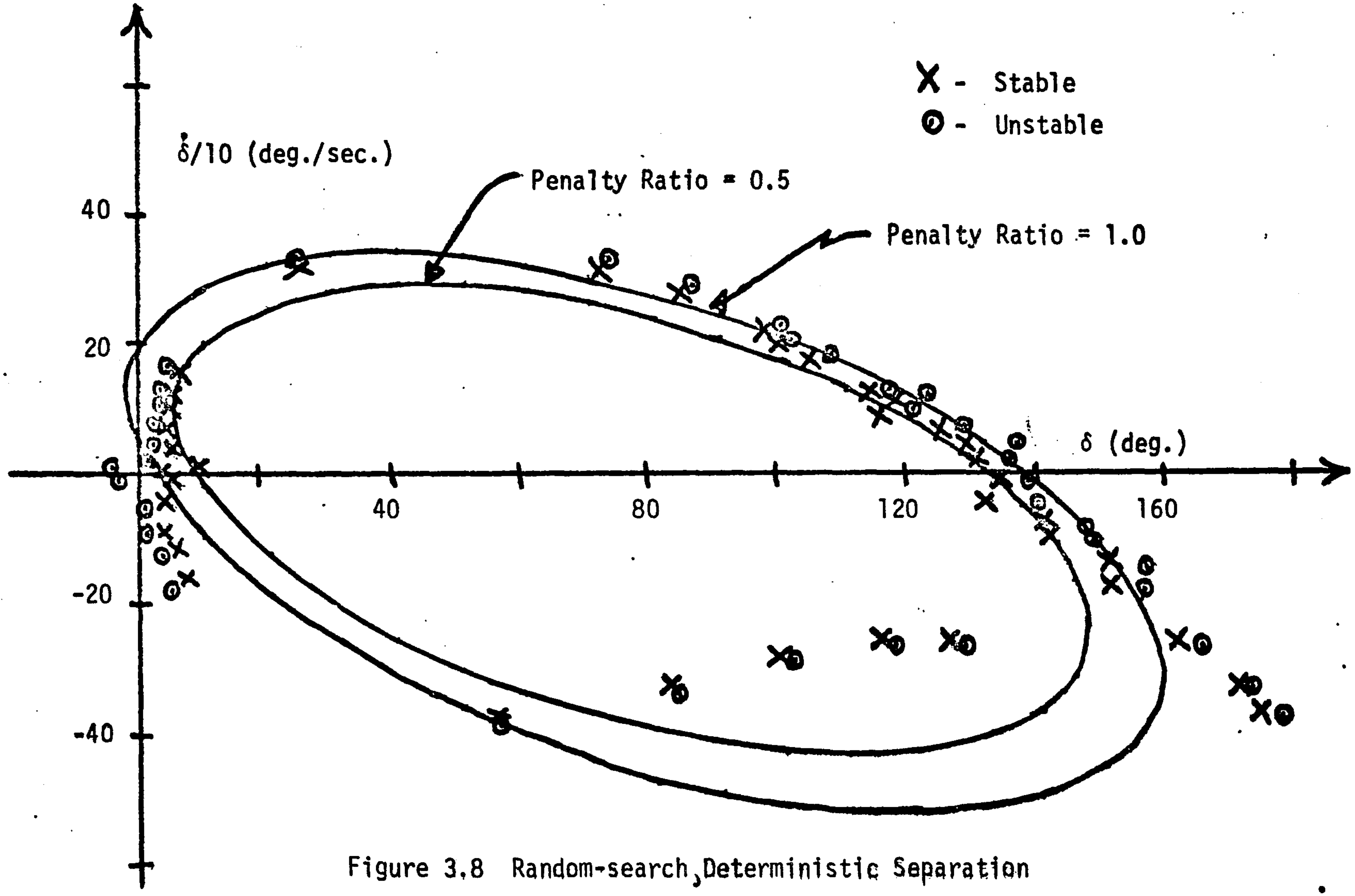


Figure 3.8 Random-search, Deterministic Separation

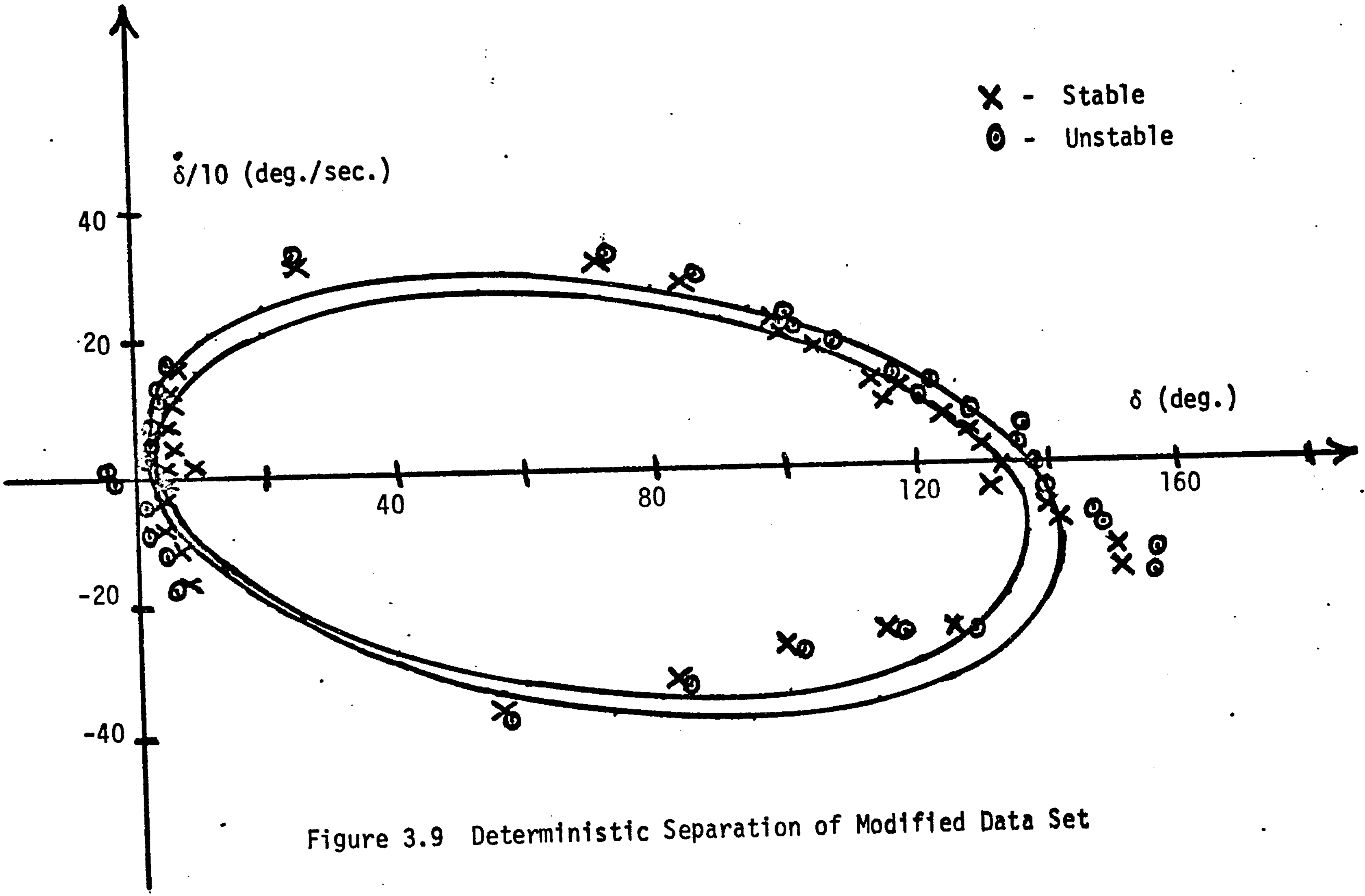


Figure 3.9 Deterministic Separation of Modified Data Set

one of 0.5, 13.4 per cent were unstable.

Comparison of Figures 3.7-3.9 reveals that all separations yield similar ellipses. The curves have almost identical centers. The orientation of the deterministic separations of Figures 3.8 and 3.9 differ slightly from the stochastic separation of Figure 3.7. The minor axes are almost identical for all curves obtained with equal weighting of the patterns as are the major axes for the equally-wieghted, full-set pattern separations shown in Figures 3.7 and 3.8. In short, there seems to be little differences between the algorithms. One might prefer the conservative separation obtained with a penalty ratio of 0.5 from a practical point of view, but there is not a large difference here either and the inherent conservativeness of the model mentioned in Chapter I might overshadow these differences entirely.

Figure 3.10 depicts the conservative Ho-Kashyap separation with the fault trajectories for Bus 2 and Bus 3 of Figure 3.1. It is seen that the estimates for the clearing times are 0.85 and 0.18 seconds respectively while the theoretical times from Table 3.1 are 0.115 seconds and 0.242 seconds.

The proposed technique has performed well in both of these examples. From Figures 3.3, 3.5, 3.7, and 3.8 it can be said that the elliptical separation does provide a reasonable approximation to the domain of attraction of the equilibrium. Differences among the algorithms are slight and clearing times obtained

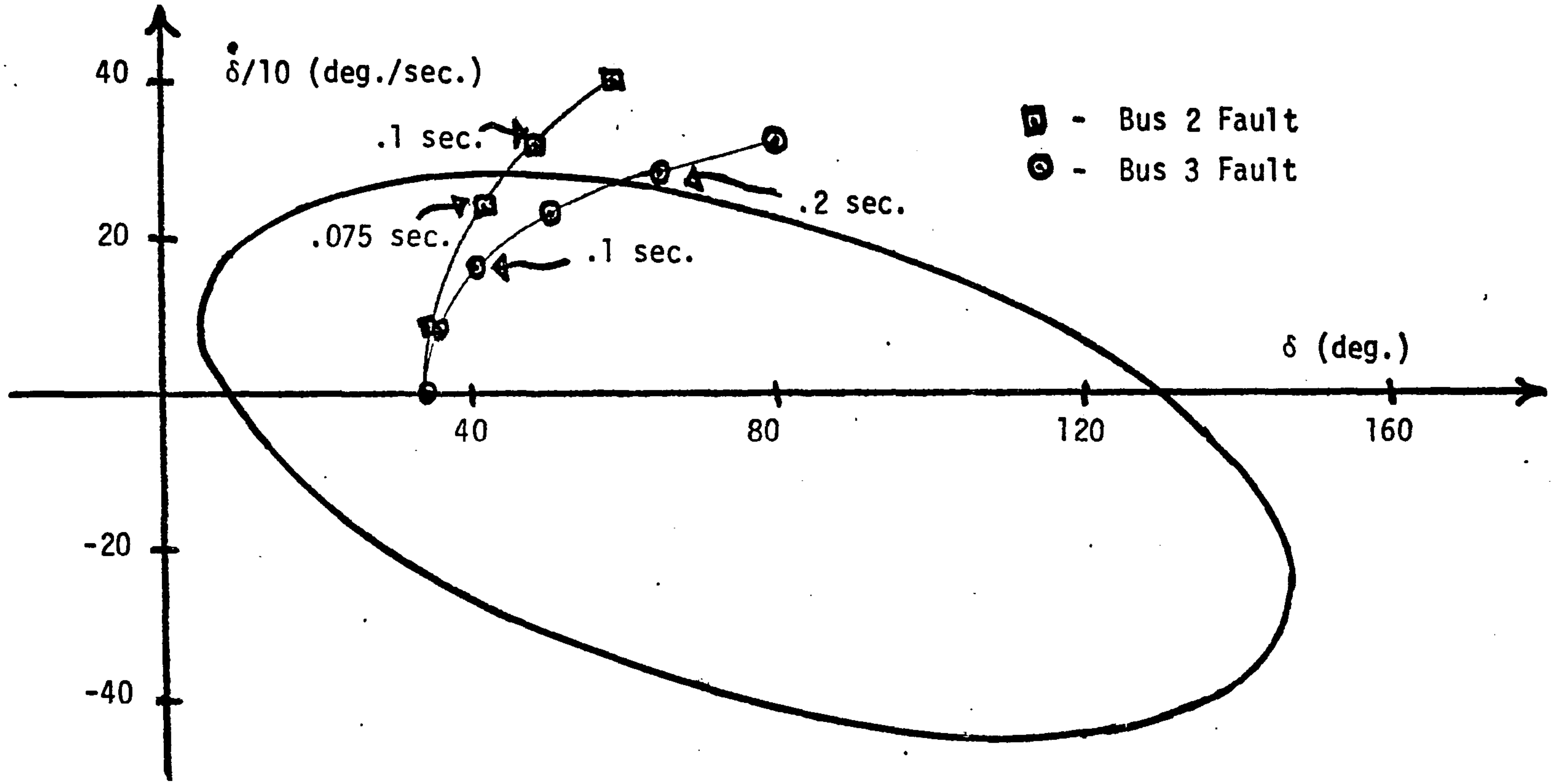


Figure 3.10 Random-search Clearing Time Estimation

with the separating surfaces are close to actual values and indeed can be used as conservative estimates of these quantities. Computationally, it has emerged that bulges in the surface can be produced for regions with negative initial velocity due to the one second maximum solution time. This effect must be tolerated in order to keep the computation time reasonable and because these directions are not easily recognizable.

Obtaining the transient stability boundary for a classical one-machine system is a straightforward problem; a simple equation for the true boundary exists and one would never resort to the technique that is demonstrated here to locate the boundary for such a system. However, the technique does seem to be feasible for application to more complicated systems where its advantages first appear. In the remainder of this chapter the ability of this technique to approximate the transient stability region of such a system is evaluated.

3.2 Three-machine Example

The three-machine system chosen for study is again taken from the Cleveland Electric Illuminating Company system. The system includes the machine used in Section 3.1. Machines 1 and 2 are actual generators while machine 3 is an equivalent representation of a large system. The interconnecting lines were chosen with consideration to the actual transmission system, and a resistive load was added. The system is depicted in Figure 3.11.

It is to be noted that machine 1 is the one used in Section 3.1. There is not an exact correspondence between the impedance values of the lines emanating from this machine in Figure 3.11 and those in Figure 3.1. This is due to the necessity of making approximations when forming the models. The 0.0393 pu. line of Figure 3.11 represents the same line as the 0.0402 pu. line assumed removed when generating the boundary in Section 3.1.

3.2.1 Full-power Allocation

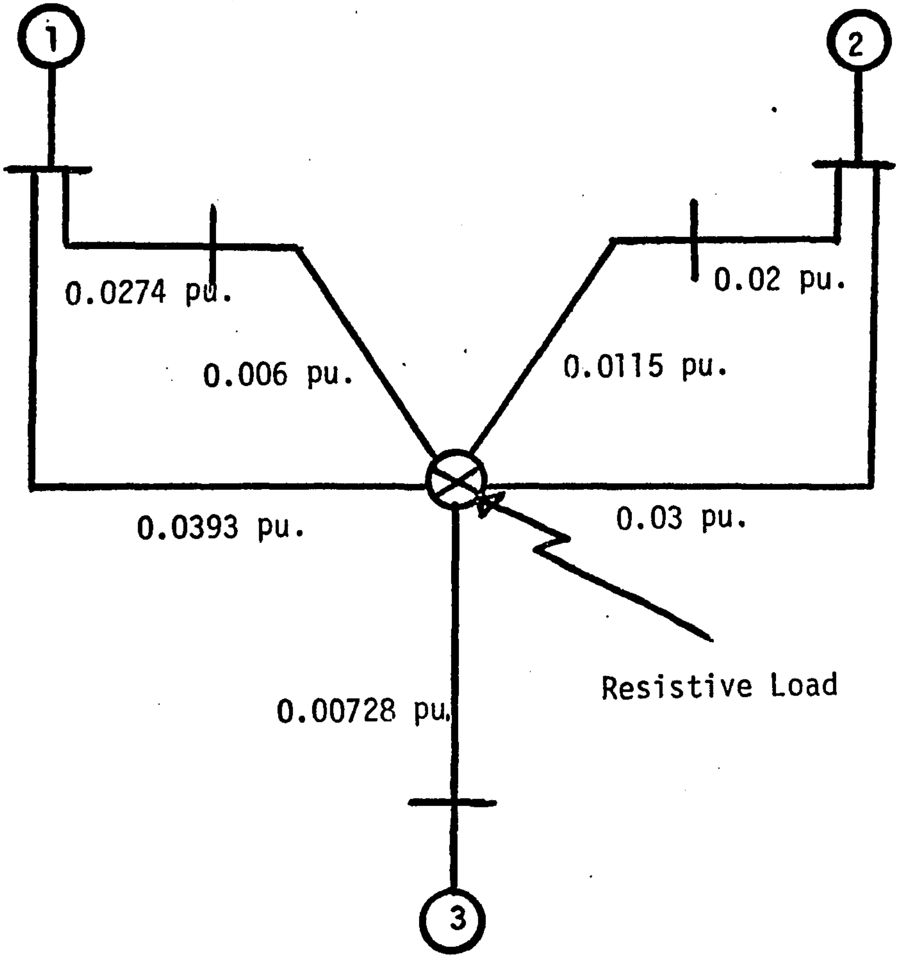
For this example, near capacity power inputs were assigned to each machine. These were 6.28, 6.01, and 14.04 pu. on Machines 1, 2, and 3 respectively. A single postfault network was chosen; the network consisted of the system of Figure 3.11 with the 0.0393 pu. line removed. This network was chosen so that some comparison can be made with the results of Section 3.1. It is noted, however, that removing the 0.0334 pu. line between machine 1 and the load allows less power transfer to the load than the fault chosen and should produce a slightly lower critical clearing time.

For this analysis, machine 3 was chosen as a reference. Thus, if we define

$$\begin{aligned} x_1 &= \delta_1-\delta_3 \\ x_2 &= \delta_2-\delta_3 \end{aligned} \quad , \quad (3.2)$$

the state vector is given by

$$\underline{x} = \{x_1, \dot{x}_1, x_2, \dot{x}_2\} \quad . \quad (3.3)$$



Machine Characteristics :

Machine No.	Inertia Constant	Transient Reactance
	$\frac{\text{mw.-sec.}^2}{\text{mva.-rad.}}$	pu.
1	0.1348	0.0582
2	0.1180	0.0492
3	0.4480	0.0202

Figure 3.11 Three-machine System

The prefault equilibrium is located at $\{1.0^{\circ},0,-.8^{\circ},0\}$ and the postfault stable equilibrium is at $\{9.6^{\circ},0.0,-.59^{\circ},0.0\}$. There is an unstable equilibrium at $\{142^{\circ},0.0,176.5^{\circ},0.0\}$.

The data gathering procedure was performed using an initial step size of 100 units and a terminating step size of 10 units. The separation was performed using the recursive stochastic procedure of Equations 2.38. After 150 directions, the separation was also performed using boundary points with the Ho-Kashyap alogrithm for equally-weighted patterns, Equations 2.23. There was little difference between the obtained coefficients. The search was then continued for another 100 directions and since the coefficients did not change appreciably, the ellipse obtained with the stochastic formulation was used as the boundary estimate. These sets of coefficients are given in Table 3.2.

The data-gathering procedure consumed little computer time for this example. The procedure tested a total of 1601 points using a total of 12 minutes and 23 seconds on an IBM 360/65. Thus, an average of 6.4 points were tested per direction, and approximately 3 seconds were consumed on each direction.

It is difficult to study the four-dimensional solid describing the boundary. However, some planar projections are drawn in Figures 3.12-3.14. Figure 3.12 depicts the x_1-$\dot{x}_1$plane along with the boundary found previously using the one-machine model. It is expected that the three-machine boundary projection should be

Coefficient Number	Deterministic	Stochastic	
	300 Points	150 Directions	250 Directions
1	0.5010	0.5010	0.4800
2	0.0153	0.0195	0.0155
3	0.2920	0.3326	0.3350
4	0.00735	0.0097	0.0105
5	0.0205	0.0244	0.0264
6	-0.2880	-0.2800	-0.2740
7	-0.0437	-0.0397	-0.0210
8	-0.0060	-0.0031	-0.0044
9	-0.0043	-0.0045	-0.0063
10	0.0200	0.0265	0.0130
11	-0.3800	-0.3960	-0.3580
12	0.0238	0.0155	0.0097
13	-0.0464	-0.0614	-0.0641
14	0.0290	0.0330	0.0234
15	-0.5060	-0.0625	-0.0474

Table 3.2 Separation Coefficients- Full-Power Allocation

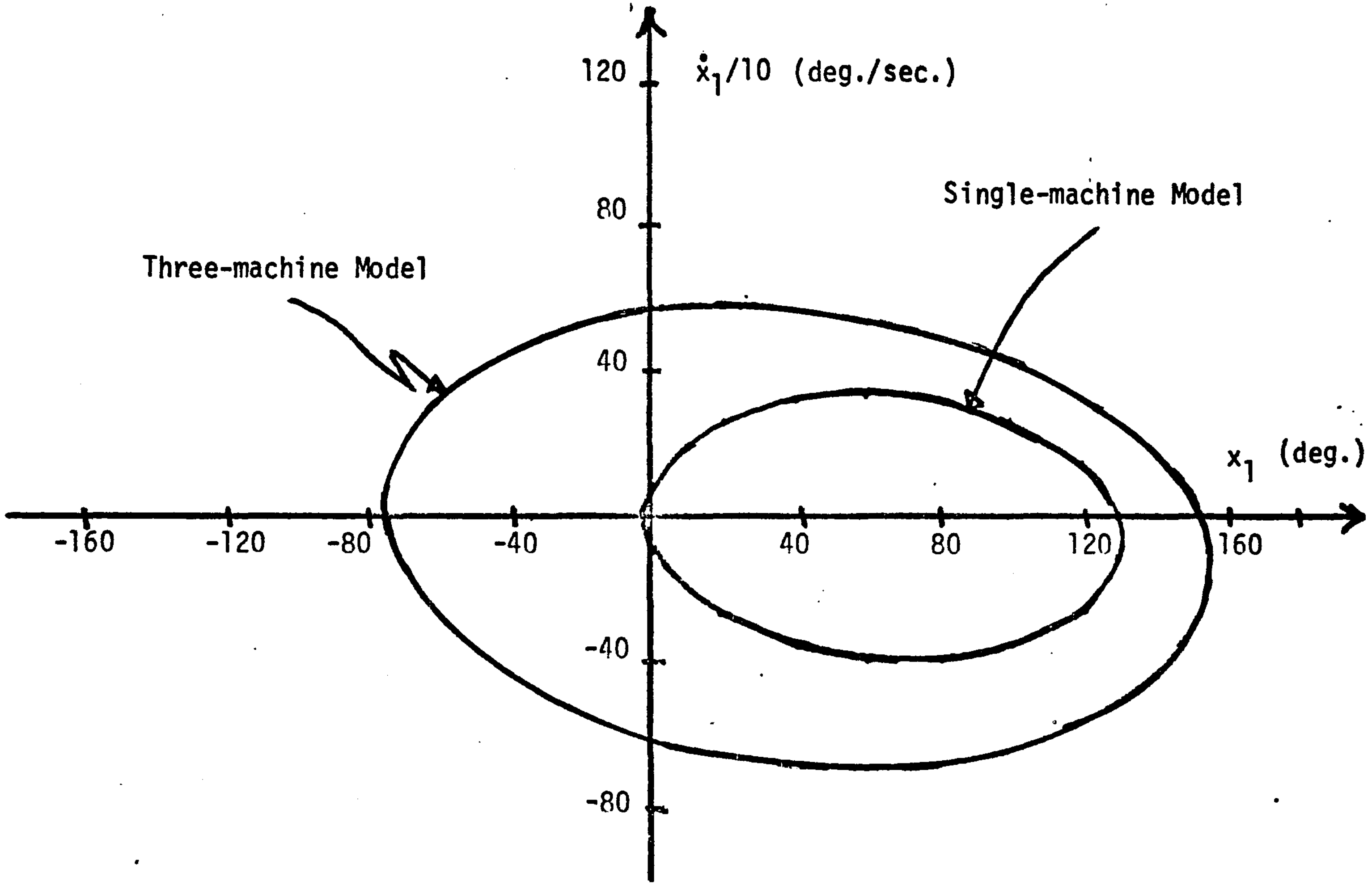


Figure 3.12 x_1-$\dot{x}_1$ Projection of Boundary Approximation- Full-Power Allocation

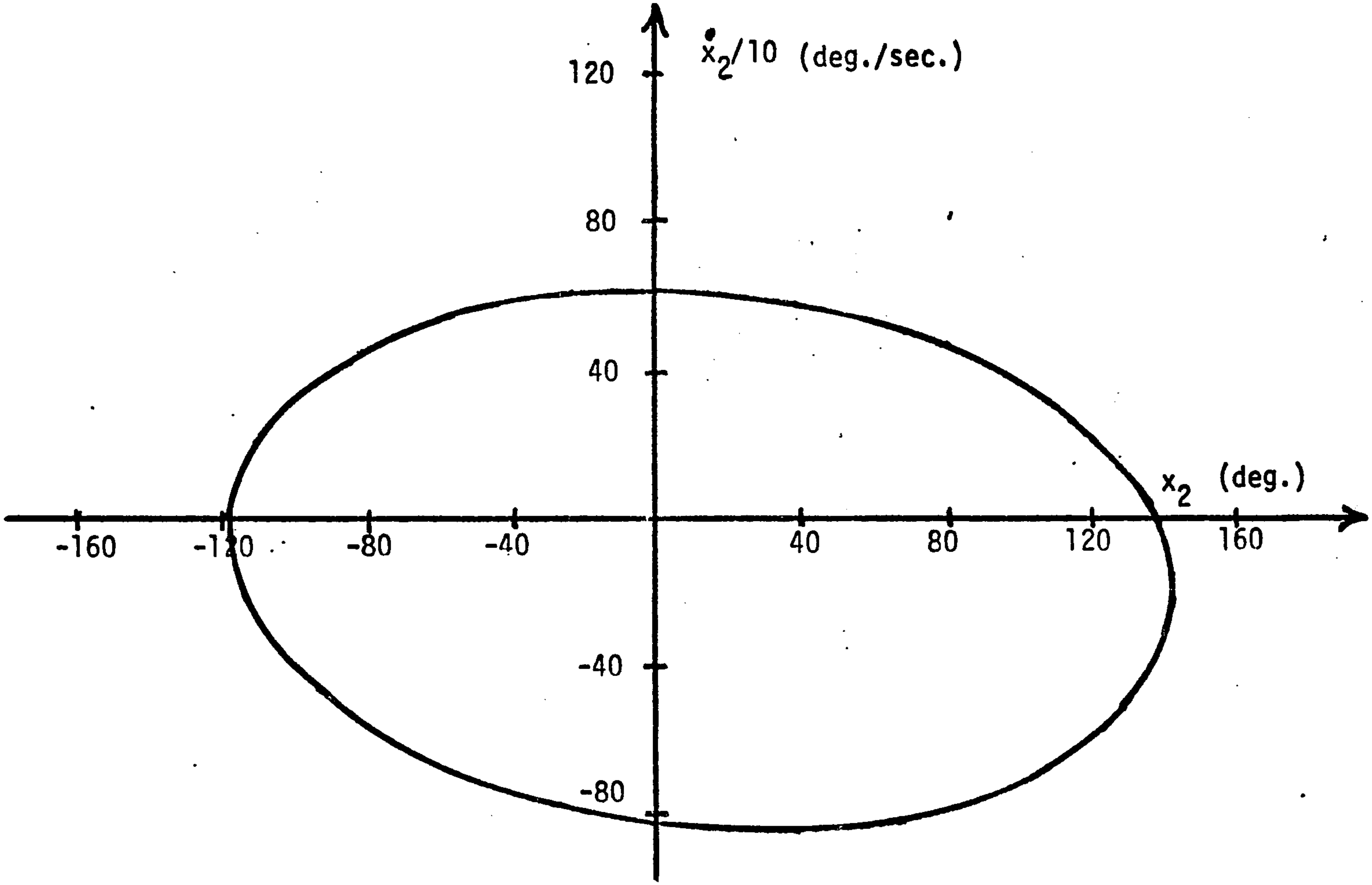


Figure 3.13 x_2-$\dot{x}_2$ Projection of Boundary Approximation- Full-Power Allocation

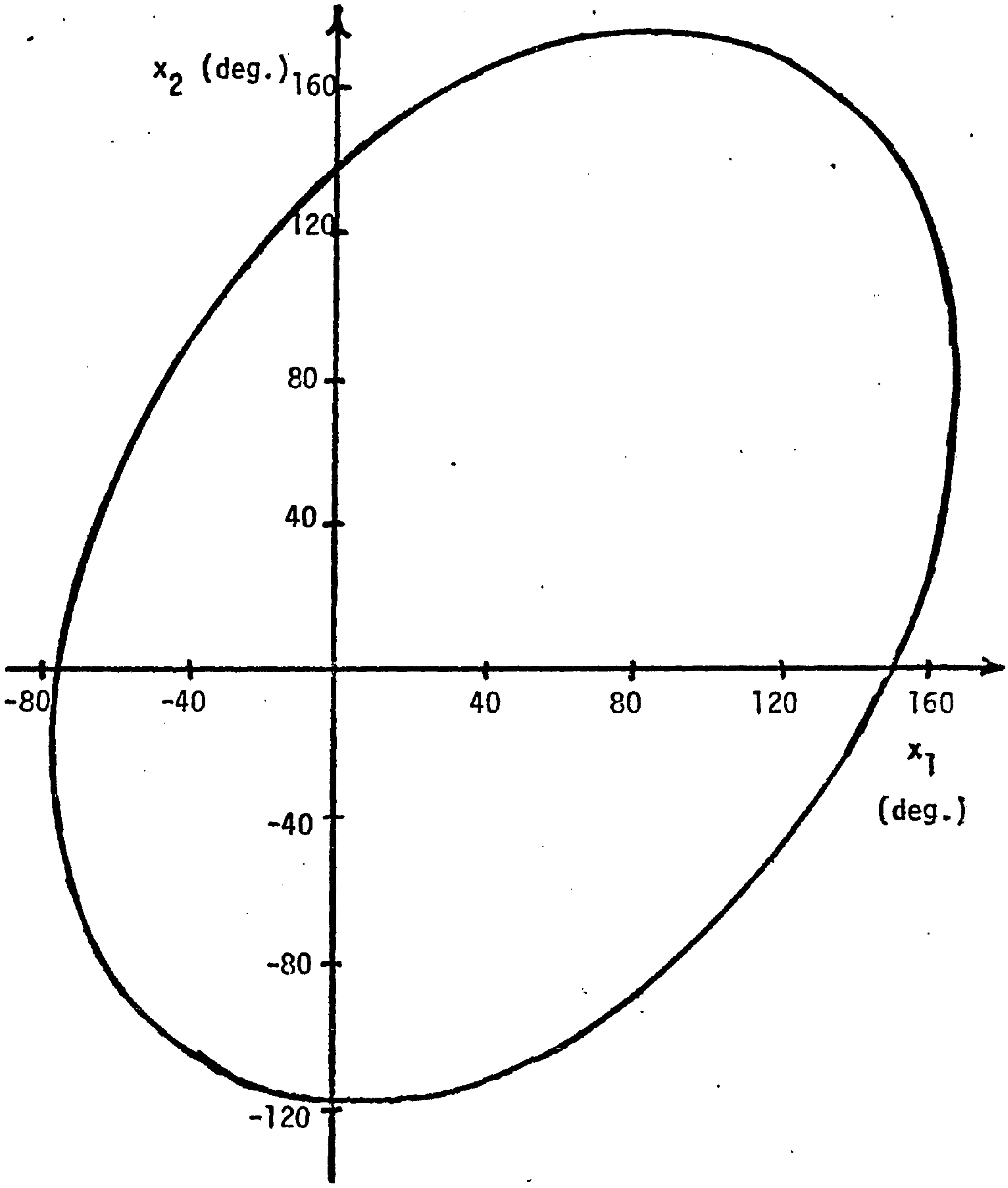


Figure 3.14 x_1-x_2 Projection of Boundary Approximation-Full-Power Allocation

larger for several reasons. First, the reference machine, machine 3, is not a rigid bus as before and hence can swing with machine 1. Second, the power allocation is smaller in this example, and less kinetic energy is absorbed for the same angular deviation from the equilibrium. Figure 3.13 shows the x_2-$\dot{x}_2$ projection and Figure 3.14 depicts the x_1-$\dot{x}_2$ plane.

The critical clearing time was evaluated for a fault grounding the output bus of machine 1 and resulting in the switching out of the 0.0393 pu. line. The predicted time was 0.28 seconds as against an actual time of 0.27 seconds. For all intents and purposes this prediction is exact since during the search procedure an integration interval of 0.01 seconds was used. A similar fault was placed on the system with the 0.0334 pu. line removed. Here the predicted time was 0.27 seconds while the actual time was 0.24 seconds. The prediction is not far off for this fault, but it is on the high side. The clearing time was also evaluated for a similar fault on machine 2 with the 0.0315 pu. line removed. The actual time for this fault was evaluated at 0.29 seconds while the predicted time was 0.28 seconds.

These clearing times were also checked against the estimate provided by the energy-integral criterion [17]. The resulting estimates for the clearing times was 0.3 seconds for the fault used to generate the ellipse and 0.29 seconds for the other fault on machine 1. The estimate for machine 2 was 0.44 seconds.

The purpose of this example was to show that pattern recognition could indeed by used to generate good estimates for critical clearing times for a multimachine example. In the given system, an almost exact estimate for the chosen fault was produced. The approximation took little time to generate. The Energy-Integral Criterion also produced a good estimate. It was easy to apply once the unstable equilibrium was generated. It did, however, take some time to locate this point and verify that there weren't any others closer to the equilibrium. At any rate, the technique did perform well. In the next section, the multiple allocation procedure is investigated via this same example.

3.2.2 Multiple Allocations

For this problem the total amount of generation was fixed at seventy-five per cent of capacity. The generation allocations used are given in Table 3.3 as per unit quantities on a one hundred megavolt-ampere base. In all allocations the amount to be given to machines 1 and 2 was chosen to reflect a given load condition and the remaining machine was allowed to pick up the balance. Allocation 3 was chosen to reflect a near maximum operating conditions on machines 1 and 2; allocations 1 and 2 were chosen to represent unbalanced conditions; allocation 4 reflects a balanced, low-power level.

Three postfault networks were chosen for this example. These consisted of the original system with the 0.0274 pu., 0.0393

Machine Allocation	1	2	3
1	6.0	4.2	9.54
2	3.1	3.1	13.54
3	6.5	6.0	7.54
4	3.5	5.0	11.24
Capacity	6.8	6.01	14.04

Table 3.3 Three-machine Power Allocations

pu., and 0.0115 pu. lines removed. Thus, there are twelve allocation-network pairs and this number of simulations are required to confirm that a point is stable.

Load flow data was used to calculate the voltages behind the transient reactances and Equations 1.6 were used to find the equilibrium points. The stable postfault equilibrium from which the boundary search emanated was $[x_1,\dot{x}_1,x_2,\dot{x}_2] = [4.6^\circ,0.0,1.6^\circ,0.0]$ where the units are in degrees and degrees per second.

The random direction data-gathering prodedure was used to obtain a separating surface with the recursive, stochastic algorithm given previously. The search was terminated after 250 directions representing 1501 tested points. The decision to terminate at this point was made by checking the changes in the values of the coefficients and by monitoring the projections on the x_1-$\dot{x}_1$ and x_2-$\dot{x}_2$ planes. Figure 3.15 shows the evolution of the x_1-$\dot{x}_1$ projection as more directions are used. After 200 directions this cross section is fixed. Figure 3.16 compares the final projection with the full-power results and with the one-machine results of Section 3.1. It is noted that both three-machine projections are larger than the one-machine ellipse due to the higher power allocation in the one-machine case and the fact that machine 3 is not an infinite bus as in the single-machine case. The full-power projection is also expected to be larger since machine 3 has a higher allocation and can move much more than in any of the 75

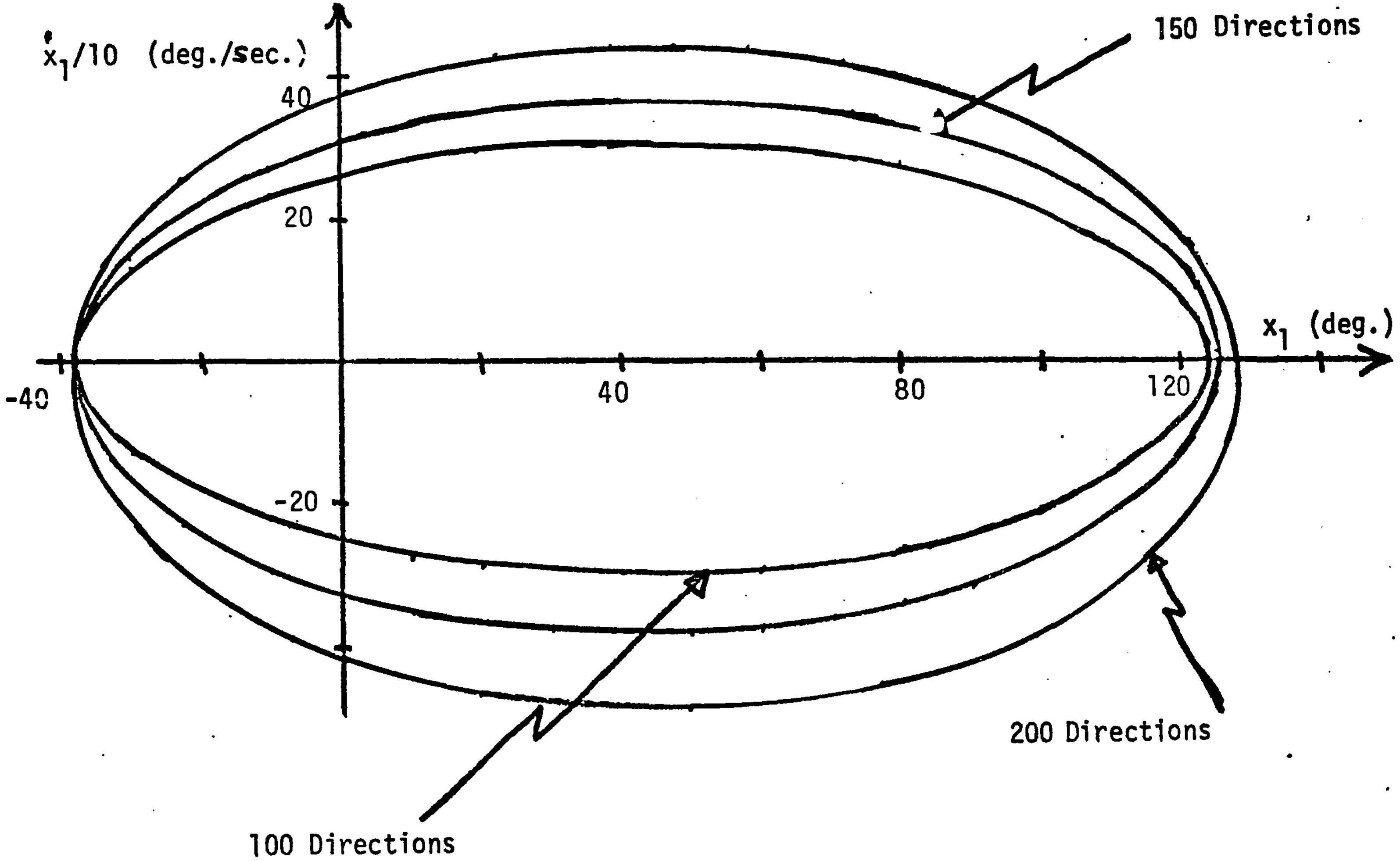


Figure 3.15 Evolution of x_1-$\dot{x}_1$ Projection-
Multiple Allocation Approximation

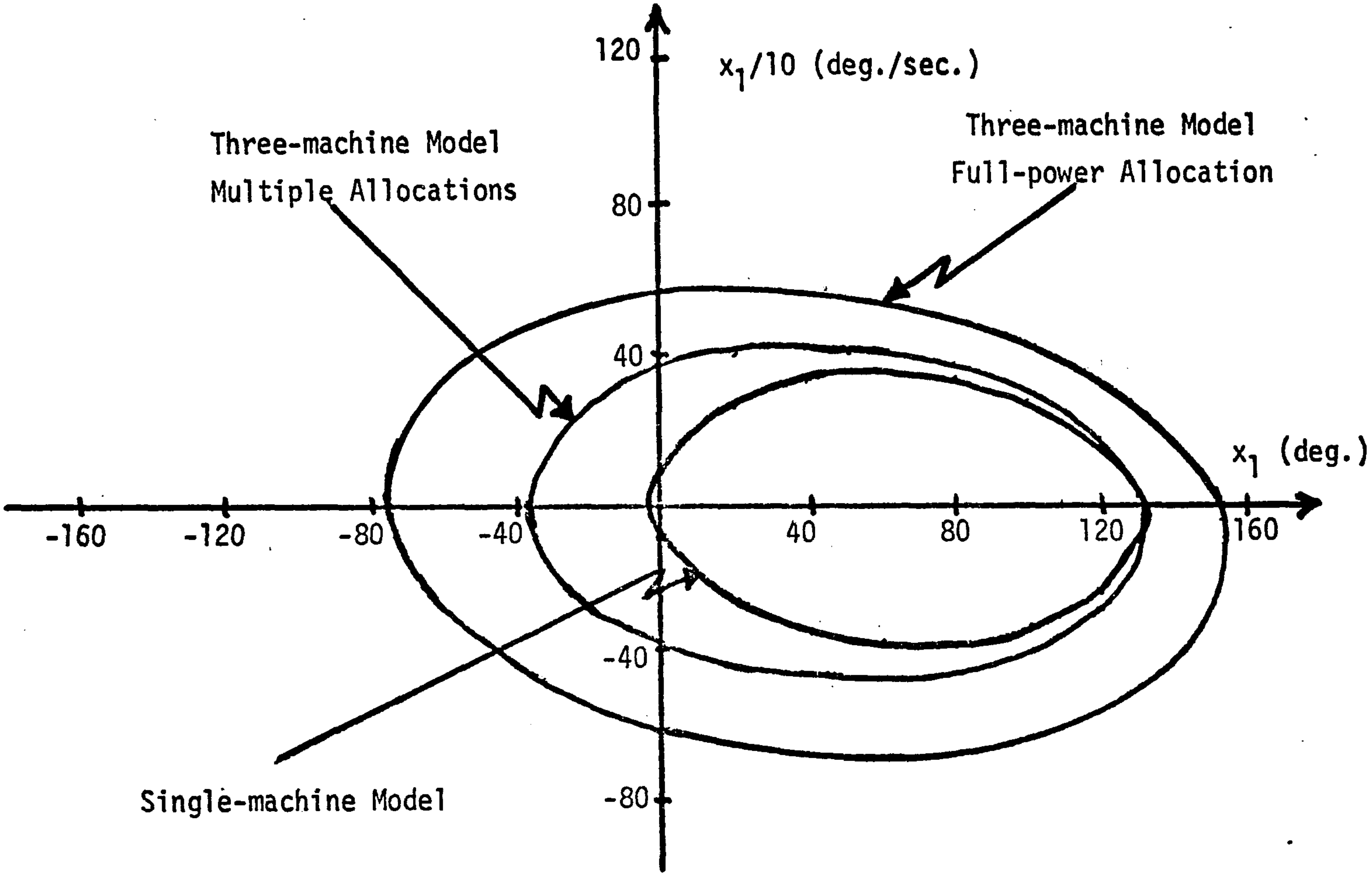


Figure 3.16 x_1-x_1 Projection - Multiple Allocation Boundary Approximation

per cent allocations used in the example being considered. Figure 3.17 depicts the evolution of the x_2-$\dot{x}_2$ projection. Although this ellipse is not as fixed as the previous projection, the orientation and the center have not significantly changed between 150 and 250 directions. Figure 3.18 shows the mean-squared-error and the percentage of correctly classified patterns versus direct number. It is seen that these quantities are fairly constant after 125 directions. Finally, Figure 3.19 shows the values of the coefficients of the separating surface versus iteration number. These values will never become constant due to the inaccuracy of approximation. However, they do become close to their final values at least 75 directions before the search was terminated. After weighing these factors, the decision to test the obtained separation was made. It is noted that the decision is much more difficult here than for the single-fault, full-power case.

Computationally, the gathering of the data required about 90 minutes of time on an IBM 360-65. Approximately 80 per cent of this time was central processor time. This is to be expected since there are a large number of simulations to be performed and this should consume the bulk of the computation time. Some average values that were tabulated are that it took about six tested points per direction to locate the boundary and each direction consumed 23 seconds.

The separation was evaluated using a Monte Carlo volume

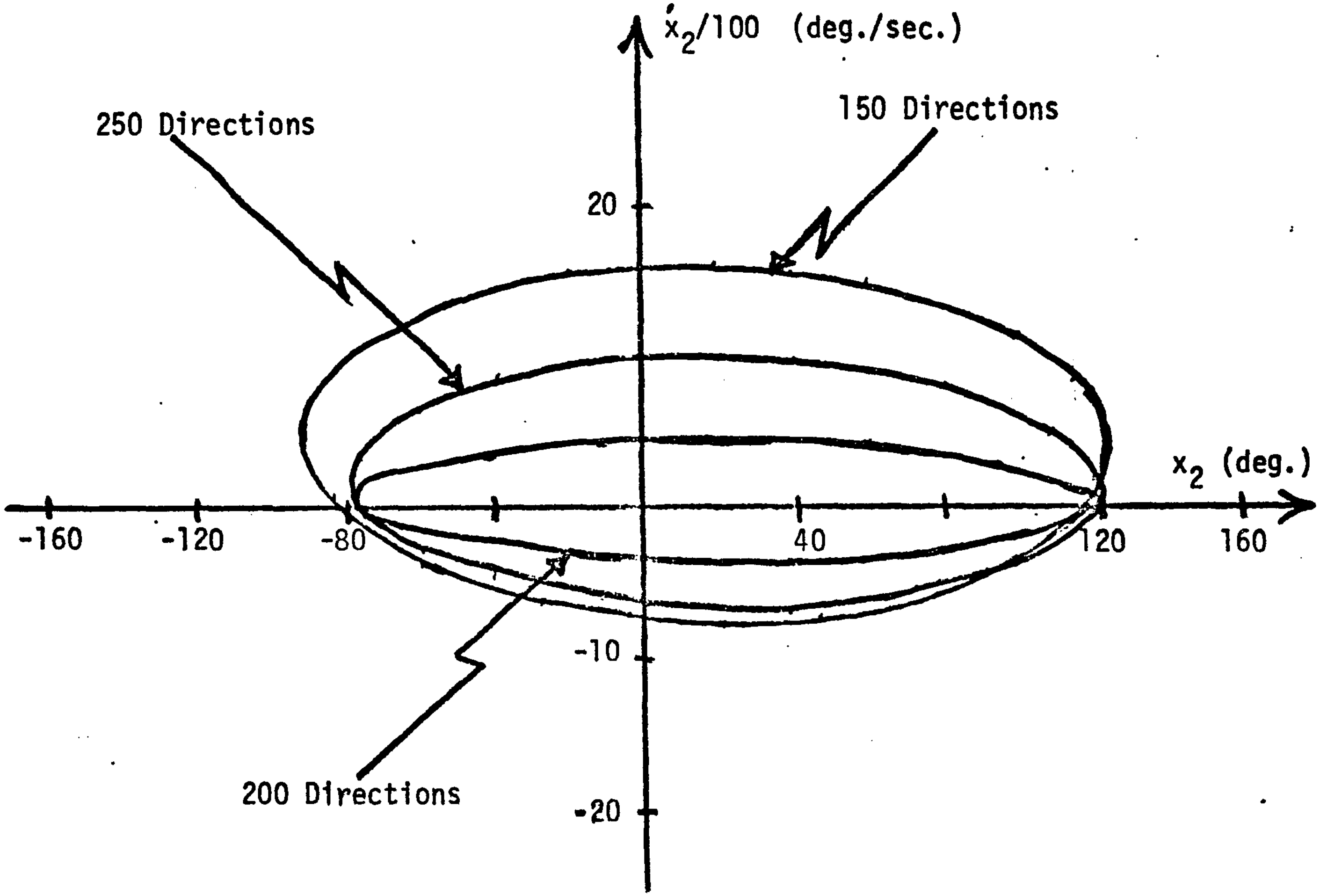


Figure 3.17 Evolution of x_2-$\dot{x}_2$ Projection - Multiple Allocation Approximation

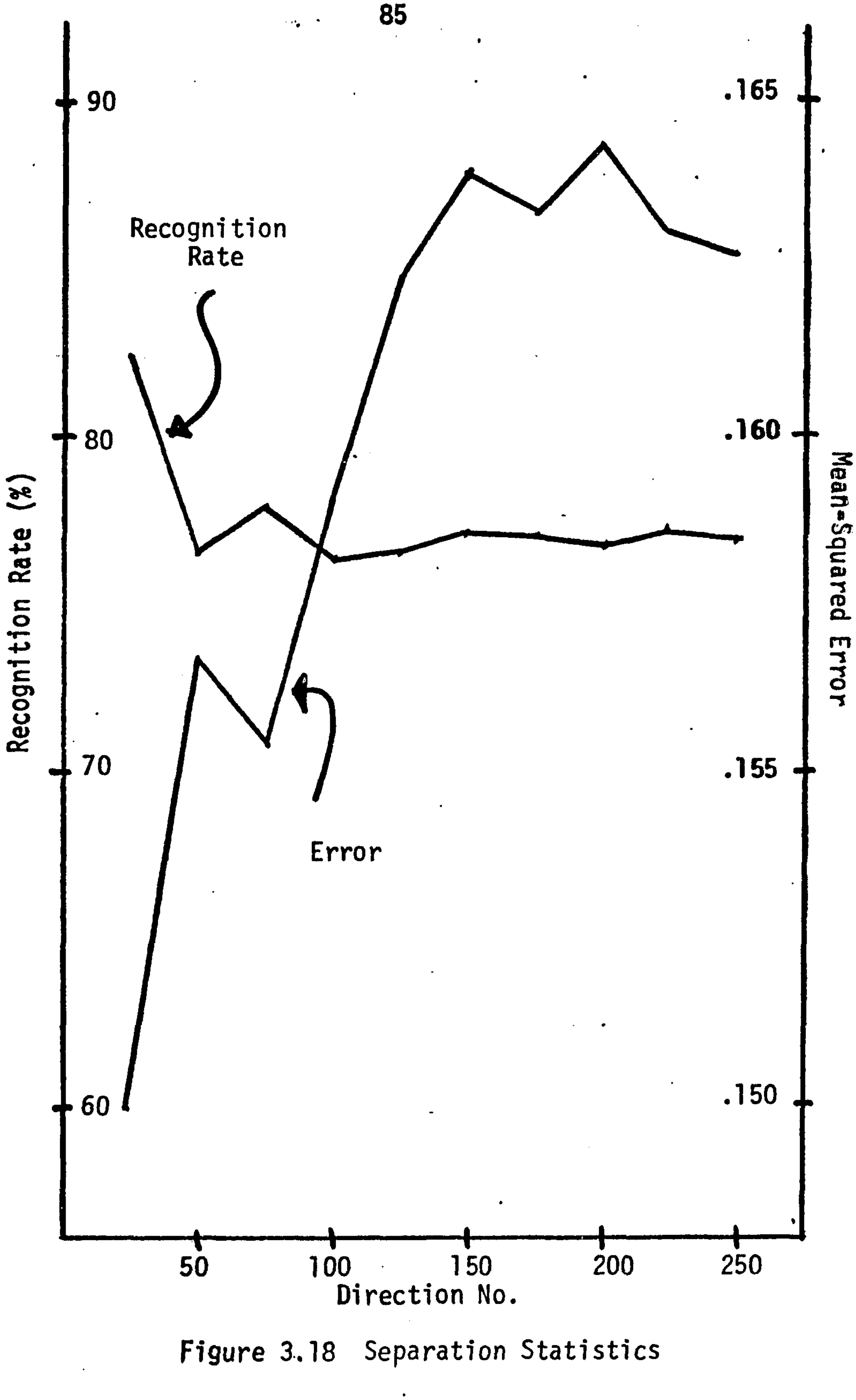


Figure 3.18 Separation Statistics

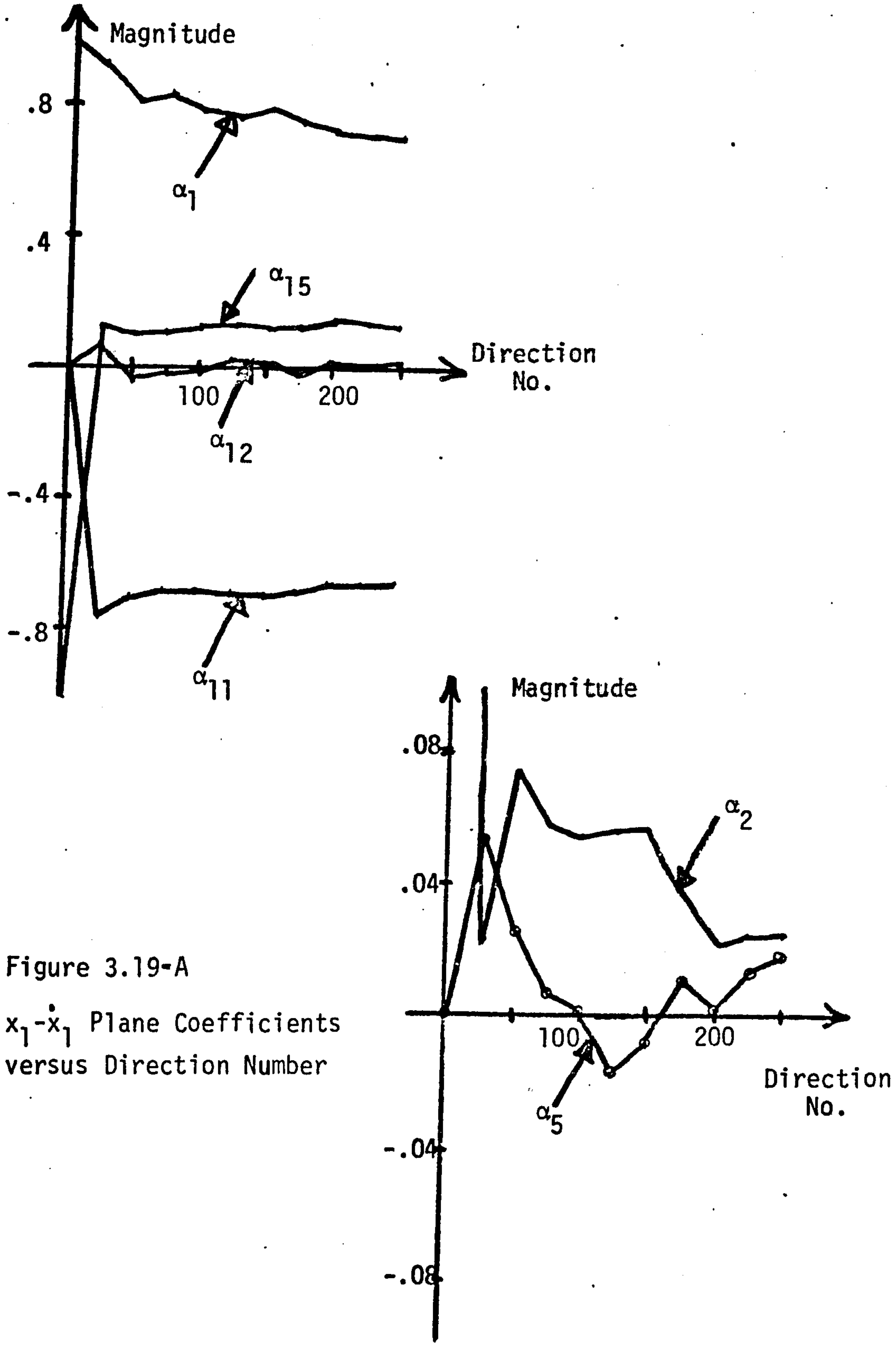


Figure 3.19-A

x_1-$\dot{x}_1$ Plane Coefficients versus Direction Number

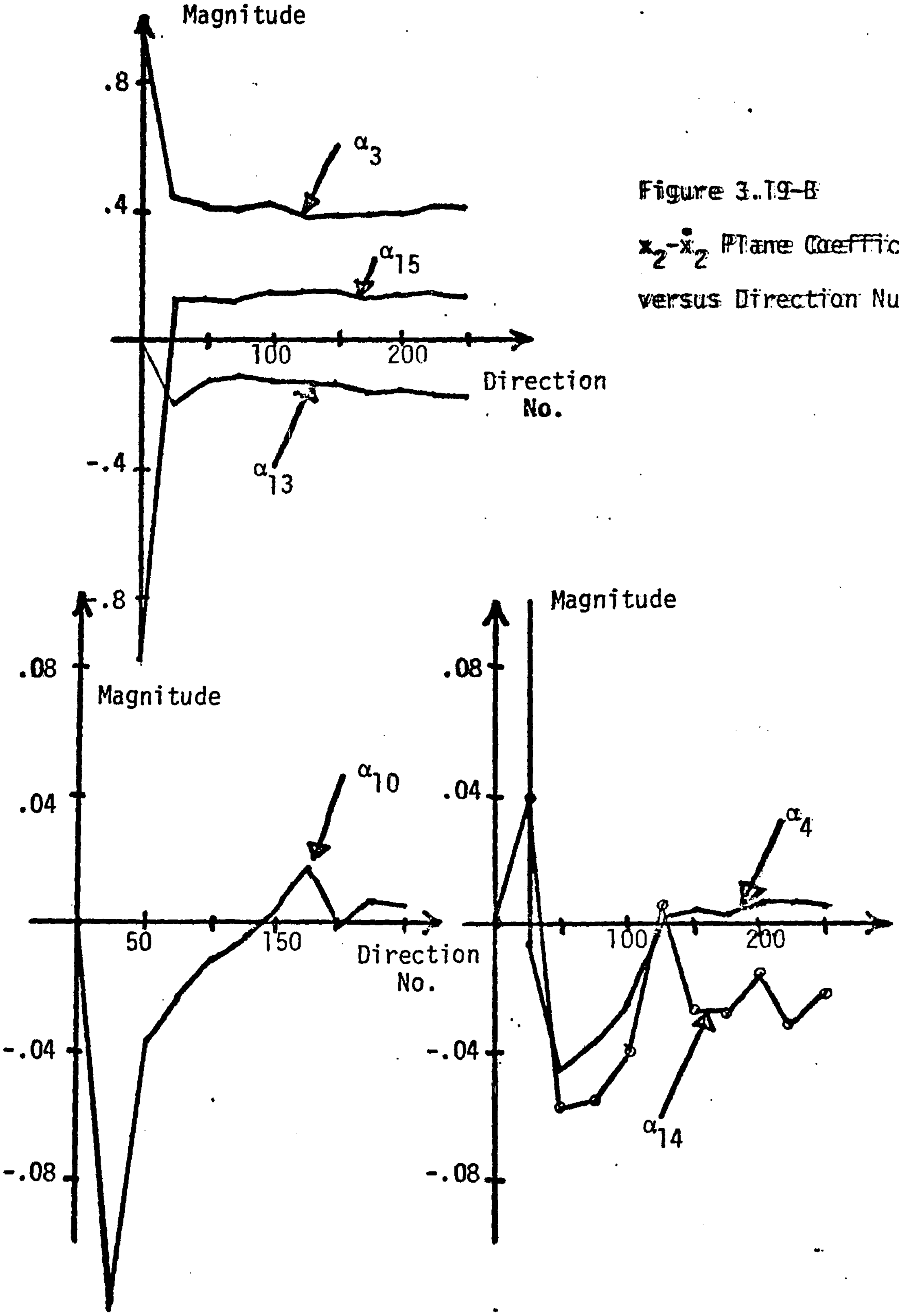


Figure 3.19-B

x_2-$\dot{x}_2$ Plane Coefficients versus Direction Number estimation as described in Section 2.3.2. It was found that the volume of the ellipse is 93.4 per cent of the total stable volume. It was also found that 21.6 per cent of the volume of the ellipse was in error and that 27 per cent of the actual stable volume was not covered. These figures state that the approximation is very nearly the same size as the solid it is trying to describe. For a random point within the ellipse the probability of correctly classifying it was found to be .784; this figure gives some measure of confidence in the classification of a point.

For the algorithm used to obtain the separation, the decision function as given in Chapter II is given by

$$\alpha^T\phi(x) = 0.5 \quad . \quad (3.4)$$

For those patterns that were misclassified in the test, the value of $\alpha^T\phi(x)$ varied from 0.5 with an average magnitude of 0.133. However, the maximum limits were very wide being from 0.1428 to 1.15 for unstable and stable misclassified patterns respectively.

From the fact that only 73 per cent of the stable area is covered by the ellipse one might continue the search procedure further to try and improve this figure. It is possible, however, for this figure to represent a good approximation depending upon the shape of the actual boundary. It is desirable that the missed patterns have values closer to 0.5. For this reason perhaps one should gather more data points without looking further. However, in the last analysis the critical clearing times are the values that it is desired to estimate and it would be interesting to see how well this can be done from the few directions that have been searched thus far.

There are several other reasons for looking at the clearing times now. The ability to stop and evaluate a separation and if it is not very successful, return and gather more data is a very desirable one to have in pattern recognition problems. It is thought that the technique under consideration can be applied to a wide class of such problems and as a rule, the amount of available data in pattern recognition problems is small. Also, for a power system of this size the amount of computer time that has been consumed is fairly substantial. Hence, in practice it may be necessary to stop at this point.

The critical clearing times were evaluated for the twelve combinations of power allocations and postfault networks used to generate the ellipse. The fault was assumed to occur right on the output bus of the machine connected to the faulted lines chosen for the networks. The values predicted and obtained for these times are given in Table 3.4.

It is well to discuss how we expect these times to be estimated by the multiple-allocation procedure. For a given fault the higher the power allocation, the larger the excursion from the prefault equilibrium for a given time. For this fault

Allocation No.	Fault		Clearing Time (sec.)	
	Machine No.	Line Out (pu.)	Predicted	Actual
1	1	0.0334	0.18	0.21
	1	0.0393	0.19	0.22
	2	0.0315	0.41	0.38
2	1	0.0334	0.35	0.41
	1	0.0393	0.38	0.43
	2	0.0315	0.31	0.30
3	1	0.0334	0.16	0.17
	1	0.0393	0.17	0.19
	2	0.0315	0.30	0.23
4	1	0.0334	0.38	0.41
	1	0.0393	0.42	0.52
	2	0.0315	0.46	0.54

Table 3.4 Three-machine Clearing Times

the major factor regarding stability is the increase in energy of the machines during the time it is on the system. Now, the larger the power allocation, the smaller the excursion from the equilibrium before this increase is realized. Hence, since each point is tested for all allocations during the data gathering, it is hoped that accurate estimates of clearing times are obtained for a high power allocation and then we expect to get conservative estimates for low power. All of this is for a given fault. We also expect the clearing times to decrease as the allocations to machines 1 and 2 increase. Note that in the chosen example machine 3 has such a large inertia that its allocation is insignificant.

The times given in Table 3.4 reflect these expectations almost completely. For allocation 1 and faults on machine 1 which has a high power input the clearing times are small. The time for a fault on machine 2 which has a low power allocation is about twice that for machine 1 faults. Note that all times are predicted accurately. For allocation 2 machine 1 has a relatively low power input and as expected, estimates of the clearing times for faults on this machine are conservative. Machine 2 has a higher power input and hence a closer prediction is anticipated. From the table it is seen that conservative estimates are obtained in the first case and in the second the prediction is for all intents and purposes exact. In allocation 3 both machines 1 and 2 have high power inputs, and hence one expects times near those for machine 1 in allocation 1. This is indeed the case and for faults on machine 1 accurate predictions are made. However, for the fault on machine 2 the prediction is 30 per cent greater than the actual time. This is the only case in which an unexpected prediction is made. Allocation 4 should produce conservative estimates and the experimental values verify this.

This author considers the estimates excellent. With only 250 directions, the times are very close for the high power inputs in all but one case; the low power estimates are conservative in all but one case and in that one, a good estimate is made. Thus, only one estimate is bad. If one uses the average missed value obtained with the Monte Carlo experiment, this same estimate is the only one exceeding the error limit given by

$$|\alpha^T\phi(x)-0.5| \leq 0.133 \qquad . \quad (3.5)$$

From these results one might desire to reevaluate the multiple allocation formulation. The clearing time estimates for the low-power case may be too conservative. It has been shown,however,that good estimates can be made for a single allocation and several faults.

3.3 Summary

In this chapter it has been shown that the pattern recognition technique presented in Chapter II is indeed applicable to the transient stability problem. A single-machine example was used to test the ability of the quadratic decision surface to approximate the actual boundary. A simplified grid approach was used to gather data and clearing times were estimated. The estimates of these times were conservative and reasonably accurate. The same system was then solved using a proposed automated data-gathering procedure. The obtained approximation was similar to the one found previously. The two algorithms described in Chapter II were used in this second case and the resulting surfaces were almost the same.

A three-machine system was also described for a near capacity generating condition and for a single fault. The estimate of the clearing time was almost exact. The boundary of this system was also described using four allocations and three faults. The multiple-allocation procedure generated an ellipse which seemed not to be greatly accurate. However, the estimates of clearing times were close in almost all the cases tested.

CHAPTER IV

CONCLUSIONS AND EXTENTIONS

4.1 Summary

The technique that has been presented here is basically an application of pattern recognition theory and technique to the power system transient stability problem. It differs from the conventional approach in that it has an organized data-gathering procedure. This procedure tries to convey more information to the recognition algorithm than would an arbitrary data set and thereby improve its separating ability. It is thought that its feasibility for application to the transient stability problem has been shown.

In Chapter I a discussion of the transient stability problem was presented. The model to be used in the thesis was presented and appears as Equations 1.5. The type of faults to be considered were restricted to symmetrical three-phase short circuits and postfault networks were assumed steady-state stable.

In Chapter II the basic method of obtaining the approximation to the stability boundary was presented. We began by reviewing the Liapunov approach to the problem and pointed out its advantages over previous methods and its inherent difficulties. From the shape of the single-machine boundary, Figure 2.1, it was postulated that perhaps an ellipse would approximate it very well. The problem could then be formulated as a pattern recognition problem with a quadratic decision surface. To gather the data it was suggested that linear searches along a set of rays using systems simulations to locate the boundary would provide a meaningful data set. This technique for generating the boundary would have several advantages over the Liapunov method and would serve to provide the same information about the system. In fact, by using several allocations and several postfault networks it was postulated that the system could be represented for a fixed amount of total generation and thereby provide information that previous methods could not. The pattern recognition problem was then mathematically formulated and several algorithms were given. Finally, a method for evaluating the separation was given.

In Chapter III several examples were worked using the given technique. A one-machine system was solved in two different ways using two data-gathering methods. In both cases the approximation was reasonably accurate and estimates of the critical clearing times were close. Both algorithms gave similar results and it was judged that there was no significant difference between the approximations achieved from them. It did emerge, however, that bulges could be produced in the surface due to the limit on the simulation time, but since this could not be detected, the effect must be tolerated for the sake of reasonable computation time. A three-machine problem was also worked using a single allocation and fault and using the multiple allocation procedure. For the first case the separation was performed rapidly and the estimate for the clearing time was almost exact. In the second problem a heuristic decision was made to terminate data gathering and test the separation. The critical clearing times were determined and were as expected in eleven of the twelve evaluated. In the allocations with a high power input to one of the low inertia machines the predictions were very accurate. The ability to respond to several allocations might be questioned if more accurate predictions were desired for all faults and allocations.

4.2 Conclusions and Extensions

From the results of our examples, it has been shown that accurate clearing time estimates can be made for a single power allocation and several faults. It is therefore envisioned that several boundaries such as we have generated here will be necessary to represent a power system. Several typical system configurations that represent the load conditions for a single day would each be used to generate a boundary. Several such groups of boundaries would have to be used to represent seasonal variations.

As mentioned in the introduction to this thesis, the boundary can be used to provide stability information to an operator. In this case it would be necessary only to simulate the system up to the switching time to determine if a disturbance will cause instability. Knowledge of the system state at switching can be used with the appropriate boundary to provide this information.

The second use mentioned pertained to a real-time computer control situation. If a fault is detected on the system, the boundary can be used to determine if corrective action is necessary. The state at switching can be determined by a simplified fast-time simulation or by direct measurement. Once it is known that action is necessary, decisions as to the type of action to be taken can be obtained from a contingency plan. Some types of available actions are load shedding, placing of capacitance in series with a machine, switching in braking resistors on appropriate machines, reallocation of generation, and combinations of these.

The technique that has been presented here is an application of pattern recognition to the power system transient stability problem. It is by no means restricted to power systems nor to stability problems. It should prove helpful wherever a boundary is to be estimated. One application that immediately comes to mind is that of estimating the attainable set of a system for a fixed time.

In the applications that have concerned us here, system simulation was used to determine the class membership of training patterns. This was natural in the stability problem under consideration. However, in other pattern recognition problems simulation may not be applicable. The technique can still be used, however, if the patterns can be placed into classes by another method. If, for example, it is desired to separate the letters of the alphabet in some appropriate space, perhaps the determination of class could be made by a human being looking at the actual letters.

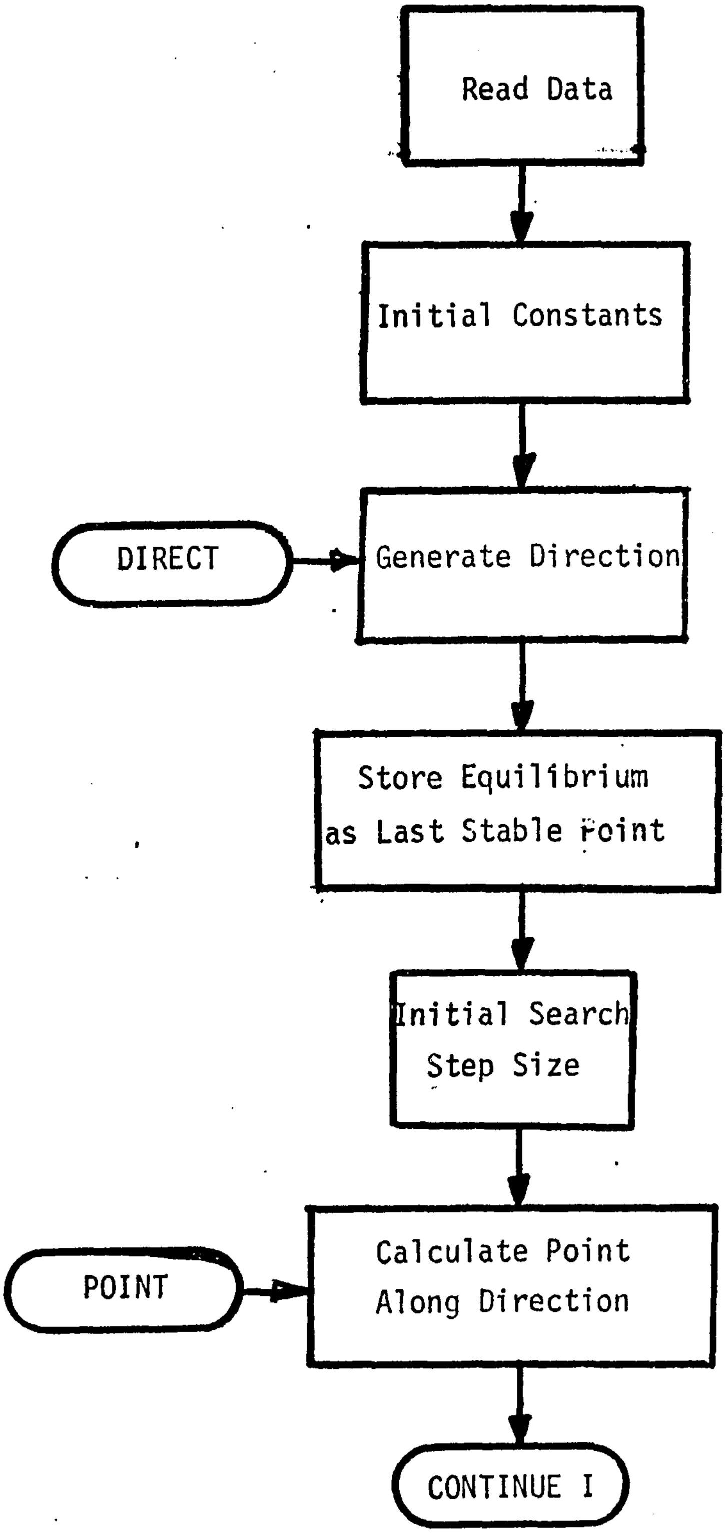
If there are more than two pattern sets, the technique can still be applied. All that is required is that at least one member of each class be known. The method would then be used to search along rays from each known member in turn. This should serve to concentrate a lot of patterns along the boundaries. The search procedure would have to be modified to account for the possibility of one or more sets being open, that is, having a portion of the boundary that is unbounded. Usually, one can set bounds on the search, large though they may be.

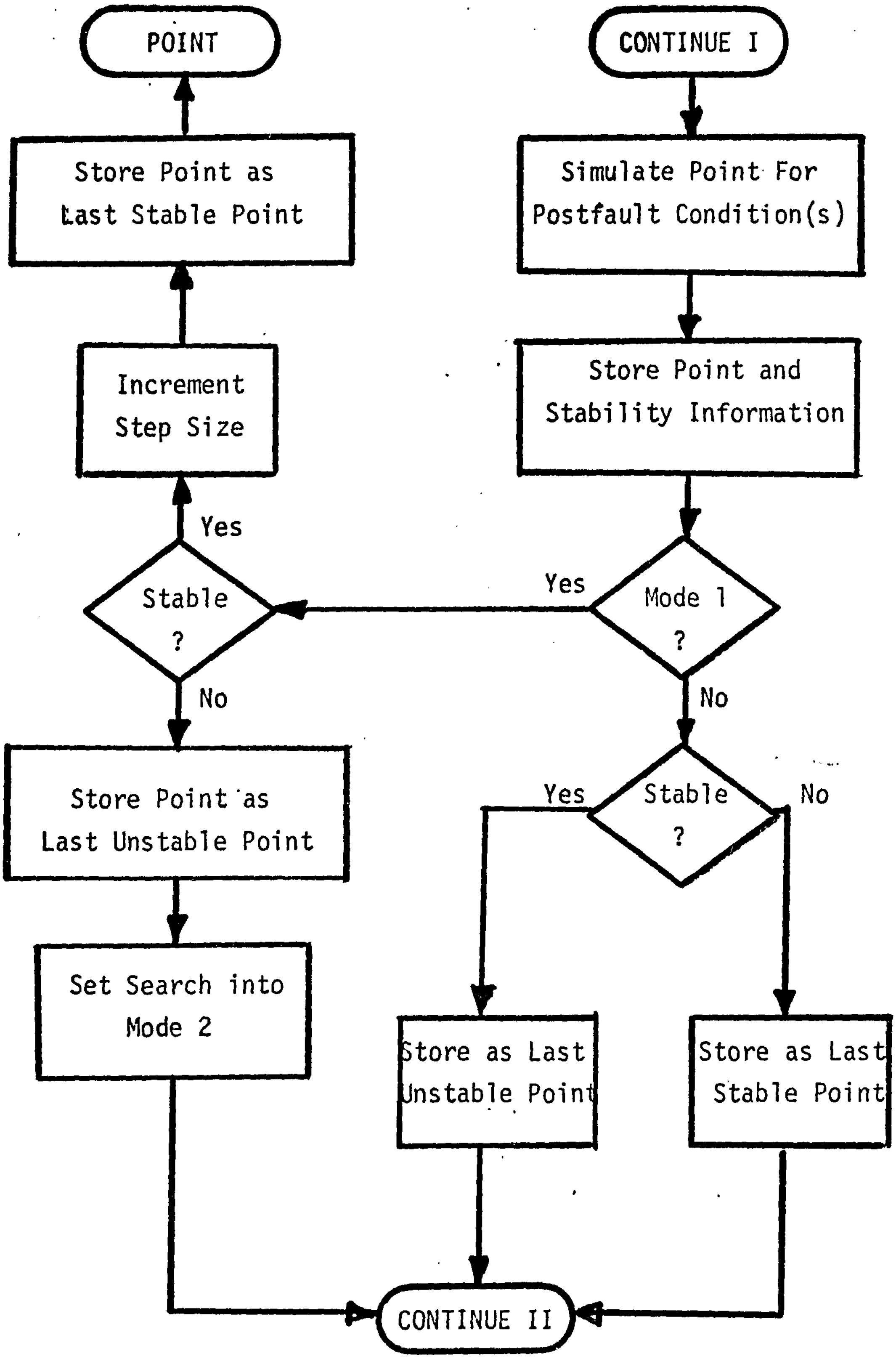
There are some immediate extensions of this work in relation to the transient stability problem. The natural one is to attempt the solution of a larger system. It is also natural to try a more complicated model.

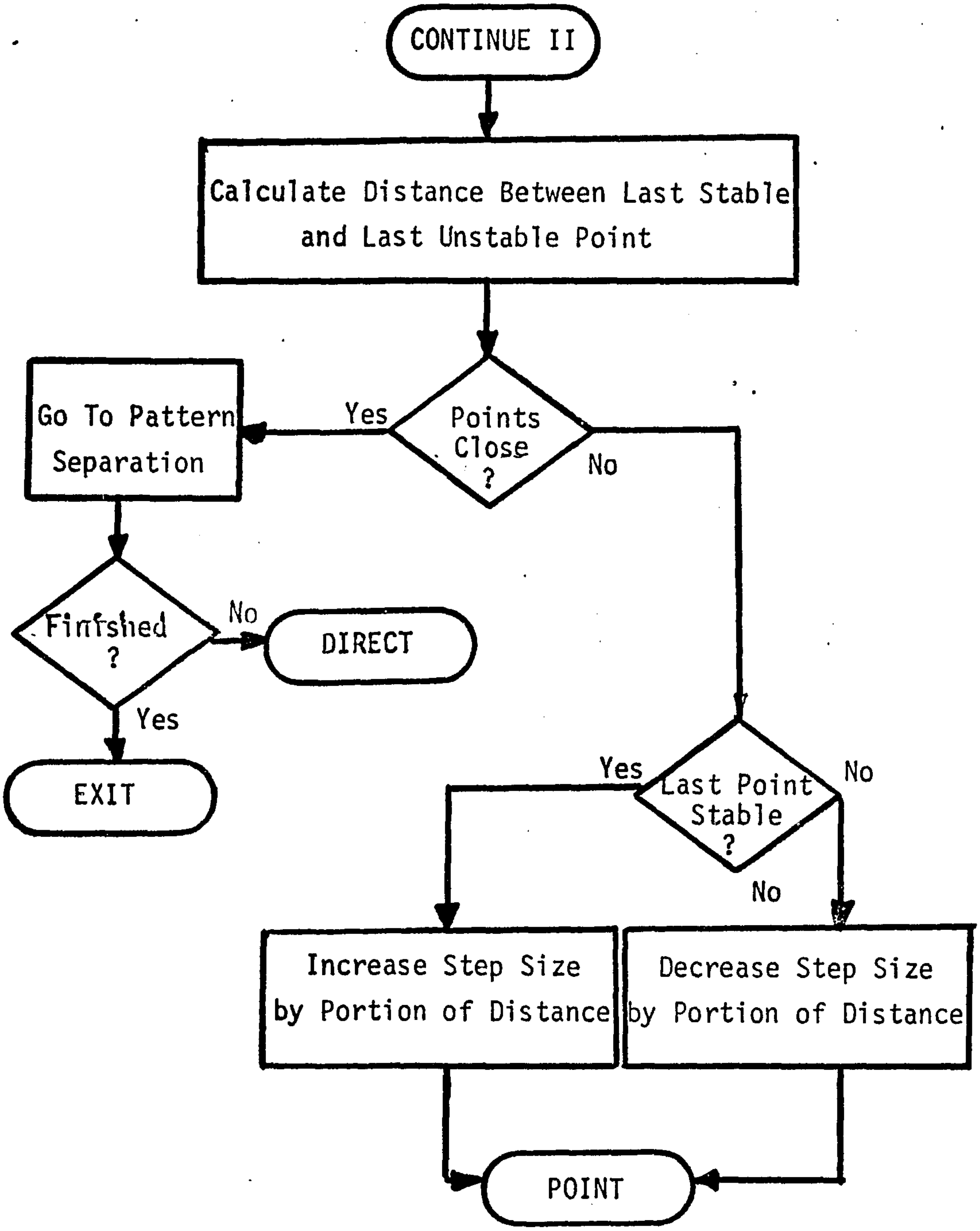
I would like to suggest that the ideas presented in this thesis might find application in many other systems where security assessment is a problem. I would also like to recommend the data-gathering technique as a way of providing a meaningful data set for pattern recognition problems. It is also hoped that further work will be done in this area so as to bring some of these applications to light.

APPENDIX I PROGRAM FLOW CHARTS

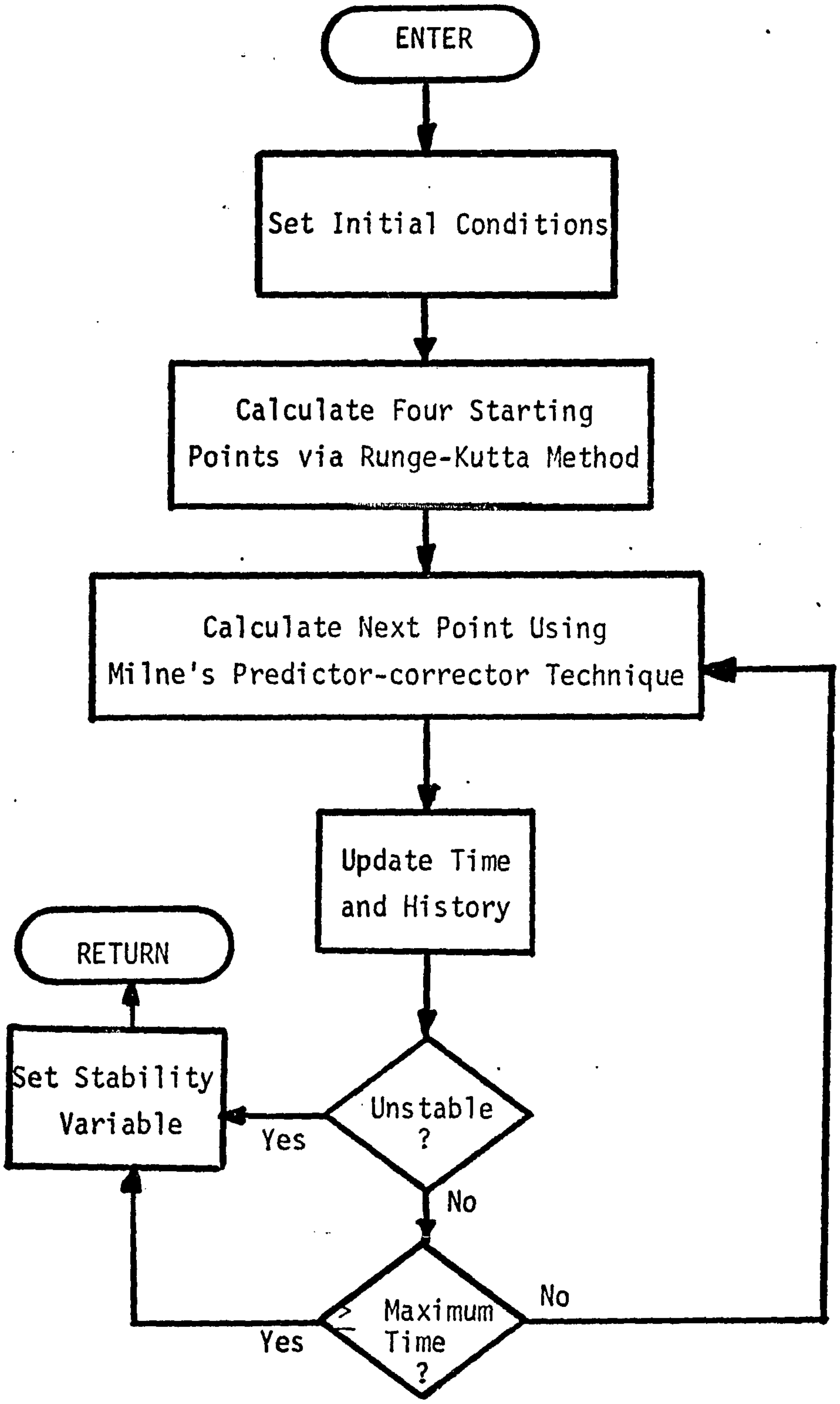
A. Main Program




POINT
CONTINUE I
Store Point as
Last Stable Point
Simulate Point For
Postfault Condition(s)
Increment
Step Size
Store Point and
Stability Information
Yes
Stable
?
Yes
Mode 1
?
No
No
Store Point as
Last Unstable Point
Yes
Stable
?
No
Set Search into
Mode 2
Store as Last
Unstable Point
Store as Last
Stable Point
CONTINUE II


CONTINUE II
Calculate Distance Between Last Stable and Last Unstable Point
Points Close ?
Yes
No
Go To Pattern Separation
Finished ?
No
DIRECT
Yes
EXIT
Last Point Stable ?
Yes
No
No
Increase Step Size by Portion of Distance
Decrease Step Size by Portion of Distance
POINT

B. Simulation Program


ENTER
Set Initial Conditions
Calculate Four Starting Points via Runge-Kutta Method
Calculate Next Point Using Milne's Predictor-corrector Technique
Update Time and History
Unstable ?
Yes
No
Set Stability Variable
RETURN
Maximum Time ?
Yes
No

Simulation Equations:

a.)Runge-Kutta Equations for $y'' = f(y)$

$\Delta t \equiv$ integration interval (sec.)

$$C_0 = \Delta t \cdot f(y_i)$$

$$C_1 = \Delta t \cdot f(y_i + \Delta t/2 \cdot y'_i)$$

$$C_2 = \Delta t \cdot f(y_i + \Delta t/2 \cdot y'_i + \Delta t/4 \cdot C_0)$$

$$C_3 = \Delta t \cdot f(y_i + \Delta t \cdot y'_i + \Delta t/2 \cdot C_1)$$

$$y'_{i+1} = y'_i + 1/6\ (C_0 + 2C_1 + 2C_2 + C_3)$$

$$y_{i+1} = y_i + \Delta t \cdot y'_i + \Delta t/6 \cdot (C_0 + C_1 + C_2)$$

b.)Milne's Predictor-corrector Equations for $y'' = f(y)$

$\Delta t \equiv$ integration interval (sec.)

Predictor

$$Py'_{i+1} = y'_{i-3} + 4\Delta t/3(2y''_i + 2y''_{i-2} - y''_{i-1})$$

$$Py_{i+1} = y_{i-1} + \Delta t/3(y'_{i-1} + 4y'_i + Py'_{i+1})$$

$$Py''_{i+1} = f(Py_{i+1})$$

Corrector

$$Cy'_{i+1} = y'_{i-1} + \Delta t/3(y''_{i-1} + 4y''_i + Py''_{i+1})$$

Iterating Corrector

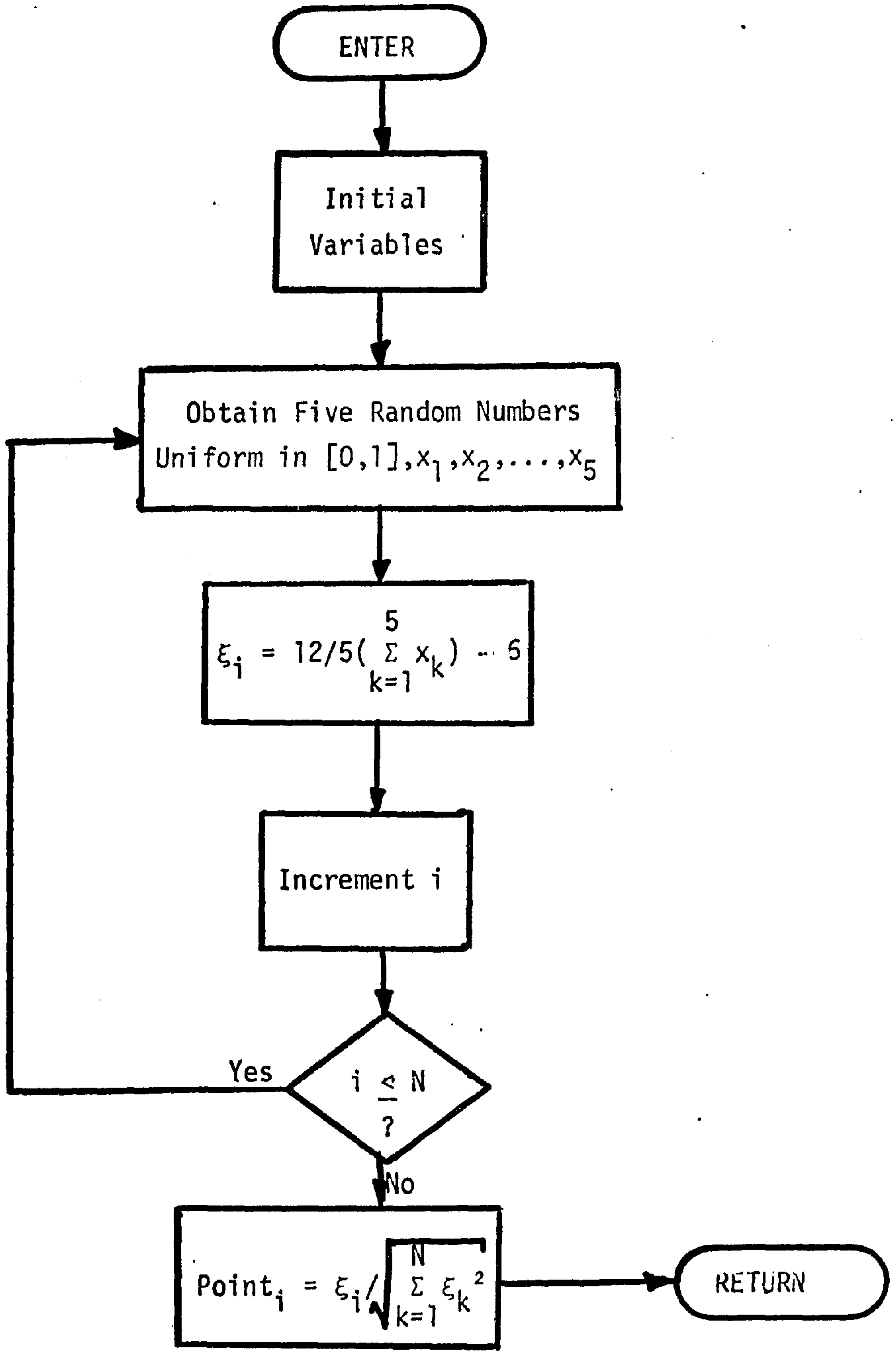
$$C'y_{i+1} = y_{i-1} + \Delta t/3(y'_{i-1} + 4y'_i + Cy'_{i+1})$$

$$C'y''_{i+1} = f(C'y_{i+1})$$

$$C'y'_{i+1} = y'_{i-1} + \Delta t/3(y''_{i-1}+4y''_{i}+C'y''_{i+1})$$

If ($|C'y' - Cy'| \leq 0.005$) exit; otherwise, repeat last three equations with $Cy' = C'y'$.

C. Generation of a Uniform Random Direction by Picking a point uniformly on an N-dimensional Sphere


ENTER
Initial Variables
Obtain Five Random Numbers Uniform in [0,1],$x_1,x_2,\ldots,x_5$
$\xi_i = 12/5(\sum_{k=1}^{5} x_k) - 6$
Increment i
$i \leq N$?
Yes
No
$\text{Point}_i = \xi_i / \sqrt{\sum_{k=1}^{N} \xi_k^2}$
RETURN

REFERENCES

[1] Otto J. M. Smith, "Power System State Estimation", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-89, No. 3, pp. 363-375, March 1970.

[2] R. E. Larson, W. F. Timmey, and J. Peschon, "State Estimation in Power Systems, Pt. I: Theory and Feasibility", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-89, No. 3, pp. 345-352, March 1970.

[3] H. Nicholoson, "Hierarchical Control of a Multimachine Power System Model", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-87, No. 7, pp. 1537-1548, July 1968.

[4] T. E. Dy-Liacco, "Control of Power Systems Via the Multi-Level Concept", Ph.D. Thesis, Case Western Reserve University, Cleveland,Ohio, June 1968.

[5] F. Schweppe, J. Wildes, and D. Rom, "Power System Static State Estimation", Power Systems Engineering Group of Mass. Insitute of Technology Report No. 10, Nov. 1968.

[6] E. W. Kimbark, Power System Stability, Vol. I, John Wiley & Sons, N. Y. ,1948.

[7] S. B. Crary, Power System Stability, Vol. II: Transient Stability, John Wiley & Sons, N. Y. ,1947.

[8] S. B. Crary, Power System Stability, Vol. I: Steady State Stability, John Wiley & Sons, N. Y. ,1945.

[9] E. W. Kimbark, Power System Stability, Vol. III, John Wiley & Sons, N. Y. ,1956.

[10] C. Concordia, Synchronous Machines, John Wiley & Sons, N. Y., 1951.

[11] B. Weedy, *Electric Power Systems*, John Wiley & Sons, N. Y., 1967.

[12] V. Venikov, *Transient Phenomena in Electrical Power Systems*, The Macmillan Company, N. Y. ,1964.

[13] M. Dyrkacz, C. Young, and F. Maginniss,"A Digital Transient Stability Program Including the Effects of Regulator, Exciter, and Governor Response", *AIEE Trans. (Power Apparatus and Systems)*, Vol. 79, pp. 1245-1257, Feb. 1961.

[14] R. Byerly, D. Ramey, and J. Skoogland, "Stability Program Data Preparation Manual", Westinghouse Electric Utility Headquarters Department Engineering Report No. 68-661, January 1968.

[15] A. Colombo, F. Redaelli, G. Ruckstuhl, and A. Vian, "Determination of the Dynamic Response of Electrical Systems by Means of a Digital Program", *IEEE Trans. on Power Apparatus and Systems*, Vol. PAS-87, No. 6, June 1968.

[16] P. Magnusson, "The Transient-Energy Method of Calculating Stability", *Trans. of AIEE*, Vol. 66, pp. 747-755, 1947.

[17] P. Aylett, "The Energy-Integral Criterion of Transient Stability Limits of Power Systems", *Proc. of IEE*, Vol. 77, Pt. A, pp. 527-536, 1958.

[18] N. Dharma Rao, "A New Approach to the Transient Stability Problem", *Trans. AIEE*, Vol. 60, pp. 186-192, June 1962.

[19] N. Dharma Rao and H. Ramachandra Rao, "Phase-Plane Techniques for the Solution of Transient-Stability Problems", *Proc. IEE*, Vol. 110, No. 8, August 1963.

[20] G. Gless, "Direct Method of Liapunov Applied to Transient Power System Stability", *IEEE Trans. on Power Apparatus and Systems*, Vol. PAS-85, No. 2, Feb. 1966.

[21] M. Siddiqee, "Direct Method of Liapunov and Transient Stability Analysis", Ph. D. Thesis, University of Minnesota, 1967.

[22] A. El-Abiad and K. Nagappan, "Transient Stability Regions of Multimachine Power Systems", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-85, No. 2, Feb. 1966.

[23] Y. Yu and K. Vongsuriya, "Nonlinear Power System Stability Study by Liapunov Function and Zubov's Method", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-86, No. 12, Dec. 1967.

[24] M. Siddiqee, "Transient Stability of an A.C. Generator by Liapunov's Direct Method", International Journal on Control, Vol. 8, No. 2, pp. 131-44, 1968.

[25] J. Willems, "Improved Lyapunov Function for Transient Power-System Stability", Proc. IEE, Vol. 115, No. 9, Sept. 1968.

[26] N. Dharma Rao, "Generation of Liapunov Functions for the Transient Stability Problem", Trans. of the Engineering Institute of Canada, Vol. 11, No. C-3, Oct. 1968.

[27] N. Dharma Rao, "Routh-Hurwitz Conditions and Liapunov Methods for the Transient Stability Problem", Proc. IEE, Vol. 116, No. 4, April 1969.

[28] F. Fallside and M. Patel, "On the Application of the Liapunov Method to Synchronous Machine Stability", Int. Journal on Control, Vol. 4, No. 6, pp. 501-513, 1966

[29] M. Pai, M. Mohan, and J. Gopala Rao, "Power System Transient Stability Regions Using Popov's Method", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-89, No. 5/6, May/June 1970.

[30] J. Willems and J. Willems, "The Application of Liapunov Methods to the Computation of transient Stability Regions for Multimachine Power Systems", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-89, No. 5/6, May/June 1970.

[31] A. MacFarlane, M. Laughton, and G. Shannon, "Generalized Non-Linear Network Analysis Applied to the Dynamic Behavior of Synchronous Machine Systems", Int. Journal on Control, Vol. 3, No. 2, pp. 97-111, 1966.

[32] F. Fallside and J. Waddington, "Exact and Volterra Analyses of Synchronous Machine Stability", Int. Journal on Control, Vol. 11, No. 2, pp. 199-211, 1970.

[33] J. Lasalle and S. Lefschetz, Stability by Liapunov's Direct Method, Academic Press, N. Y. ,1961.

[34] R. Teichgraeber, F. Harris, and G. Johnson, "New Stability Measure for Multimachine Power Systems", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-89, No. 2, Feb. 1970.

[35] J. Willems, "Comments on 'Transient Stability of an A.C. Generator by Liapunov's Direct Method'", Int. Journal on Control, Vol. 10, No. 1, pp. 113-116, 1969.

[36] S. Weissenburger, "Stability-Boundary Approximations for Relay-Control Systems Via a Steepest Ascent Construction of Lyapunov Functions", Trans. ASME, Series D, Vol. 88, No. 2, June 1966.

[37] N. J. Nilsson, Learning Machines, Mcgraw-Hill, N. Y. ,1965.

[38] C. Blaydon, "Recursive Algorithms for Pattern Classification", Technical Report No. 520, Division of Engineering and Applied Physics, Harvard University, March 1967.

[39] G. Sebestyen, Decision-Making Processes in Pattern Recognition, The Macmillan Company, N. Y. ,1962.

[40] Y. Ho and R. Kashyap, "An Algorithm for Linear Inequalities and Its Applications", IEEE Trans., Vol. EC-14, No. 5, Oct. 1965.

[41] D. Wilde, Optimum Seeking Methods, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1964.

[42] H. Robbins and S. Monro, "A Stochastic Approximation Method", Annals of Mathematical Statistics, Vol. 23, pp. 462-66, 1952.

[43] A. Dvoretzky, "On Stochastic Approximation", Proc. of the Third Berkeley Symposium on Mathematical Statistics and Probability, Univ. of California Press, Los Angeles, Vol. 1, pp. 39-55, 1956.

[44] B. Harris, Theory of Probability, Addison-Wesley, Reading, Mass., 1966.

[45] Y. Shreider, Ed., The Monte Carlo Method, Pergamon Press, N. Y., 1966.

[46] M. Muller, "A Note on a Method for Generating Points Uniformly on N-dimensional Spheres", Communications of the Assoc. of Computing Machinery, Vol. 2, No. 4, pp. 19-20, 1959.