NEXT-GENERATION, PHASOR-BASED DISTANCE RELAY
WITH FAULT-PATH RESISTANCE IMMUNITY

Master's Thesis

Submitted To The Faculty Of Virginia Tech In
Partial Fulfillment Of The Requirements For The Degree Of

MASTER OF SCIENCE
ELECTRICAL ENGINEERING

by

David Matthew Smith

APPROVED:

[Signatures]

Arun G. Phadke

Yilu Liu

Jaime De La Ree Lopez

VIRGINIA TECH
Bradley Department of Electrical Engineering

February, 1996
Blacksburg, Virginia
NEXT-GENERATION, PHASOR-BASED DISTANCE RELAY WITH FAULT-PATH RESISTANCE IMMUNITY

by

David M. Smith

Arun G. Phadke, Chairman
Bradley Department of Electrical Engineering

Abstract

Distance relays are commonly utilized for the protection of transmission lines. A distance relay calculates the apparent impedance between the relay location and the fault location, and uses this value to determine the approximate location of the fault. Traditional distance relays, however, are plagued with problems caused by faults through a resistance. The fault-path resistance causes the apparent impedance determined by the distance relay to appear larger than the actual impedance between the relay location and the transmission line fault location. This error in the apparent impedance may deceive the relay and cause it to misoperate. Errors due to fault-path resistance are a fundamental problem of distance relaying.

The solution to the problems caused by fault-path resistance is presented in the form of the advanced distance relay. The advanced distance relay is a computer-based relay which employs a fault locator algorithm that is immune to the effects of resistance in the fault-path. The major substance of this document is dedicated to explaining the algorithms utilized to create the advanced distance relay. A major advantage of the advanced distance relay is that it is a single-ended relay. In other words, it requires voltage and current information from only one end of the transmission line that it protects. Another advantage is that knowledge of the system equivalent source impedance is not required. While knowledge of the distribution factor phase angle, parallel line zero-sequence current, and parallel line zero-sequence impedance are not required, in the event that they are available, the relay can provide even more refined, accurate results.

The advanced distance relay prototype was written in the "C" programming language. It utilizes a sampling frequency of 1440 Hz, and includes an algorithm that eliminates DC offset in the current waveform. The relay also employs a 1/2 cycle DFT phasor calculation algorithm, which provides the data required by the per-unit fault locator equation. The implication of this information is that the relay is capable of making a "trip / block" decision in just over 1/2 of a cycle.

As mentioned, the most powerful feature of the advanced distance relay is its immunity to the effects of fault-path resistance. It is this immunity that allows the relay to overcome the aforementioned fundamental problem of distance relaying. Consequently, the advanced distance relay is a powerful addition to the tools of protective relaying.
Dedicated to Christine, Allen, Betty, Arnold, Gertrude, and Dee Dee.
ACKNOWLEDGMENTS

I would like to thank Dr. Arun Phadke, my advisor, for generously contributing his time to help bring this project to a successful conclusion. I also wish to thank Dr. Yulu Liu and Dr. Jaime De La Ree Lopez for serving on my graduate committee.
# TABLE OF CONTENTS

**CHAPTER 1:** Introduction.................................................................................. 1  
  1.1 Prelude ................................................................................................. 1  
  1.2 The Problem ........................................................................................ 2  
  1.3 The Proposed Solution ........................................................................ 2  
  1.4 Overview .............................................................................................. 3  

**CHAPTER 2:** Traditional Distance Relaying ................................................. 4  
  2.1 Fundamentals of Traditional Distance Relaying ............................... 4  
  2.2 Derivation of Traditional Fault Distance Equations ...................... 5  
  2.3 The Effect of Fault Resistance............................................................... 10  
  2.4 Stepped Distance Protection ................................................................. 11  

**CHAPTER 3:** The Advanced Distance Relay ............................................. 15  
  3.1 The Per-Unit Fault Locator Equation .................................................. 15  
  3.2 The Advanced Distance Relay .............................................................. 25  
  3.3 The Components of the Advanced Distance Relay ......................... 27  

**CHAPTER 4:** Advanced Distance Relay Performance Tests ...................... 44  
  4.1 Advanced Distance Relay Sample Test Cases ................................. 44  
  4.2 Sample Test Case Results ................................................................... 46  
  4.3 Comprehensive Developmental Testing ............................................... 52  

TABLE OF CONTENTS
## FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Phase-to-phase fault symmetrical component circuit</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Double phase-to-ground fault symmetrical component circuit</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>Three-phase fault symmetrical component circuit</td>
<td>8</td>
</tr>
<tr>
<td>2.4</td>
<td>Phase-to-ground fault symmetrical component circuit</td>
<td>9</td>
</tr>
<tr>
<td>2.5</td>
<td>Distance relay characteristics</td>
<td>11</td>
</tr>
<tr>
<td>2.6</td>
<td>Stepped Distance relaying zones of protection</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>Positive-sequence phase-to-ground fault superposition circuit</td>
<td>19</td>
</tr>
<tr>
<td>3.2</td>
<td>Phase-to-phase fault on transmission line</td>
<td>21</td>
</tr>
<tr>
<td>3.3</td>
<td>Phase-to-phase fault superposition circuit</td>
<td>22</td>
</tr>
<tr>
<td>3.4</td>
<td>Advanced distance relay characteristic: small fault-path resistance</td>
<td>25</td>
</tr>
<tr>
<td>3.5</td>
<td>Advanced distance relay characteristic: large fault-path resistance</td>
<td>25</td>
</tr>
<tr>
<td>3.6</td>
<td>Advanced distance relay system-level diagram</td>
<td>27</td>
</tr>
<tr>
<td>3.7</td>
<td>General-purpose positive-sequence directionality test</td>
<td>38</td>
</tr>
<tr>
<td>3.8</td>
<td>Advanced distance relay positive-sequence directionality test</td>
<td>39</td>
</tr>
<tr>
<td>3.9</td>
<td>CCVT: coupling capacitor voltage transformer</td>
<td>41</td>
</tr>
<tr>
<td>4.1</td>
<td>Simplified power system simulated in sample test cases</td>
<td>44</td>
</tr>
<tr>
<td>4.2</td>
<td>Test case '1A' (phase-to-ground): internal relay quantities</td>
<td>47</td>
</tr>
<tr>
<td>4.3</td>
<td>Test case '1A' (phase-to-ground): calculated per-unit fault distance</td>
<td>48</td>
</tr>
<tr>
<td>4.4</td>
<td>Test case '1A' (phase-to-ground): calculated per-unit fault distance</td>
<td>49</td>
</tr>
<tr>
<td>Figure</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.5</td>
<td>Test case '1A' (phase-to-phase): internal relay quantities</td>
<td>49</td>
</tr>
<tr>
<td>4.6</td>
<td>Test case '1A' (phase-to-phase): calculated per-unit fault distance</td>
<td>50</td>
</tr>
<tr>
<td>4.7</td>
<td>Test case '1A' (phase-to-phase): calculated per-unit fault distance</td>
<td>50</td>
</tr>
<tr>
<td>B.1</td>
<td>Test case '1B' (phase-to-ground): internal relay quantities</td>
<td>70</td>
</tr>
<tr>
<td>B.2</td>
<td>Test case '1B' (phase-to-ground): calculated per-unit fault distance</td>
<td>70</td>
</tr>
<tr>
<td>B.3</td>
<td>Test case '1B' (phase-to-ground): calculated per-unit fault distance</td>
<td>70</td>
</tr>
<tr>
<td>B.4</td>
<td>Test case '1B' (phase-to-phase): internal relay quantities</td>
<td>71</td>
</tr>
<tr>
<td>B.5</td>
<td>Test case '1B' (phase-to-phase): calculated per-unit fault distance</td>
<td>71</td>
</tr>
<tr>
<td>B.6</td>
<td>Test case '1B' (phase-to-phase): calculated per-unit fault distance</td>
<td>71</td>
</tr>
<tr>
<td>B.7</td>
<td>Test case '2A' (phase-to-ground): internal relay quantities</td>
<td>72</td>
</tr>
<tr>
<td>B.8</td>
<td>Test case '2A' (phase-to-ground): calculated per-unit fault distance</td>
<td>72</td>
</tr>
<tr>
<td>B.9</td>
<td>Test case '2A' (phase-to-ground): calculated per-unit fault distance</td>
<td>72</td>
</tr>
<tr>
<td>B.10</td>
<td>Test case '2A' (phase-to-phase): internal relay quantities</td>
<td>73</td>
</tr>
<tr>
<td>B.11</td>
<td>Test case '2A' (phase-to-phase): calculated per-unit fault distance</td>
<td>73</td>
</tr>
<tr>
<td>B.12</td>
<td>Test case '2A' (phase-to-phase): calculated per-unit fault distance</td>
<td>73</td>
</tr>
<tr>
<td>B.13</td>
<td>Test case '2B' (phase-to-ground): internal relay quantities</td>
<td>74</td>
</tr>
<tr>
<td>B.14</td>
<td>Test case '2B' (phase-to-ground): calculated per-unit fault distance</td>
<td>74</td>
</tr>
<tr>
<td>B.15</td>
<td>Test case '2B' (phase-to-ground): calculated per-unit fault distance</td>
<td>74</td>
</tr>
<tr>
<td>B.16</td>
<td>Test case '2B' (phase-to-phase): internal relay quantities</td>
<td>75</td>
</tr>
<tr>
<td>B.17</td>
<td>Test case '2B' (phase-to-phase): calculated per-unit fault distance</td>
<td>75</td>
</tr>
<tr>
<td>B.18</td>
<td>Test case '2B' (phase-to-phase): calculated per-unit fault distance</td>
<td>75</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Prelude

Over the last several decades, power systems have become increasingly overworked. The system load has risen steadily, but has not been matched by a correspondingly steady increase in generation. The implication is that power system equipment is required to operate closer to absolute performance limits than ever before. This is equivalent to stating that the power system stability margins are smaller than they have ever been. With no apparent end to the aforementioned trend in sight, a heavy burden must be shouldered by the power system protection due to the decreasing stability margins.

Decreases in stability margins loosely translate into decreases in the "critical clearing time”. The critical clearing time is the maximum time interval that a fault may remain on the system before causing instability. Decreases in the critical clearing time require an increase in the speed of operation of power system protection equipment, namely protective relays and circuit breakers, in order to maintain the stability of the system. Traditionally, power system protective relays have consisted solely of electromechanical devices. While these devices functioned adequately on the power system in existence at the time they were designed, they are sorely pressed to operate fast enough on the power system of today. Assuming that the load continues to increase without a corresponding increase in generation, as has been the trend thus far, electromechanical relays may be unable to operate fast enough to continue to be useful as a means of power system protection.

Due to the need for increased speed of operation, digital computer-based relays have emerged as an important protective tool. Only within about the last 25 years have advancements in microprocessor technology rendered computer relays feasible [1]. These advancements have led to faster, more cost-effective computers capable of rapidly performing complex calculations. These computers are ideal platforms for the implementation of digital relaying algorithms -- some of which have been around since the 1960's [1].

With the stage now set, the majority of the remainder of this document focuses on the synthesis and implementation of algorithms and computer code required to create a next-generation computer-based distance relay. An introduction to the problem solved by this particular implementation of distance relay is presented in the following section.
1.2 The Problem

An impedance relay, also known as a distance relay, is a protective device extensively employed on transmission lines which operates based upon the value of "apparent impedance" that it detects at its input terminals. During a fault, the apparent impedance is equal to the positive-sequence impedance between the relay location and the fault location. By convention, the relay is energized with the appropriate voltages and currents to facilitate calculation of the positive-sequence apparent impedance [2]. Since the apparent impedance is determined differently, depending upon the type of fault, it must be calculated for each possible fault type to be certain that the relay will provide complete protection for every type of fault possible (see chapter 2, section 2.2). A complicating factor, however, is fault-path resistance.

The problem of fault-path resistance is inherent to traditional distance relaying. Resistance in the fault-path introduces errors in the calculated apparent impedance. If the fault-path resistance and the associated errors are large enough, the relay may be deceived and believe that no fault is present within its "zone of protection", as described in chapter 2, section 2.3. In this instance, the relay does not signal its associated circuit breakers to open. This allows the fault to remain on the system, and may cause the power system to become unstable, barring the operation of any backup protection.

1.3 The Proposed Solution

The proposed solution to the problem of fault-path resistance in reference to distance relaying comes in the form of a next-generation, computer-based distance relay. The advanced distance relay, as it will be referred to throughout the remainder of this document, is based upon an algorithm which calculates the positive-sequence transmission line impedance between the relay location and the fault location, independent of fault-path resistance. The implications are that the advanced distance relay is nearly immune to the effects of fault-path resistance. Since misoperations due to fault-path resistance are a fundamental drawback of distance relaying, the immunity to fault-path resistance offered by the advanced distance relay is an enormous asset. In this sense, the advanced distance relay solves a fundamental problem of distance relaying.

The central component of the advanced distance relay is the per-unit fault locator equation, discussed at length in chapter 3. The per-unit fault locator equation determines the impedance between the relay location and the fault location, excluding the fault-path resistance. It is this equation and its immunity to resistance in the fault-path which leads to the classification of the advanced distance relay as a next-generation, computer-based relay.
1.4 Overview

Chapter 1 consists of an introduction to the material presented throughout the remainder of the document. It summarizes the motivation for computer-based relaying, and it describes a fundamental problem facing traditional distance relaying. Chapter 1 also introduces the manner in which the advanced distance relay solves the aforementioned problem. Finally, the chapter ends with a brief overview of the entire document.

Chapter 2 is intended to provide a brief background of traditional distance relaying in order to prepare the reader for the material to be presented in chapter 3. Traditional distance relaying is introduced via a review of the fundamentals that are necessary to facilitate the discussions that follow. Following the review, the derivations of the 10 traditional fault distance equations are presented. These equations are then discussed, along with the effect that fault-path resistance has upon them. Lastly, the stepped distance relay protection scheme, a common distance relaying arrangement, is discussed.

Chapter 3 is devoted to the advanced distance relay, and is the substance of this document. Initially, the per-unit fault locator equation is introduced and presented. Moreover, the derivations of the two versions of the fault locator equation are also included. Following a discussion of fault locator equation properties, the apparent relay characteristic of the advanced distance relay is described. Finally, the actual component algorithms of the advanced distance relay are presented and explained at some length.

Chapter 4 discusses various power system test cases that are simulated in order to test the performance of the advanced distance relay prototype. First, the test power system is presented and described, along with the faults that are simulated on it. Next, the results of the application of the advanced distance relay to the simulated faults on the power system are discussed.

Chapter 5 brings the body of this document to a close. First, the conclusions drawn from the first four chapters are summarized. Next, ideas for future work are presented.

The appendices contain supplementary material. First, appendix 'A' includes the actual "C" language source code of the advanced distance relay prototype. Next, example simulation results are presented in appendix 'B' to illustrate the relay's performance. Finally, appendix 'C' contains a table of EMTP trial cases that were used to test the relay during its development.
CHAPTER 2

TRADITIONAL DISTANCE RELAYING

2.1 Fundamentals of Traditional Distance Relaying

A distance relay is a device that measures voltages and currents and converts them into an equivalent impedance. Should a fault occur, the impedance seen by the relay is the total impedance between the relay location and the point of the fault. This impedance may be compared with the known impedance of the protected transmission line in order to determine the approximate fault location. Should a fault be determined to exist on a protected transmission line segment, the relay would command circuit breakers to open in order to isolate the fault and the line. Under normal (non-faulted) operating conditions, the impedance detected by the relay is significantly larger than the impedance that is detected during a fault. The ability of a distance relay to distinguish between fault and non-fault conditions and to approximately locate a fault during its occurrence suggests the usefulness of the device for the implementation of protection schemes. A typical protection method, called "stepped distance protection", is described in a subsequent section of this chapter.

The theory behind distance relaying is based largely upon the concept of impedance and its application to the R-X diagram. For a single-phase system, voltage, current, and power-factor angle are measured and used to calculate the impedance that the relay "sees". This impedance is a complex quantity with magnitude equal to the ratio of the measured voltage to the measured current, and with angle equal to the measured power-factor angle [2]. The detected impedance may be separated into its real and imaginary components to be plotted on the R-X diagram, or it may be plotted in polar form on the R-X diagram as a complex quantity. Both representations yield the same plot.

Symmetrical Components

For a three-phase system, the principles are the same, but the method employed to obtain the impedance viewed by the relay is different. Since faults on a three-phase system often result in phase imbalance, symmetrical components are utilized to study these faults. The method of symmetrical components allows an unbalanced three-phase system of phasors to be decoupled into 3 balanced three-phase systems of phasors. The 3 balanced systems are: the zero-sequence components, positive-sequence components, and negative-sequence components. The positive-sequence components all have the same magnitude, are each separated by 120°, and have the same phase sequence as the original system of phasors [3]. The negative-sequence components all have the same magnitude (but not necessarily the same as the positive-sequence components), are also separated by 120°,
and have a phase sequencing that is opposite to the original system of phasors [3]. Lastly, the zero-sequence components all have the same magnitude (but not necessarily the same as the positive-sequence or negative-sequence components), and have no separation between them [3]. Note that a single-phase system may be represented by symmetrical components with zero-sequence component magnitude and negative-sequence component magnitude equal to zero, and positive-sequence component magnitude equal to the phasor magnitude of the original system. In different terms, the positive-sequence phasors of a single-phase system are identically equal to phasors of the original system.

**Sequence Networks**

Another concept that is related to the method of symmetrical components is that of sequence networks. Just as there are 3 systems of symmetrical components in a three-phase system, there are also 3 sequence networks. The 3 networks are known as the zero-sequence network, the positive-sequence network, and the negative-sequence network. There is a direct correlation between symmetrical components and sequence networks which have the same prefix. The sequence networks represent the impedance of the original circuit to the corresponding symmetrical components [3]. To clarify, the positive-sequence network is the resulting circuit when only positive-sequence quantities exist [3]. In addition, the negative-sequence and zero-sequence networks are the resulting circuits when only the negative-sequence and zero-sequence quantities exist, respectively [3]. Sequence networks may be interconnected in differing configurations to facilitate the analysis of various types of imbalance. This technique is utilized in the derivations presented in the following section. However, a discussion of the rules for the construction of the sequence networks, as well as the analysis of unbalanced conditions utilizing sequence networks, is beyond the scope of this document. For a thorough treatment of the topic, see *Elements of Power System Analysis*, by W. D. Stevenson [3].

As previously mentioned, the utilization of symmetrical components to study faults on three-phase power systems results in the decomposition of the original system of phasors into 3 balanced three-phase systems of phasors. By convention, a distance relay is required to be provided with the quantities that permit the positive-sequence impedance to be determined [2]. The quantities required to calculate positive-sequence impedance are derived in the next section.

### 2.2 Derivation of Traditional Fault Distance Equations

It is commonly known that there are 10 types of faults that may occur on a three-phase power system. There are 3 possible phase-to-ground faults, 3 possible phase-to-phase faults, 3 possible double phase-to-ground faults, and 1 possible three-phase fault [2]. One may argue that there is also a three-phase-to-ground fault, but since it is indistinguishable from an ordinary three-phase fault, it is excluded from discussion. As stated in the
previous section, distance relays must be provided with the quantities that allow the
determination of the positive-sequence impedance [2]. These quantities differ depending
upon the type of fault present. Therefore, the derivation of the required quantities for
each type of fault is presented as follows. Note that the derivations follow closely the

**Phase-to-Phase Faults:**

![Diagram of Phase-to-Phase Fault Symmetrical Component Circuit](image)

**Figure 2.1** Phase-to-phase fault symmetrical component circuit.

Assume that the fault occurs between phases 'b' and 'c'.

\[
\begin{align*}
E_{1f} &= E_{2f} = E_1 - I_1 Z_{1f} = E_2 - I_2 Z_{1f} \\
E_1 - E_2 &= (I_1 - I_2) Z_{1f} \\
Z_{1f} &= \frac{E_1 - E_2}{I_1 - I_2}
\end{align*}
\]

Note that due to symmetrical components (with \( \alpha = 1 \angle 120^\circ \)):

\[
\begin{align*}
E_b &= E_0 + \alpha^2 E_1 + \alpha E_2 \\
E_c &= E_0 + \alpha E_1 + \alpha^2 E_2 \\
E_b - E_c &= (\alpha^2 - \alpha)(E_1 - E_2)
\end{align*}
\]

\[
\begin{align*}
I_b &= I_0 + \alpha^2 I_1 + \alpha I_2 \\
I_c &= I_0 + \alpha I_1 + \alpha^2 I_2 \\
I_b - I_c &= (\alpha^2 - \alpha)(I_1 - I_2)
\end{align*}
\]

Substitution yields the following expression for the phase-to-phase positive-sequence fault
impedance of a fault between phases 'b' and 'c':

CHAPTER 2
\[ Z_{1f} = \frac{E_1 - E_2}{I_1 - I_2} = \frac{(E_b - E_c)(\alpha^2 - \alpha)}{(I_b - I_c)(\alpha^2 - \alpha)} = \frac{E_b - E_c}{I_b - I_c} \]

The derivation is readily extended to other combinations of phase-to-phase faults.

**Double Phase-to-Ground Faults:**

![Diagram of double phase-to-ground fault symmetrical component circuit.]

The derivation proceeds the same as that of the phase-to-phase fault, yielding the same result. The result is reproduced here for a fault between phases 'b', 'c', and ground.

\[ Z_{1f} = \frac{E_1 - E_2}{I_1 - I_2} = \frac{(E_b - E_c)(\alpha^2 - \alpha)}{(I_b - I_c)(\alpha^2 - \alpha)} = \frac{E_b - E_c}{I_b - I_c} \]
Three-Phase Faults:

\[ E_1 = E_a = I_1 Z_{1f} = I_0 Z_{1f} \]
\[ E_0 = E_2 = 0 \]
\[ I_1 = I_a \]
\[ I_e = I_2 = 0 \]
\[ Z_{1f} = \frac{E_1}{I_1} = \frac{E_a}{I_a} = \frac{E_b}{I_b} = \frac{E_c}{I_c} \]

Note that due to symmetrical components (with \( \alpha = 1 \angle 120^\circ \)):

\[ E_b = \alpha^2 E_1 \]
\[ E_c = \alpha E_1 \]
\[ E_a - E_b = (1 - \alpha^2) E_1 \]

\[ I_b = \alpha^2 I_1 \]
\[ I_c = \alpha I_1 \]
\[ I_a - I_b = (1 - \alpha^2) I_1 \]

Substitution yields:

\[ Z_{1f} = \frac{E_1}{I_1} = \frac{(E_a - E_b)(1 - \alpha^2)}{(I_a - I_b)(1 - \alpha^2)} = \frac{E_a - E_b}{I_a - I_b} \]
\[ Z_{1f} = \frac{E_1}{I_1} = \frac{(E_b - E_c)(\alpha^2 - \alpha)}{(I_b - I_c)(\alpha^2 - \alpha)} = \frac{E_b - E_c}{I_b - I_c} \]
\[ Z_{1f} = \frac{E_1}{I_1} = \frac{(E_c - E_a)(\alpha - 1)}{(I_c - I_a)(\alpha - 1)} = \frac{E_c - E_a}{I_c - I_a} \]

Therefore, the expression for the three-phase positive-sequence fault impedance is:

\[ Z_{1f} = \frac{E_a}{I_a} = \frac{E_b}{I_b} = \frac{E_c}{I_c} = \frac{E_a - E_b}{I_a - I_b} = \frac{E_b - E_c}{I_b - I_c} = \frac{E_c - E_a}{I_c - I_a} \]
Phase-to-Ground Faults:

Assume that the fault occurs on phase 'a'.

\[ E_{1f} = E_1 - I_1 Z_{1f} \]
\[ E_{2f} = E_2 - I_2 Z_{1f} \]
\[ E_{0f} = E_0 - I_0 Z_{0f} \]
\[ E_{af} = 0 = E_{0f} + E_{1f} + E_{2f} \]

Substitution yields:

\[ E_{af} = 0 = (E_0 + E_1 + E_2) - (I_1 + I_2) Z_{1f} - I_0 Z_{0f} \]
\[ E_{af} = 0 = E_a - I_a Z_{1f} - (Z_{0f} - Z_{1f}) I_0 \]
\[ E_{af} = 0 = E_a - (I_a + \frac{(Z_{0f} - Z_{1f})}{Z_{1f}} I_0) Z_{1f} \]

Define the compensated current as: \( I'_a = I_a + \frac{(Z_{0f} - Z_{1f})}{Z_{1f}} I_0 \)

Substitution yields:

\[ E_a = I'_a Z_{1f} \]
Therefore, the phase-to-ground positive-sequence fault impedance for phase 'a' is:

\[ Z_{f} = \frac{E_{a}}{I_{a}} \]

The derivation is readily extended for phase-to-ground faults on phases 'b' or 'c'.

In summary, for phase-to-phase faults and double phase-to-ground faults, a distance relay must be provided with the ratio of the difference of the affected phase voltages, "(Ex-Ey)" , at the relay location to the difference in the affected phase currents, "(Ix-Iy)" , at the relay location. These differences in phase quantities are referred to as "delta" quantities [2]. For phase-to-ground faults, the distance relay must be provided with the ratio of the affected phase voltage, "Ex", at the relay location to the affected phase compensated current, "Ix", at the relay location. The positive-sequence impedance of a three-phase fault may be computed utilizing either method. When utilizing the phase-to-ground formula for the determination of the positive-sequence impedance of a three-phase fault, it is of interest to note that since there is no zero-sequence current present during three-phase faults, the compensated current reduces to the actual affected phase current, "Ix", at the relay location.

The implication of the previous discussion is that in order to be capable of detecting each type of fault on a protected transmission line, three distance relays are required to be configured so as to detect multiphase faults, while another three distance relays are required to be configured to detected phase-to-ground faults [2]. Therefore, a total of six distance relays are required to provide complete coverage of a protected transmission line.

2.3 The Effect of Fault Resistance

The previous section described the derivation of the positive-sequence impedance calculation for each possible type of fault. These derivations were constructed assuming that there was no fault-path resistance. In cases of faults through resistance, the impedance that is calculated by the distance relay is greater than the actual positive-sequence impedance between the relay location and the fault location. Depending upon the magnitude of the fault resistance, the relay may be deceived and decide that the fault is outside of the protected zone. This is a fundamental problem of distance relaying -- the fact that fault-path resistance may cause a distance relay to misjudge the location of the fault.

The fact that fault-path resistance may introduce errors into the impedance calculated by a distance relay has led to the implementation of various "relay characteristics". A relay characteristic is described by the area of the R-X diagram for which a distance relay will decide that a fault has occurred. Since fault-path resistance may alter the impedance that the relay computes, one method of reducing the impact of varying fault resistance is to
ensure that the relay characteristic includes additional area surrounding the protected transmission line impedance. In this manner, small variations in fault location or resistance will not have as detrimental an impact. Section 2.4 describes another method used to reduce the impact of fault-path resistance.

![Relay Characteristics](image)

**Figure 2.5** Distance relay characteristics.

Relay characteristics are identified according to four general types. The four types of relay characteristics are that of the impedance relay, the admittance relay, the reactance relay, and the quadrilateral relay [2]. Figure 2.5 illustrates the basic shapes of each of the four main types of relay characteristics. Typically, when the impedance calculated by the distance relay enters the shaded region of the relay characteristic, the distance relay indicates that a fault has occurred and takes the appropriate action. This action usually consists of sending a "trip" signal to the breakers that are capable of isolating the fault.

### 2.4 Stepped Distance Protection

Now that the basics of traditional distance relaying have been presented, it is possible to discuss the "stepped distance" transmission line protection scheme. Distance relays have a typical uncertainty of about 5% of the nominal line impedance [2]. This uncertainty prevents the relay from being able to protect a transmission line exactly. Therefore, the stepped distance scheme intentionally under-reaches the end of the line with one "zone", while it intentionally over-reaches the end of the line with another zone. Typically, a third zone is also included and over-reaches not only the protected line, but the line that follows it. As may be surmised, time coordination is critical to ensure that the system operates as desired. Throughout the rest of this section, it may be helpful to refer to figure 2.6, an R-X diagram illustrating the different zones of protection.
Zones of Protection

The first zone of protection, "zone 1", is commonly limited to 85% of the protected line impedance [2]. (Note that the end of the line at which the relay is located is assumed to be 0% of the line, while the end opposite the relay is considered to be 100% of the line.) If the vector denoting the protected line impedance is placed on the R-X diagram (with the 0% end of the line located at the origin), the zone 1 setting corresponds to a relay characteristic that intersects the protected impedance vector at about 85% of the total magnitude. This means that if the distance relay calculates a fault impedance that falls within the characteristic, a fault has been detected in zone 1. The zone 1 relay is set so that it will act without delay when it detects a fault.

The limit of "zone 2", the second zone of protection, is typically selected to be 120% to 150% of the protected line impedance [2]. On an R-X diagram, the zone 2 relay characteristic completely encompasses the protected line impedance. The action of the zone 2 distance relay, however, is intentionally delayed in order to allow the zone 1 relay time to act, should the fault be common to both zones. In addition, should the fault occur on the adjoining line segment and still be within zone 2 of the protected line (i.e. a fault at 105% of the protected line impedance), the delay allows the zone 1 protection of the adjoining transmission line to act to isolate the fault. A benefit of this type of arrangement is that the zone 2 of the protected line will trip breakers for faults within its jurisdiction on an adjoining line in the event that the adjoining line's zone 1 protection fails. For example, assume that a fault occurs, and the relay computes the fault impedance to be 50% of the protected line impedance. Both zones 1 and 2 would detect the fault. Zone 2 would not be allowed to take any action until a specified delay had passed. In this time, zone 1 would trip the appropriate breakers, and the fault would be cleared. Alternately, assume that the relay computes the fault impedance to be 95% of the protected line impedance.
Zone 1 would not detect the fault. Zone 2 would detect the fault and trip the appropriate breakers after the specified delay. The typical delay that is introduced for zone 2 is 0.3 seconds [2].

"Zone 3" acts as a backup for zones 1 and 2 of the protected line and for zones 1 and 2 of the adjoining line. The limit of the reach of the zone 3 relay is typically equal to the impedance of the protected line in addition to 150% of the impedance of the adjoining line [2]. The zone 3 relay commonly has an intentional time delay of approximately 1 second [2]. This delay ensures that both the zone 1 and 2 relays of both lines have ample opportunity to act in order to clear the fault.

Each zone has an associated group of breakers that it trips in the event of a fault. When the zone 1 protection operates, only the protected line is removed from service. When zone 2 operates, it isolates the protected line, and the adjoining line. This is necessary since zone 2 provides backup protection for the zone 1 relay of the adjoining line. When zone 3 operates, it not only removes the protected line and the one adjoining it; it also removes the next line "downstream" as well. It should be readily apparent that it is desirable to have the smallest zone trip in order to minimize the portion of the power system that is removed from service. While desirable, this is not always possible. Obviously, zone 2 will have to operate in instances of faults between 85% and 100% of the protected line length. Zone 3, however, should never have to operate so long as the protection on the adjoining line is functioning correctly. Clearly, time coordination of the three zones is essential.

Problems with Stepped Distance Relaying

Stepped distance relaying, while a popular method of distance protection, has several difficulties associated with its use. As previously mentioned, resistance in the fault-path tends to increase the impedance that is detected by the distance relay. The fact that the detected apparent impedance is greater than the true impedance of the line segment up to the point of the fault may result in the fault appearing to be located in a different zone. If the resistance is great enough, the relay may not be even aware that a fault is present.

To illustrate, assume that a fault occurs through a resistance at approximately 80% of the protected line impedance. Depending on the value of the fault-path resistance, the apparent impedance that the relay detects will vary. In the instance that the resistance is high enough, the fault will appear to exist in zone 2 instead of zone 1. This means that the fault will have to wait for the specified delay until zone 2 trips its breakers and isolates the fault. The undesirable aspects are two-fold. First, the fault remains on the power system 0.3 seconds longer than it would have had it been correctly detected and isolated by zone 1. Second, since zone 2 initiates the breaker trips, a larger section of the power system (the protected line and its downstream adjoining line) is removed than is necessary. Had
the fault been correctly isolated by zone 1, only the protected line would have been removed.

Assuming that the same fault occurs at 80% of the protected line impedance, but through a higher resistance, it is evident that the fault may appear to the relay to be located in zone 3. This is even worse, for the fault will remain on the power system for a full 1 second before zone 3 is allowed to trip its breakers. When the zone 3 protection does operate, it will remove a much larger section of the power system (the protected line, the adjoining line, and the next line downstream) than is required to isolate the fault.

Another problem with stepped distance protection is created by the transformers that condition the power system currents and voltages so that they are of a suitable level to serve as input signals for a distance relay. The CCVT, or Coupling Capacitor Voltage Transformer, is often used to produce a scaled-down replica of the power system voltage for input by a distance relay. The CT, or Current Transformer, is frequently used to produce a scaled-down replica of the power system current for input by a distance relay.

During steady-state power system conditions, the secondary voltage of the CCVT is a scaled representation of the primary voltage. A problem arises, however, when a power system transient occurs. During a transient condition on the power system, the secondary output of the CCVT ceases to be an accurate representation of the primary input voltage. The transient that results at the CCVT output is called a subsidence transient. Since the input to the distance relay (which is the output of the CCVT) is not a true representation of the events occurring on the power system during the subsidence transient, the relay can not be expected to function correctly for the duration of the subsidence transient.

During the normal steady-state operation of a power system, the CT faithfully reproduces a scaled-down version of the power system current that it is monitoring. The problem that occurs with the CT is that it may saturate during a high-current power system transient, such as a fault. During the time that the CT is saturated, the distance relay can not be expected to function correctly. Often, in order to prevent saturation, a CT with a larger core and a higher primary current rating is used. Obviously, CCVT subsidence transients and CT saturation must be considered when designing any power system protection scheme.
CHAPTER 3

THE ADVANCED DISTANCE RELAY

3.1 The Per-Unit Fault Locator Equation

In the previous chapter, the traditional distance calculation was explained to be sensitive to variations in fault-path resistance. Obviously, it would be advantageous for the purpose of distance relaying to have a distance equation that is not affected by resistance in the fault path. The per-unit fault locator equation fulfills this need. The per-unit fault locator equation (hereafter referred to as the "locator equation") yields the per-unit distance to the fault (on the protected transmission line positive-sequence impedance base), and is not affected by fault-path resistance. Therefore, the locator equation is ideally suited for distance relaying.

As with the traditional distance equations, the locator equation comes in two versions. The first version, known as the phase-to-ground version, calculates the fault location for phase-to-ground, double phase-to-ground, and three-phase faults. The second version, known as the phase-to-phase version, calculates the fault location for phase-to-phase faults, double phase-to-ground faults, and three-phase faults. Both versions must be used in order to be certain of obtaining the correct fault location, since the fault type will not be known beforehand. The version that is not applicable to the type of fault occurring will over-estimate the fault distance. Therefore, fault classification is not required -- the fault distance is simply the lesser of the calculated per-unit distances.

In order to provide coverage for all three phases of a transmission line, the phase-to-ground version of the locator equation must be applied to each phase, while the phase-to-phase version must be applied to each combination of phases. This results in six equations that must be computed -- three for the phase-to-ground version and three for the phase-to-phase version.

The parameters of the locator equation are the real and imaginary components of complex phasor quantities. While the complex phasor quantities must be available, calculation of the locator equation itself involves only real mathematical operations.

Phase-to-Ground Version of Per-Unit Fault Locator Equation

The phase-to-ground version of the per-unit fault locator equation is not only applicable in the case of phase-to-ground faults, but also double phase-to-ground faults and three-phase faults. The parameters of the equation are the real and imaginary components, as indicated, of the designated complex phasor quantities. All parameters are considered
post-fault values, unless otherwise indicated. The phase-to-ground version of the locator equation, which follows, is derived in a forthcoming section.

\[
k = \frac{E_{y'0} - E_{y'sf}}{R_s(I_y'y' - I_y'sf) - X_s(I_y'y' + I_y'sf)}
\]

The only scalar value in the formula is the result, which is defined as:

\[
k = \text{per-unit fault distance (on transmission line positive-seq. impedance base)}
\]

**Note:** The appended subscript 'r' or 'i' in the equation refers to the real or imaginary component, respectively, of the given quantity.

- \(y = \text{faulted phase 'y'}\)
- \(E_y = \text{phase 'y' voltage phasor at relay}\)
- \(I_{y'} = \text{fault current as seen by relay}\)
- \(I_{ypre} = \text{phase 'y' current phasor at relay}\)
- \(I_{ypre} = \text{pre-fault phase 'y' current phasor at relay}\)
- \(I'_y = I_y + m*I_0 + m'*I_{0m}\)
- \(I_0 = \text{zero-sequence current phasor at relay}\)
- \(I_{0m} = \text{zero-sequence current phasor in mutually coupled parallel line}\)
- \(m = (Z_0-Z_I)/Z_I\)
- \(m' = Z_{0m}/Z_I\)
- \(Z_{0m} = \text{zero-sequence mutual impedance}\)
- \(Z_0 = \text{zero-sequence impedance of protected line}\)
- \(Z_I = \text{positive-sequence impedance of protected line}\)
- \(R_I = \text{positive-sequence resistance of protected line}\)
- \(X_I = \text{positive-sequence reactance of protected line}\)

**Phase-to-Phase Version of Per-Unit Fault Locator Equation**

The phase-to-phase version of the per-unit fault locator equation is not only applicable in the case of phase-to-phase faults, but also double phase-to-ground and three-phase faults. The parameters of the equation are the real and imaginary components, as indicated, of the designated complex phasor quantities. All parameters are considered post-fault values, unless otherwise indicated. The phase-to-phase version of the locator equation, which follows, is derived in a forthcoming section.
\[ k = \frac{E_d - E_{d'}}{R_s(I_d' - I_{d''}) - X_s(I_d' + I_{d''})} \]

The only scalar value in the formula is the result, which is defined as:
\[ k = \text{per-unit fault distance (on transmission line positive-seq. impedance base)} \]

**Note:** The appended subscript 'r' or 'i' in the equation refers to the real or imaginary component, respectively, of the given quantity.

\[ y, z = \text{faulted phases 'y' and 'z', respectively} \]
\[ E = (Ey - Ez) \]
\[ Ey = \text{phase 'y' voltage phasor at relay} \]
\[ Ez = \text{phase 'z' voltage phasor at relay} \]
\[ I' = (Iy - Iz) \]
\[ Iy = \text{phase 'y' current phasor at relay} \]
\[ Iz = \text{phase 'z' current phasor at relay} \]
\[ If = [(Iy - Iz) - (I_{ypre} - I_{zpre})] \]
\[ I_{ypre} = \text{pre-fault phase 'y' current phasor at relay} \]
\[ I_{zpre} = \text{pre-fault phase 'z' current phasor at relay} \]
\[ R_I = \text{positive-sequence resistance of protected line} \]
\[ X_I = \text{positive-sequence reactance of protected line} \]

It is evident that the phase-to-phase version of the fault locator has the same structure as the phase-to-ground version. The difference is in the quantities that are the parameters of the equation.

**Derivation of Phase-to-Ground Version of Per-Unit Fault Locator Equation**

This section presents the derivation of (3.1), the phase-to-ground version of the per-unit fault locator equation. This derivation closely follows a derivation presented by A. G. Phadke and M. A. Xavier in the paper, "Limits to Fault Location Accuracy" [4]. The derivation supplied is for phase 'a', but it proceeds the same for phases 'b' and 'c' as well. Therefore, the result obtained is obviously for phase 'a', but it is readily extended to the general case as previously presented in (3.1).

The quantities included in the derivation are defined as follows:
\[ k = \text{per-unit fault distance (on transmission line positive-seq. impedance base)} \]
\[ Ea = \text{phase 'a' voltage phasor at relay} \]
\[ I_{af} = I_a - I_{a pre} = \text{fault current as seen by relay} \]
\[ I_a = \text{phase 'a' current phasor at relay} \]
\[ I_{a pre} = \text{pre-fault phase 'a' current phasor at relay} \]
\[ I_{af1}, I_{af2}, I_{af0} = \text{symmetrical components of fault current as seen by relay} \]
\[ I_f1, I_f2, I_f0 = \text{symmetrical components of true fault current} \]
\[ I_a' = I_a + m^*I_0 + m'^*I_{0m} \]
\[ I_0 = \text{zero-sequence current phasor at relay} \]
\[ I_{0m} = \text{zero-sequence current phasor in mutually coupled parallel line} \]
\[ m = (Z_0 - Z_L)/Z_L \]
\[ m' = Z_0m'/Z_L \]
\[ Z_{0m} = \text{zero-sequence mutual impedance} \]
\[ Z_0 = \text{zero-sequence impedance of protected line} \]
\[ Z_L = \text{positive-sequence impedance of protected line} \]
\[ R_L = \text{positive-sequence resistance of protected line} \]
\[ X_L = \text{positive-sequence reactance of protected line} \]
\[ R_f = \text{fault-path resistance} \]

**Note:** The appended subscript 'r' or 'i' throughout the derivation refers to the real or imaginary component, respectively, of the given quantity.

Assuming a transposed transmission line, and neglecting charging capacitance, the phase 'a' voltage at the relay location is represented by [4]:
\[ E_a = kZ_L(I_a + mI_0 + m'I_{0m}) + R_fI_f \]
\[ E_a = kZ_L(I_a') + R_fI_f \]

It is evident from the calculation of the symmetrical components of a phase-to-ground fault that:
\[ I_{f1} = I_{f2} = I_{f0} = (1/3)I_f \]

The positive-sequence distribution factor, 'd_f', describes the component of the positive-sequence fault current that flows in the line segment between the relay location and the fault location. To obtain the positive-sequence distribution factor, the concept of the positive-sequence fault superposition circuit must be introduced.
The positive-sequence fault superposition circuit is shown in figure 3.1. Note that the voltage source in the circuit is equivalent to the positive-sequence pre-fault voltage at the fault location. The voltages and currents of the positive-sequence fault superposition circuit, when added to the voltages and currents of the corresponding positive-sequence pre-fault superposition circuit, yield the positive-sequence post-fault voltages and currents. Therefore, it is apparent that the quantities determined from the circuit represented in figure 3.1 consist only of the incremental quantities introduced by the fault (i.e. no load current contribution).

The positive-sequence fault current \( I_{af1} \), calculated via superposition, is defined as follows. Note that \( I_{af1} \) is the positive-sequence post-fault current.

\[
I_{af1} = I_{af1}^{pre} + I_{af1}
\]

\[
I_{af1} = I_{af1} - I_{af1}^{pre}
\]

From the positive-sequence fault superposition circuit, the positive-sequence distribution factor may be determined as follows.

\[
I_{af1} = \frac{Z_{1f}''}{Z_{1f}'' + Z_{1f}'} I_{f1}
\]

Let \( d_1 = \frac{Z_{1f}''}{Z_{1f}'' + Z_{1f}'} = |d_1| e^{j\alpha} \)

\[
I_{af1} = d_1 I_{f1} = (1/3) d_1 I_f
\]

\[
I_f = \frac{3I_{af1}}{d_1}
\]

Substitution yields:

\[
E_a = kZ_{1f} I_s + \frac{3R_f}{d_1} I_{af1}
\]

\[
E_a = kZ_{1f} I_s + \frac{3R_f}{|d_1|} I_{af1} (e^{-j\alpha})
\]

Assume: \( \theta_1 = 0^\circ \)
Since the impedances involved in the calculation of the distribution factor are positive-sequence impedances, the associated phase angles are approximately equal [5]. Therefore, the distribution factor phase angle, $\theta_1$, which is the difference of two approximately equal angles, is nearly zero. "In many cases, the angle $\theta_1$ is zero, and it may be assumed to be identically zero without introducing a serious error in the calculation" [5]. This approximation is valuable, since in practice the distribution factor may not be available. However, if the distribution factor phase angle is known, it may be substituted into the formula to make the calculation exact, and to eliminate any error that may possibly be introduced by the approximation.

Let $R = \frac{3R_f}{d_1}$

$$E_a = kZ_1I_a' + RI_{af}$$

Experimental evidence has shown that utilization of the actual current, $I_{af}$, yields better performance than may be obtained by using its positive-sequence component, $I_{af1}$ [4]. Therefore, $I_{af}$ is substituted for $I_{af1}$. This alters the value of $R$ slightly, but the change is inconsequential due to the fact that $R$ is to be eliminated from the equation [4].

(3.3)  

$$E_a = kZ_1I_a' + RI_{af}$$

Converting the complex quantities into rectangular form and performing the multiplication yields:

\[
E_{ar} + jE_{ai} = k(R_x + jX_x)(I_{ar} + jI_{ai}) + RI_{afr} + jRI_{afi}
\]

\[
E_{ar} + jE_{ai} = k[(R_xI_{ar} - X_xI_{ai}) + j(R_xI_{ai} + X_xI_{ar})] + RI_{afr} + jRI_{afi}
\]

Separating the equation into real and imaginary components results in two real equations.

\[
E_{ar} = k(R_xI_{ar} - X_xI_{ai}) + RI_{afr}
\]

\[
jE_{ai} = j[k(R_xI_{ai} + X_xI_{ar}) + RI_{afi}]
\]

Multiplying the first equation by $I_{af}$ and the second equation by $I_{af}$ results in:

\[
E_{adI_{af}} = k(R_xI_{adI_{af}} - X_xI_{adI_{af}}) + RI_{adI_{af}}
\]

\[
E_{adI_{af}} = k(R_xI_{adI_{af}} + X_xI_{adI_{af}}) + RI_{afI_{af}}
\]

The two resulting equations are subtracted to obtain the following result. Note that the resistance, $R'$, is eliminated.

\[
E_{adI_{af}} - E_{adI_{af}} = k(R_xI_{adI_{af}} - X_xI_{adI_{af}}) - k(R_xI_{adI_{af}} + X_xI_{adI_{af}}) + RI_{afI_{af}} - RI_{afI_{af}}
\]

\[
E_{adI_{af}} - E_{adI_{af}} = k(R_xI_{adI_{af}} - X_xI_{adI_{af}}) - k(R_xI_{adI_{af}} + X_xI_{adI_{af}})
\]

Solving for $k'$ yields the per-unit fault locator equation for phase-to-ground, double phase-to-ground, and three-phase faults involving phase 'a'. The same derivation may be performed for phases 'b' and 'c' to obtain the generalized result stated in (3.1).

\[
k = \frac{E_{adI_{af}} - E_{adI_{af}}}{R_x(I_{adI_{af}} - I_{adI_{af}}) - X_x(I_{adI_{af}} + I_{adI_{af}})}
\]
Derivation of Phase-to-Phase Version of Per-Unit Fault Locator Equation

This section presents the derivation of (3.2), the phase-to-phase version of the per-unit fault locator equation. The derivation is based upon a formulation provided the author by Dr. A. G. Phadke. The derivation is presented specifically for a phase 'a' to phase 'b' fault, but it proceeds the same for any phase-to-phase fault. Therefore, the result is readily extended to the general case as previously presented in (3.2).

The quantities included in the derivation are defined as follows:

\[ k = \text{per-unit fault distance (on transmission line positive-seq. impedance base)} \]
\[ a, b = \text{faulted phases} \]
\[ E_A = \text{phase 'a' voltage phasor at relay} \]
\[ E_B = \text{phase 'b' voltage phasor at relay} \]
\[ I_A = \text{phase 'a' current phasor at relay} \]
\[ I_B = \text{phase 'b' current phasor at relay} \]
\[ I_f = \text{fault current} \]
\[ I_{Apre} = \text{pre-fault phase 'a' current phasor at relay} \]
\[ I_{Bpre} = \text{pre-fault phase 'b' current phasor at relay} \]
\[ Z_I = \text{positive-sequence impedance of protected line} \]
\[ R_f = \text{fault-path resistance} \]

![Diagram](image)

**Figure 3.2** Phase-to-phase fault on transmission line.

From figure 3.2, it is apparent that the application of Kirchhoff's Voltage Law yields:

\[ E_A - E_B = kZ_1(I_A) - kZ_1(I_B) + I_f R_f \]
Rearranging terms yields:

\[ E_A - E_B = kZ_1(I_A - I_B) + I_f R_f \]

(3.4) \[ E_A - E_B = kZ_1(I_A - I_B) + \frac{1}{2}(I_f + I_f)R_f \]

Notice that the phase currents do not include mutually coupled zero-sequence components. There are two reasons for this. First, phase-to-phase faults have no zero-sequence components. However, in the case of double phase-to-ground faults, zero-sequence currents may exist, but the mutually coupled zero-sequence components of the phase currents are canceled due to symmetry. Second, the symmetry of phase-to-phase faults is such that the mutually coupled contributions of parallel lines are canceled. The cancellation is readily apparent if phase currents with mutually coupled zero-sequence parallel line contributions are substituted into (3.4).

![Phase-to-phase fault superposition circuit](image)

**Figure 3.3** Phase-to-phase fault superposition circuit.

Analysis of the phase-to-phase fault superposition circuit in figure 3.3 results in the following relations.

\[ I_{Af} = \frac{2 \cdot Z''_1}{2 \cdot Z'_1 + 2 \cdot Z''_1} I_f = \frac{Z''_1}{Z'_1 + Z''_1} I_f \]

\[ I_{Bf} = -\frac{2 \cdot Z''_1}{2 \cdot Z'_1 + 2 \cdot Z''_1} I_f = -\frac{Z''_1}{Z'_1 + Z''_1} I_f \]

Therefore, the distribution factor, determined from the fault superposition circuit in figure 3.3, is defined as follows.

Let \[ d = \frac{Z''_1}{Z'_1 + Z''_1} = |d|e^{j\theta} \]

Via the principle of superposition, the phase 'a' and phase 'b' fault currents are:

\[ I_{Af} = dI_f = I_A - I_{Apre} \]

\[ I_{Bf} = -dI_f = I_B - I_{Bpre} \]
Substituting the previous relations into (3.4) yields:

$$E_A - E_B = kZ_1(I_A - I_B) + \frac{R_f}{2d} \left[ (I_A - I_{Apre}) - (I_B - I_{Bpre}) \right]$$

$$E_A - E_B = kZ_1(I_A - I_B) + \frac{R_f}{2|d|} e^{-j\theta} \cdot \left[ (I_A - I_B) - (I_{Apre} - I_{Bpre}) \right]$$

Assume: $\theta = 0^\circ$

As in the case of the phase-to-ground fault locator derivation, the impedances involved in the calculation of the distribution factor are positive-sequence impedances. For this reason, the associated phase angles are approximately equal [5]. Therefore, the distribution factor phase angle, $\theta$, which is the difference of two approximately equal angles, is nearly zero. "In many cases, the angle $\theta$ is zero, and it may be assumed to be identically zero without introducing a serious error in the calculation" [5]. As before, this approximation is valuable, since the distribution factor may not be available. However, if the distribution factor phase angle is known, it may be substituted into the formula to make the calculation exact, and to eliminate any error that may possibly be introduced by the approximation.

Let $R = \frac{R_f}{2|d|}$

(3.5)  

$$E_A - E_B = kZ_1(I_A - I_B) + R \left[ (I_A - I_B) - (I_{Apre} - I_{Bpre}) \right]$$

Comparing (3.5) with (3.3) reveals that both are identical, provided that certain substitutions are performed. If the following assignments are made, and the quantities are substituted into (3.3), then (3.5) results.

Let $I'_A = (I_A - I_B)$
Let $I'_B = (I_A - I_B) - (I_{Apre} - I_{Bpre})$
Let $E_A = (E_A - E_B)$

Therefore, the result of the derivation of the phase-to-phase version of the fault locator, (3.2), is the same as the result of the derivation of the phase-to-ground version, (3.1), provided that the previous substitutions are made. As a side note, the value of $R'$ is not the same in (3.5) as in (3.3). This is inconsequential, however, since any scalar multiple of $R'$ is eliminated from the equations, as depicted in the phase-to-ground fault locator derivation.

**Properties of the Per-Unit Fault Locator Equation**

The derivations of both versions of the fault locator equation include certain assumptions that must be considered before utilization of the locator equation. One of these assumptions is that the fault current distribution factor phase angle must be zero. This assumption is necessary if the distribution factor phase angle is not known. In the event
that the distribution factor phase angle is known, the simplification performed in the
derivations is not necessary, and the per-unit locator equation may be modified to
accurately account for the non-zero distribution factor angle. In most practical instances,
however, the network impedances are not known, and therefore the distribution factor is
not known. For these instances, the assumption of a distribution factor angle of zero is
necessary. This zero angle assumption is quite good in practical systems, and is typically
within one degree of being correct [5].

The second assumption required in the derivation of the locator equation is that the zero-
sequence mutual impedance and the zero-sequence current of mutually coupled parallel
lines must be known. In the event that these parameters are not known, they must be
omitted from the equation. This omission introduces an error in the calculated fault
location. This error is related to the zero-sequence current in the parallel line, as well as
to the amount of zero-sequence coupling that is present. The error causes the distance to
be most-likely overestimated during remote faults, and underestimated during near faults
[5]. This discussion of the second assumption required in the derivation of the locator
equation applies only to the phase-to-ground version, (3.1). The phase-to-phase version is
exempt from this discussion due to the fact that the symmetry of the fault circuit leads to a
cancellation of all zero-sequence current contributions.

*Apparent Relay Characteristic of the Advanced Distance Relay*

Chapter 2 discussed the concept of the relay characteristic. For traditional impedance
(distance) relays, the relay characteristic is static. In other words, the relay characteristic
is fixed on the R-X diagram and does not vary. The advanced distance relay, however,
has a dynamic characteristic.

The characteristic of the advanced distance relay is more precisely termed an "apparent
relay characteristic". This is because the characteristic changes depending upon the fault
conditions. The advanced distance relay characteristic is defined by the per-unit fault
locator equation and the directionality equation, which is discussed later.

The apparent characteristic varies according to the fault-path resistance. The result is that
the advanced distance relay is nearly immune to the effects of fault-path resistance. The
relay bases its protective decisions solely on the value of reactance between the relay
location and the fault location. Since the apparent characteristic of the advanced distance
relay varies, the most effective method in which to diagram it is to present the
characteristics of two extreme cases. The characteristic that corresponds to the first case,
a fault with small fault-path resistance, is depicted in figure 3.4. The characteristic that
corresponds to the second case, a fault with large fault-path resistance, is presented in
figure 3.5.
3.2 The Advanced Distance Relay

The traditional distance relays discussed in chapter 2 have a serious drawback -- they are prone to errors in fault location due to variations in the apparent impedance caused by variable fault-path resistance. Protection schemes have attempted to minimize the effects of this source of error by employing overlapping zones of protection, as discussed in chapter 2. However, this practice is not always successful. A more desirable course of action would be to minimize the sensitivity of the relay to resistance in the fault path. The advanced distance relay program does just this -- it minimizes the sensitivity of the relay to resistance in the fault path.

The advanced distance relay is a high-speed computerized relay. Its main algorithm uses a Discrete Fourier Transform (DFT) operating on a 1/2 cycle data window to calculate the phasors that correspond to the input samples, which are taken at a frequency of 1440 samples / second, or 24 samples / cycle. The significance of this is that the relay can approximately locate the fault in just over 1/2 of a cycle. This is an improvement of nearly 1 to 2 cycles over the performance of traditional distance relays. An anti-aliasing filter with a cutoff of around 700 Hz is required to filter the input signals before sampling. Due to the speed requirement, the filter must be hardware implemented -- Finite Impulse Response (FIR) digital filtering algorithms are too slow to be of practical use. The advanced distance relay uses a computerized "mimic" circuit to remove any DC offset that may exist in the sampled input currents. This rejection of DC offset increases the reliability and accuracy of the relay during transient power system phenomena. The voltage inputs are processed by an "inverse CCVT" routine which eliminates subsidence transients (see section 2.4, Problems with Stepped Distance Relaying). By removing the subsidence transients, the routine dramatically increases the speed with which the relay can make a valid decision. A "transient monitor" algorithm acts as a fault detector, or more precisely, a transient detector. When a transient (fault, switching event, line energization, etc.) is detected, the transient monitor routine activates the fault directionality and locator routines to determine if the transient is a fault, and if so, the fault location. The advanced
distance relay analyzes positive-sequence voltages and currents in order to determine the fault's direction. This "directionality" check prevents the relay from tripping for faults occurring "behind" the relay. The information provided by the fault locator algorithm (described in the previous section) in conjunction with the directionality algorithm is used by the relay to make "trip / block (no trip)" decisions based on fault distance and direction. In instances of faults very near the relay location, the "memory voltage" is used by the distance calculation. This improves the ability of the relay to function correctly for close-in faults.

The "heart" of the advanced distance relay is the per-unit fault locator equation, described in the previous section. As previously discussed, the locator equation produces the correct fault location for a wide range of fault-path resistances. The remaining components that make up the relay exist largely to provide the quantities required for input by the locator equation. These components are introduced and explained in the section that follows.
3.3 The Components of the Advanced Distance Relay

The advanced distance relay is composed of many different algorithms. Its system-level diagram is presented in figure 3.6. Each component of the system-level diagram, along with its function, is described in the material that follows.

![System-level diagram of advanced distance relay](image)

**Figure 3.6** Advanced distance relay system-level diagram.
Anti-Aliasing

Since the advanced distance relay samples the input currents and voltages, the "aliasing" phenomenon must be considered. Aliasing is a function of sampling frequency. It is a condition whereby an original continuous-time waveform can no longer be reconstructed from its corresponding discrete-time samples.

Paraphrased, the sampling theorem states that the sampling frequency must be more than twice the highest frequency component in the continuous-time waveform to prevent aliasing. Since the sampling frequency of the advanced distance relay is 1440 samples / second, the highest allowable frequency component in the continuous-time waveform must be less than 720 Hz. While the fundamental power system frequency is 60 Hz, there are always higher frequency components present during transient conditions. Therefore, the continuous-time signal must be low-pass filtered to removed the undesirable high frequency components. A low-pass filter, used for this purpose, is known as an "anti-aliasing" filter.

A cutoff frequency of 700 Hz was chosen for the second-order anti-aliasing filter employed by the advanced distance relay. This cutoff frequency is low enough to prevent aliasing, yet high enough to pass several harmonics with minimal attenuation. The anti-aliasing filter must be hardware implemented, as previously mentioned, due to the speed requirements of the relay. Finite Impulse Response (FIR) digital filtering algorithms are simply too slow to be of practical use for on-line application.

The Mimic Circuit Algorithm

Power system transients are often the cause of large DC current offsets. These offsets introduce errors into the calculated phasors. Subsequently, the fault location determined by the per-unit fault locator equation would be erroneous due to the errors in the phasors. Therefore, the DC offset must somehow be removed from the input current signals.

The traditional method of removing a DC current offset is to use a "mimic" circuit. The circuit is so-named because it mimics the power system fault-path impedance (X/R ratio). The input current of the mimic circuit is the current containing the DC offset, while the output voltage of the mimic circuit is a DC offset-free, scaled representation of the input current. The mimic circuit consists of a resistance, 'r', and an inductance, 'x', connected in series. The choice of the correct ratio, 'x/r', is imperative to ensure removal of the DC current offset. In order to ensure elimination of the offset, the ratio 'x/r' must be made equal to the transmission line positive-sequence X/R ratio. This sounds straightforward, but one must bear in mind that the X/R ratio may change due to switching or fault resistance [1].
As stated, the output voltage of the mimic circuit is a DC-free, scaled representation of the input current signal, assuming the correct choice of 'r' and 'x'. It must be noted, however, that the phase of the "mimic voltage" leads the phase of the input current by the impedance angle of the mimic circuit. This error must be taken into account because it alters the observed relative phase angle between the power system current and the power system voltage. It is also important to note that the magnitude of the mimic voltage is a scaled replica of the input current. The choice of values for 'r' and 'x' determines the amount of scaling present.

The advanced distance relay employs an algorithm that duplicates the function of the analog mimic circuit. The mimic algorithm utilized by the advanced distance relay operates on the discrete current samples, which are taken at a rate of 1440 samples / second. After processing by the mimic algorithm, the current samples are DC offset-free; however, the phase of the current waveform is advanced by the "mimic angle", which is chosen to be equal to the positive-sequence line impedance angle. As with the analog mimic circuit, the output quantity, called the "mimic current", is not only phase shifted, but magnitude scaled as well. It is desirable to choose the parameters of the mimic equation so that the magnitude scaling factor is unity. The manner in which this is accomplished is illustrated by the following derivation [6].

The time-domain equation that models the mimic circuit is:

\[ v = r \cdot i + I \cdot \frac{di}{dt} \]

Transforming this time-domain equation into a phasor equation at 60 Hz yields:

\[ V' = (r + jx)I \]

It is desirable for the mimic circuit to have a scaling factor of unity. This ensures that the mimic circuit output waveform is the same magnitude as the input waveform. It is evident from the above equation that a mimic impedance of unity magnitude will produce the desired mimic scaling factor. The mimic impedance unity magnitude is obtained via the following assignments:

Let \( r = \cos \psi \) and \( x = \sin \psi \)

Noting that

\[ I = \frac{x}{\omega_0} = \frac{\sin \psi}{\omega_0} \quad \text{with} \quad \omega_0 = 377 \text{ rad / sec (power system frequency)} \]
And substituting back into the time-domain equation yields:

\[ i'' = v = i \cdot \cos \psi + \frac{di}{dt} \cdot \frac{\sin \psi}{\omega_0} \]

\[ i'' = \text{mimic output "current" (which, in reality, is a voltage)} \]
\[ i = \text{mimic input current (possibly containing DC offset)} \]

The only value still left to determine is \('\psi'.\) The mimic \('x/r'\) ratio is given by

\[ \frac{x}{r} = \frac{\sin \psi}{\cos \psi} = \tan \psi \]

And the transmission line \(X/R\) ratio is:

\[ \frac{X}{R} = \frac{(\sqrt{R^2 + X^2}) \cdot \sin \Phi_1}{(\sqrt{R^2 + X^2}) \cdot \cos \Phi_1} = \tan \Phi_1 \]

\[ \Phi_1 = \text{transmission line positive-sequence impedance angle} \]

Since the \('x/r'\) ratio of the mimic circuit is required to be equal to the transmission line \(X/R\) ratio, it follows that:

\[ \psi = \Phi_1 = \tan^{-1} \left( \frac{X_1}{R_1} \right) = \text{positive-sequence impedance angle} \]

Therefore, the fully determined time-domain equation that describes the mimic circuit is:

\[ i'' = v = i \cdot \cos \Phi_1 + \frac{di}{dt} \cdot \frac{\sin \Phi_1}{\omega_0} \]

To apply this equation to the sampled input current data, it must be converted from a continuous-time equation to a difference equation. This is accomplished by converting the derivative to a ratio of differences. When the \('n-1' mimic output point is being calculated, if points \('n' and \('n-2'\) are used to calculate the numerator of the derivative and if the denominator of the derivative is equal to twice the sampling period, then the three point mimic equation results.

\[ i''_{n-1} = i_{n-1} \cdot \cos \Phi_1 + \frac{i_n - i_{n-2}}{2\delta} \cdot \frac{\sin \Phi_1}{\omega_0} \]

\[ i''_{n-1} = \text{'n-1' mimic output sample (free of DC offset)} \]
\[ i_n = \text{'n' mimic input current sample} \]
\[ \delta = \text{sampling period} \]
\[ \omega_0 = \text{power system frequency (rad / sec)} \]
\[ \Phi_1 = \text{positive-sequence impedance angle of transmission line} \]

CHAPTER 3
In the case of the advanced distance relay, which has a sample period of \((1/1440) = 6.95 \times 10^{-4}\) seconds / sample, formula (3.6) reduces to:

\[
i_{n-1}'' = i_{n-1} \cdot \cos \Phi_1 + \frac{i_n - i_{n-2}}{\pi/6} \cdot \sin \Phi_1
\]

It is extremely important to remember that the mimic circuit advances the phase of the mimic circuit output voltage, which is referred to as the "mimic current". The amount of phase advance is equal to the mimic angle, which has been chosen to be equal to the positive-sequence impedance of the transmission line. This phase advance must be removed before the mimic current will be useful to the relay.

**The Transient Monitor Algorithm**

The "transient monitor" routine acts as the relay's fault sentinel. Whenever a transient condition is detected on the power system (fault, switching event, line energization, etc.), the transient monitor activates the directionality check algorithm and the fault locator algorithm. In this sense, the transient monitor is employed as a fault detector.

As will be discussed shortly, the phasor calculation routine estimates the fundamental frequency (60 Hz) phasor representation of voltages and currents from sample data. The transient monitor reverses this process. The transient monitor estimates the current samples that correspond to a given current phasor. It then compares the estimated current samples to the actual mimic current samples. The residual is the quantity formed by subtracting an estimated current sample value from an actual mimic current sample value.

As stated, the advanced distance relay uses a 1/2 cycle window which contains 12 mimic current samples at one time. Therefore, the sum of the absolute values of the 12 residuals calculated from the estimated currents and the data window mimic currents is the transient monitor value, 'T'. In simpler terms, the transient monitor 'T' is equal to the 1-norm of the residual vector. Of course, as the newest mimic current sample enters the data window (and the oldest sample exits the window), the value of 'T' is recalculated. Therefore, the computation of 'T' must be performed each time a new mimic current sample becomes available. The derivation of the 1/2 cycle transient monitor algorithm proceeds as follows [6].
The 1/2 cycle (12 samples) RMS phasor calculation formula, to be described later, is:

\[
\begin{bmatrix}
X_r \\
X_i
\end{bmatrix} = \frac{\sqrt{2}}{12}
\begin{bmatrix}
\cos \theta & \cos 2\theta & \ldots & \cos 12\theta \\
-\sin \theta & -\sin 2\theta & \ldots & -\sin 12\theta
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_{12}
\end{bmatrix}
\]

\(X_r\) = estimated fundamental frequency real phasor component
\(X_i\) = estimated fundamental frequency imaginary phasor component
\(x_n\) = actual data window mimic current sample number \(n\) (1 ≤ n ≤ 12)
\(\theta\) = sample interval in radians = \((120\pi)\)*(sampling frequency)^-1
\([\theta = (\pi/12)\) for 1440 Hz sampling rate used by advanced distance relay]

The previous formula estimates phasor values that correspond to given sample values. However, for application by the transient monitor algorithm, it is desired to estimate the samples that correspond to a given phasor value. In order to accomplish this, the expression for the least squares estimate of the sample values must be utilized [1]. This expression is defined as follows.

\[
\begin{bmatrix}
\bar{x}_1 \\
\bar{x}_2 \\
\bar{x}_3 \\
\vdots \\
\bar{x}_{12}
\end{bmatrix} = \sqrt{2}
\begin{bmatrix}
\cos \theta & -\sin \theta \\
\cos 2\theta & -\sin 2\theta \\
\cos 3\theta & -\sin 3\theta \\
\vdots & \vdots \\
\cos 12\theta & -\sin 12\theta
\end{bmatrix}
\begin{bmatrix}
\bar{X}_r \\
\bar{X}_i
\end{bmatrix}
\]

\(\bar{x}_n\) = estimated sample value that corresponds to data window actual mimic current sample number \(n\) (1 ≤ n ≤ 12)
\(\theta\) = sample interval in radians = \((120\pi)\)*(sampling frequency)^-1
\(\bar{X}_r\) = estimated fundamental frequency real phasor component
\(\bar{X}_i\) = estimated fundamental frequency imaginary phasor component

Substituting (3.7) into (3.8) and performing the multiplication yields the following formula that predicts the estimated current samples directly from the actual data window mimic current samples.
\[
\begin{align*}
\bar{x}_n &= \text{estimated sample value that corresponds to data window actual} \\
&\quad \text{mimic current sample number 'n' (1 \leq n \leq 12)} \\
x_n &= \text{actual data window mimic current sample number 'n' (1 \leq n \leq 12)} \\
\theta &= \text{sample interval in radians} = (120\pi)\%(\text{sampling frequency})^{-1}
\end{align*}
\]

The components of the residual vector are of the form: \( r_n = x_n - \bar{x}_n \), with \( x_n \) and \( \bar{x}_n \) defined as in (3.9). Since a 1/2 cycle (12 sample) transient monitor is utilized, the dimension of the residual vector is 12 by 1. Performing the vector subtraction and simplifying yields the following direct expression for the residual vector.

\[
\begin{align*}
\begin{bmatrix}
r_1 \\
r_2 \\
r_3 \\
r_{12}
\end{bmatrix} &= \frac{1}{6} \begin{bmatrix}
5 & -\cos \theta & -\cos 2\theta & -\cos 3\theta & \ldots & -\cos 10\theta & -\cos 11\theta \\
-\cos \theta & 5 & -\cos \theta & -\cos 2\theta & \ldots & -\cos 9\theta & -\cos 10\theta \\
-\cos 2\theta & -\cos \theta & 5 & -\cos \theta & \ldots & -\cos 8\theta & -\cos 9\theta \\
-\cos 11\theta & -\cos 10\theta & -\cos 9\theta & -\cos 8\theta & \ldots & -\cos \theta & 5
\end{bmatrix} \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_{12}
\end{bmatrix}
\end{align*}
\]

\( r_n \) = residual corresponding to difference between the data window \\
\quad mimic current sample number 'n', (1 \leq n \leq 12), \text{and estimated sample} \\
\quad number 'n', (1 \leq n \leq 12). \\
x_n = \text{actual data window mimic current sample number 'n' (1 \leq n \leq 12)} \\
\theta = \text{sample interval in radians} = (120\pi)\%(\text{sampling frequency})^{-1}

While the above expression allows the computation of the residual vector from the mimic current sample data contained in the 1/2 cycle data window, it requires that many computer floating point operations be performed. Since operating speed is extremely important with respect to the relay application, it is desirable to reduce the number of calculations that must be performed by the computer. The following recursion formulas significantly reduce the number of computations that must be performed each time a new residual vector is calculated. Before the recursion formulas may be used, however, the residual vector must first be initialized via the preceding equation. The initialization may occur any time after the first 12 mimic current input samples enter the data window. Once
the residual vector has been initialized, the recursion formulas may be utilized exclusively. The recursion formulas are defined as follows [6].

\[
\Delta x = \frac{1}{6}(x_i + x_{i+3})
\]

\[
r_{i}^{\text{new}} = r_{i}^{\text{old}} + \Delta x \cdot \cos \theta
\]

\[
r_{2}^{\text{new}} = r_{3}^{\text{old}} + \Delta x \cdot \cos2 \theta
\]

\[
\ldots
\]

\[
r_{11}^{\text{new}} = r_{12}^{\text{old}} + \Delta x \cdot \cos11 \theta
\]

\[
r_{12}^{\text{new}} = -r_{1}^{\text{old}} + 5 \cdot \Delta x
\]

Note that mimic current sample 'x_i' is the most recent sample to enter the 1/2 cycle data window, while mimic current sample 'x_{i+3}' is the sample that most recently exited the data window. Following initialization, the recursion formula is applied as each new mimic current sample enters the 1/2 cycle data window. Obviously, the recursion formula greatly simplifies the computational process.

Once the residual vector has been calculated, the transient monitor 'T' is obtained by computing the 1-norm of the residual vector. The 1-norm of the residual vector, and hence 'T', is defined as follows [6].

\[
T = \sum_{k=1}^{12} |r_k|
\]

The transient monitor value, 'T', is an indicator of the fit of the actual input waveform to the sinusoid that corresponds to the estimated phasor. For small values of 'T', the fit is quite good. When the value of 'T' becomes large, the fit is poor. A large value of 'T', and the fact that it suggests a poor fit, indicates that a rapid change has occurred on the power system. In this manner, 'T' is a transient indicator.

Since there is no definitive value that differentiates between a large 'T' and a small 'T', a relative scale is employed by the advanced distance relay. The utilization of the relative scale requires that the values of the relative maximum of 'T', "tmax", and the relative minimum of 'T', "tmin", be stored. Whether 'T' is considered large or small is determined in the following manner.

If 'T' is greater than "tmin" multiplied by an arbitrary constant, then 'T' is assumed to be large, and a transient is assumed to have occurred. Once 'T' is ruled large, "tmax" is assigned the corresponding value of 'T', and the transient monitor routine begins to watch for 'T' to fall. This is accomplished by comparing 'T' to the new value of "tmax". If 'T' becomes less than "tmax" divided by an arbitrary constant, then 'T' has become small again, and "tmin" is assigned the corresponding value of 'T'. The arbitrary constant used by the transient monitor routine in the advanced distance relay is 4. The relative level checking works very well, except when 'T' lingers in the vicinity of zero. In this case,
small variations may cause the transient monitor to decide that 'T' is large when it is actually small. For example, if the value of "tmin" is 0.1 and 'T' is 0.5, then 'T' would be ruled large, assuming that the value of the arbitrary constant were 4. In reality, a value of 'T' equal to 0.5 is rather small. In order to prevent this and other similar types of misoperation, an additional limitation must be placed upon the tests that determine the relative size of 'T'. Before 'T' may be ruled large, it must not only be greater than "tmin" multiplied by an arbitrary constant, but it must also be greater than an arbitrary absolute threshold. Conversely, for 'T' to be ruled small, it may either be less than "tmax" divided by an arbitrary constant, or it may be less than the arbitrary absolute threshold. For the advanced distance relay, this absolute threshold corresponds to a value of 0.5.

It is important to note that during the first 1/2 cycle of execution of the advanced distance relay program, the values of the phasor quantities, and hence the transient monitor residuals, change very rapidly as they are initialized. The transient monitor incorrectly interprets this as a power system transient. This "initialization transient" is handled very simply by having the advanced distance relay program ignore the very first transient detected. Since this all occurs in less than the first cycle (16.67 milliseconds) of program operation, it is not an inconvenience.

The DFT Phasor Calculation Algorithm

The Discrete Fourier Transform (DFT) phasor calculation algorithm converts sampled continuous-time waveforms into corresponding fundamental frequency RMS phasor representations. The sampled input voltage waveforms are converted to RMS voltage phasors, and the mimic current samples are converted to RMS current phasors. As discussed in the section concerning the mimic circuit algorithm, the mimic current samples are free of DC offset, and are therefore better suited for phasor conversion.

The advanced distance relay uses a 1/2 cycle Discrete Fourier Transform (DFT) to compute the fundamental frequency phasor from corresponding data samples. Since the sampling frequency is 1440 Hz (24 samples / cycle), the 1/2 cycle DFT algorithm requires 12 sample points with which to compute the corresponding phasor. Since a phasor is merely a fundamental harmonic frequency-domain representation of a time-domain quantity, the DFT is considered ideal to perform the conversion. The 1/2 cycle DFT phasor calculation equations are defined as follows [1].

CHAPTER 3
\[
\overline{X}_r = \frac{\sqrt{2}}{K} \sum_{n=1}^{K} x_n \cdot \cos(n\theta)
\]

\[
\overline{X}_i = \frac{\sqrt{2}}{K} \sum_{n=1}^{K} x_n \cdot \sin(n\theta)
\]

\(\overline{X}_r\) = estimated fundamental frequency real phasor component
\(\overline{X}_i\) = estimated fundamental frequency imaginary phasor component
\(x_n\) = sample 'n' of a 1/2 cycle continuous-time waveform (1 ≤ n ≤ K)
\(\theta\) = sample interval in radians = \((120\pi)\ast(sampling\ frequency)^{-1}\)
\(K\) = number of samples per 1/2 cycle

Expressing the 1/2 cycle DFT phasor calculation in matrix notation yields [6]:

\[
\begin{vmatrix}
\overline{X}_r \\
\overline{X}_i
\end{vmatrix} = \frac{\sqrt{2}}{K} \begin{vmatrix}
\cos \theta & \cos 2\theta & \ldots & \cos K\theta \\
-\sin \theta & -\sin 2\theta & \ldots & -\sin K\theta
\end{vmatrix} \begin{vmatrix}
x_1 \\
x_2 \\
\vdots \\
x_K
\end{vmatrix}
\]

\(\overline{X}_r\) = estimated fundamental frequency real phasor component
\(\overline{X}_i\) = estimated fundamental frequency imaginary phasor component
\(x_n\) = sample 'n' of a 1/2 cycle continuous-time waveform (1 ≤ n ≤ K)
\(\theta\) = sample interval in radians = \((120\pi)\ast(sampling\ frequency)^{-1}\)
\(K\) = number of samples per 1/2 cycle

In the case of the advanced distance relay, \('K' = 12\) samples per 1/2 cycle. Also, \('\theta' = \pi/12\) radians.

The problem with using (3.10) to calculate phasors from the 1/2 cycle data window samples is that as new samples move into the data window, the computed phasor rotates clockwise at a constant rate of \('\theta'\) radians per sample, which is equal to the sample interval [1]. If ratios of phasors are to be computed, the rotation is not a problem because the constant rate of rotation of each phasor cancels. When calculations other than ratios must be computed, however, the phasor rotation must be eliminated. To clarify, the phase-to-ground version of the per-unit fault locator, equation (3.1), relies on pre-fault and post-fault phasor data. The erroneous rotation would render (3.1) useless, and therefore must be eliminated.

Another limitation of (3.10) is the relatively large number of computations that must be performed. It is important to bear in mind that due to the real-time nature of operation of
a relay, the number of computations in a given algorithm must be minimized in order to maximize the overall speed at which the relay operates.

Both of the aforementioned limitations may be corrected by implementing the calculations in the form of a recursion relationship, as in the case of the transient monitor algorithm. The problematic phasor rotation is eliminated by introducing a compensating term into the recursion relationship to advance the phasor counter-clockwise by 'Θ' radians per sample. The effect of the compensating term is to cancel the clockwise rotation. Therefore, the phasor remains stationary. The phasor computation is obviously simplified via the introduction of the recursion relation, resulting in an increase in computational speed. The recursion relation is given by [6]:

\[ Δx_k = \frac{\sqrt{2}}{12}(x_k + x_{k-12}) \]

\[ \bar{X}_r^{\text{new}} = \bar{X}_r^{\text{old}} + Δx_k \cdot \cos\left(\frac{k\pi}{12}\right) \]

\[ \bar{X}_i^{\text{new}} = \bar{X}_i^{\text{old}} - Δx_k \cdot \sin\left(\frac{k\pi}{12}\right) \]

- \(\bar{X}_r^{\text{new}}\) = new estimated fundamental frequency real phasor component
- \(\bar{X}_i^{\text{new}}\) = new estimated fundamental frequency imaginary phasor component
- \(\bar{X}_r^{\text{old}}\) = old estimated fundamental frequency real phasor component
- \(\bar{X}_i^{\text{old}}\) = old estimated fundamental frequency imaginary phasor component
- \(k\) = sample index

The recursion relation is applied to the 1/2 cycle (12 sample) data window. The term, 'Δx_k', is recalculated each time the window advances, or in other words, as the newest sample enters the window and the oldest sample exits the window. Just after the window advances, the 'k'th sample becomes the newest sample that has just entered the window, and the '(k-12)'th sample becomes the oldest sample that has just exited the window.

The transcendental functions in the recursion relations depend on the value of 'k'. Since the sine and cosine are periodic, the value of 'k' may be reset to 1 at the end of any period without introducing error. Therefore, the simplest configuration is a continuous loop in which 'k' ranges from 1 to 24 and then begins again at 1.

As with the transient monitor algorithm, the phasor calculation algorithm must be initialized via (3.10) before the recursion relations may be utilized. The initialization is performed after the first 12 samples have entered the 1/2 cycle data window. Upon initialization, the recursion relations may be used exclusively.
The Positive-Sequence Directionality Algorithm

The positive-sequence "directionality" algorithm performs just as the name suggests. The algorithm computes the direction in which the fault occurs, with reference to the relay location, based on the positive-sequence voltages and positive-sequence mimic currents.

The advanced distance relay employs the directionality algorithm in conjunction with the per-unit fault locator equation in order to make "trip / block (no trip)" decisions. Not only must the fault distance be judged within the relay's zone of protection, but it must be also ruled a "forward fault" (i.e. fault in the direction of the relay's protective zones) before the relay will signal a trip. Should the fault occur behind the relay (i.e. a "reverse fault"), the relay will not trip its associated breakers.

The directionality algorithm is based on the following general-purpose equation. All of the quantities in the equation, including the result, represent complex, phasor values [7].

\[
D = \frac{E_1}{(I_1 - I_{1,\text{pre}})}
\]  

(3.11)

\[D = \text{general-purpose directionality}\]
\[E_1 = \text{positive-sequence faulted voltage phasor at relay location}\]
\[I_1 = \text{positive-sequence faulted actual current phasor at relay location}\]
\[I_{1,\text{pre}} = \text{positive-sequence faulted actual pre-fault current phasor at relay}\]

As stated, the result, 'D', is a complex value. The angle of 'D', in polar coordinates, indicates the fault directionality. The parameters of the equation are positive-sequence values due to the fact that faults often produce extreme conditions of unbalance.

![Figure 3.7 General-purpose positive-sequence directionality test.](image-url)

In reference to the general-purpose directionality equation (3.11), the line that divides the R-X diagram into the "forward" and "backward" zones intersects the origin of the R-X diagram, and is perpendicular to the positive-sequence line impedance, as illustrated in
For angles of \( D \) that are in the shaded area, the fault is considered a forward fault. For angles of \( D \) that are outside of the shaded area, the fault is considered a reverse fault. The mathematical description of the forward zone for the general-purpose directionality equation (3.11) is given by:

\[
\angle Z_i^* - 90^\circ \leq Angle(D)^\circ \leq 90^\circ + \angle Z_i^*
\]

\( \angle Z_i^* \) = angle of positive-sequence line impedance (degrees)

\( Angle(D)^\circ \) = directionality angle (degrees)

The general-purpose directionality equation, which employs the positive-sequence actual line current, must be modified for use with the advanced distance relay. As explained, the mimic current leads the actual line current by the mimic angle, which is equal to the positive-sequence impedance angle of the transmission line. This phase angle difference must be considered. Since the directionality equation contains the current phasors in the denominator, the phase lead that is equal to the mimic angle translates into an equivalent lag in the directionality angle. This lag is equal to the mimic angle, and is therefore equal to the positive-sequence impedance angle of the transmission line.

The boundary of the forward zone may be altered to compensate for the current phasor rotation caused by the mimic routine. This is accomplished by subtracting the positive-sequence line impedance from the boundary conditions of the mathematical description of the forward zone of the general-purpose directionality equation. The modified directionality equation, which operates on mimic currents instead of actual currents, is given in (3.12). The mathematical description of the modified forward zone, which denotes forward faults when (3.12) is used, is illustrated in figure 3.8.
\[ (3.12) \quad D_{\text{mim}} = \frac{E_1}{(I_{\text{mim}1} - I_{\text{mim}1\text{pre}})} \]

\( D_{\text{mim}} \) = directionality when mimic currents are provided as inputs
\( E_1 \) = positive-sequence faulted voltage phasor at relay location
\( I_{\text{mim}1} \) = positive-sequence faulted mimic current phasor at relay location
\( I_{\text{mim}1\text{pre}} \) = positive-sequence faulted mimic pre-fault current phasor at relay

The mathematical description of the modified forward zone, illustrated in figure 3.8 is:
(Mimic) Modified Forward Zone: \(-90^\circ \leq \text{Angle}(D) \leq 90^\circ \)

\( \text{Angle}(D) \) = directionality angle (degrees)

It is apparent from the preceding conditions that if \((3.12)\) is used, which is the case for the advanced distance relay, the direction of the fault is determined by the following conditions. If the directionality angle, \( \mathcal{D} \), is determined to be in quadrant 1 or quadrant 4 of the R-X diagram, the fault is considered a forward fault. If the directionality angle is found to be in quadrant 2 or quadrant 3, the fault is considered a reverse fault.

At times, a three-phase fault may occur extremely close to the relay location. When this happens, the positive-sequence voltage phasor that the relay uses to determine fault directionality may be too small to be of use. In this case, the positive-sequence "memory voltage" is used for the directionality determination. Memory voltage is simply another name for pre-fault voltage. The memory voltage is used any time that the positive-sequence voltage at the relay location falls below 5% of its nominal value.

The directionality equations are evaluated based upon positive-sequence phasor values. In the advanced distance relay, before the phasors may be decomposed into their symmetrical components (to obtain the positive-sequence component), they must first be calculated via the 1/2 cycle phasor calculation algorithm. The significance of this information is that since the phasor values may not be valid until just over 1/2 of a cycle after a transient occurs, the directionality algorithm cannot be expected to provide a truly accurate result until the phasors do become valid.

**The Inverse CCVT Algorithm**

The CCVT, or Coupling Capacitor Voltage Transformer, is quite often used to provide power system voltage measurements for the purpose of relaying. The widespread use of the CCVT is due largely to the fact that it is more cost-efficient to employ than an appropriately sized PT, or Potential Transformer. The monetary savings come at the expense of transient performance, however.
The CCVT, illustrated in figure 3.9, consists of a transformer connected through a "tuning inductance" to a capacitive voltage divider. The capacitive divider is then connected to a power system bus. An advantage of this configuration is that the transformer required for the CCVT may be rated for a much lower voltage than the PT that would be required to be connected directly to the bus. It is the use of the lower rated voltage transformer that results in the considerable monetary savings.

The resonant circuit formed by the tuning inductance and the capacitive divider causes serious problems during transient conditions. During a transient, the CCVT output ceases to be an accurate representation of the voltage waveform that exists on the power system. This erroneous CCVT output waveform is known as a subsidence transient, as described in chapter 2 (section 2.4). Since the CCVT output serves as the input voltage of a relay, the relay cannot be expected to function correctly for the duration of the subsidence transient. Therefore, it is important to somehow eliminate the subsidence transient.

The subsidence transient may be eliminated, provided that the CCVT network representation is available. The method suggested by A. G. Phadke and R. B. Sollero consists of reconstructing the CCVT input voltage from the CCVT output voltage [8]. The CCVT input "sees" the actual power system voltage even during the transient conditions that may produce subsidence transients on the CCVT output. Therefore, by having the relay utilize a properly scaled version of the reconstructed CCVT input, the problems associated with the subsidence transient may be eliminated.

The "inverse CCVT" algorithm proposed for use with the advanced distance relay is based upon the method originally employed by Phadke and Sollero. The difference between the present method and the original method is the use of the trapezoidal method of integration to facilitate real-time computation. The overview of the inverse CCVT algorithm that follows is taken directly from a derivation provided the author by Dr. Phadke.
The inverse CCVT algorithm is based upon the discrete-time solution of the node admittance matrix of the CCVT network via the trapezoidal method of integration. The matrix equation that results is:

\[(3.13) \quad GE = I - h\]

\(G = \text{CCVT real admittance matrix containing Dommel conductances [8]}\)
\(E = \text{CCVT network node voltage vector}\)
\(I = \text{CCVT network node injection current vector}\)
\(h = \text{"history" vector containing node injections from reactive elements due to their state at the previous time step [8]}\)

The node injection current vector, \(I\), has a non-zero component only when the corresponding node is connected to a source. In the case of the CCVT network, the only node that is connected to a source is the CCVT input node, which is denoted node 1. This suggests the following partition of (3.13).

\[
\begin{pmatrix}
G_{11} & G_{12} \\
G_{21} & G_{22}
\end{pmatrix}
\begin{pmatrix}
E_1 \\
E_2
\end{pmatrix}
=
\begin{pmatrix}
I_1 \\
0
\end{pmatrix}
\begin{pmatrix}
h_1 \\
h_2
\end{pmatrix}
\]

Assuming that \(G \in \mathbb{R}^{n \times n}\) (i.e. \(G\) is an 'n' by 'n' real matrix), the dimension of the elements of the partitioned matrix are defined as follows.
\(G_{11} \in \mathbb{R}^{1 \times 1}\)
\(G_{12} \in \mathbb{R}^{1 \times (n-1)}\)
\(G_{21} \in \mathbb{R}^{(n-1) \times 1}\)
\(G_{22} \in \mathbb{R}^{(n-1) \times (n-1)}\)
\(E_1 \in \mathbb{R}^{1 \times 1}\)
\(h_1 \in \mathbb{R}^{1 \times 1}\)
\(E_2 \in \mathbb{R}^{(n-1) \times 1}\)
\(h_2 \in \mathbb{R}^{(n-1) \times 1}\)
\(I_1 \in \mathbb{R}^{1 \times 1}\)

The equation that results from the evaluation of the lower partition of the matrix is:

\(G_{21}E_1 + G_{22}E_2 = -h_2\)

Rearranging this equation yields:

\[(3.14) \quad E_2 = G_{22}^{-1}[ -G_{21}E_1 - h_2] = -G_{22}^{-1}G_{21}E_1 - G_{22}^{-1}h_2\]

\('E_2'\) is the vector that contains the voltages of the nodes that are not connected to sources, including the output node of the CCVT, which will be referred to as \('E_s'\). Since the voltage at \('E_s'\) is the voltage to which the relay has access, but the voltage at \('E_1'\) is the voltage that is desired, the previous equation is rearranged to yield \('E_1'\) as a function of \('E_s'\). Using this relation, which is derived below, the relay can reconstruct the true CCVT input voltage waveform from the CCVT output voltage waveform at \('E_s'\), even in the presence of subsidence transients.
To obtain \(E_1\) as a function of \(E_S\), (3.14) is first rearranged as follows:

\[
E_2 = -G_{22}^{-1}G_{21}E_1 - G_{22}^{-1}h_2
\]

\[
-G_{22}^{-1}G_{21}E_1 = E_2 + G_{22}^{-1}h_2
\]

Let \(d = -G_{22}^{-1}G_{21} \in \mathbb{R}^{(n-1) \times 1}\)

\[
d \cdot E_1 = E_2 + G_{22}^{-1}h_2
\]

Assuming that \(E_S\), the CCVT output voltage, is the \(k^{th}\) element of the vector \(E_2\), and that the notation \(\{E_2\}_k\) specifies the \(k^{th}\) element of the vector \(E_2\), then it is evident that:

\[
\{d \cdot E_1\}_k = \{E_2\}_k + \{G_{22}^{-1}h_2\}_k
\]

\[
\{d\}_k \cdot E_1 = \{E_2\}_k + \{G_{22}^{-1}h_2\}_k
\]

Let \(d_k = \{d\}_k = \{-G_{22}^{-1}G_{21}\}_k\)

Since \(E_s = \{E_1\}_k\), substitution yields the following scalar equation

\[
d_k \cdot E_1 = E_s + \{G_{22}^{-1}h_2\}_k
\]

Solving the scalar equation for \(E_1\) as a function of \(E_S\) results in:

(3.15) \[
E_1 = \frac{E_s + \{G_{22}^{-1}h_2\}_k}{d_k}
\]

Once (3.15) has been used to find the CCVT input voltage as a function of the CCVT output voltage, (3.14) may be used to find the voltage, and subsequently the current, at each other node in the CCVT network. With knowledge of the branch currents, the history vector is recalculated and stored [8]. Upon receipt of the next CCVT output sample, the process is begun again.

Time constraints prevented implementation of the inverse CCVT algorithm in the present version of the advanced distance relay program. Since subsidence transients degrade high-speed relay performance, and since the inverse CCVT algorithm had not yet been implemented, the CCVTs were replaced with PTs during the simulated relay testing. This, in effect, removed the problem of CCVT subsidence transients from the consideration of this thesis. However, due to the abundance and popularity of CCVTs, future work on the advanced distance relay should include implementation of the inverse CCVT algorithm.
CHAPTER 4

ADVANCED DISTANCE RELAY PERFORMANCE TESTS

4.1 Advanced Distance Relay Sample Test Cases

This section describes sample test cases which illustrate the operation of the advanced distance relay. Following the test case description, the test results are presented and explained.

There are four test cases that are provided in order to illustrate the operation of the advanced distance relay. All four test cases are based on the simplified power system in figure 4.1. In each case, 'EMTP' simulation produces the sampled waveforms upon which the advanced distance relay operates. The specified fault occurs 1 cycle into the simulation at the midpoint of the transmission line, and stays on the system for the remainder of the simulation duration, which is an additional 2.5 cycles. The positive-sequence and negative-sequence impedance of the protected transmission line is $1.638 + j5.700$ Ohms, while the zero-sequence impedance is $1.764 + j9.659$ Ohms.

![Diagram](image)

**Figure 4.1** Simplified power system simulated in sample test cases '1A', '1B', '2A', and '2B'.

**Test Case '1A'**

Test case '1A' consists of the power system illustrated in figure 4.1 with switches 'SW1' and 'SW2' open. The open switches ensure that the mutually coupled parallel transmission line 'L2', shown in the figure, is not included in this test case. Switch 'FSW1', placed at the midpoint of line 'L1' (0.5 per-unit length), simulates the occurrence of a fault by closing
after 1 cycle. The fault-path resistance, \( R_f \), is equal to 0.001 Ohms in order to simulate a zero resistance fault. The conditions of test case '1A' are summarized as follows:

- Transmission line 'L1': Enabled
- Mutually coupled parallel line 'L2': Disabled
- Fault location: 0.5 per-unit (line midpoint)
- Fault-path resistance \( R_f \): 0.001 Ohms

**Test Case '1B'**

Test case '1B' is identical to test case '1A' except in one regard. The fault-path resistance, \( R_f \), is 25 Ohms in case '1B'. Therefore, the conditions of test case '1B' are summarized as follows:

- Transmission line 'L1': Enabled
- Mutually coupled parallel line 'L2': Disabled
- Fault location: 0.5 per-unit (line midpoint)
- Fault-path resistance \( R_f \): 25.0 Ohms

**Test Case '2A'**

Test case '2A' is provided to illustrate the effect of the inclusion of a mutually coupled parallel line. Therefore, switches 'SW1' and 'SW2' are both closed so that the mutually coupled parallel line is incorporated into the simulation. As in the previous test cases, the fault is simulated at the midpoint of line 'L1' via the closing of switch 'FSW1' after 1 simulated cycle. The fault-path resistance, \( R_f \), is 0.001 Ohms. Test case '2A' is summarized as follows:

- Transmission line 'L1': Enabled
- Mutually coupled parallel line 'L2': Enabled
- Fault location: 0.5 per-unit (line midpoint)
- Fault-path resistance \( R_f \): 0.001 Ohms

**Test Case '2B'**

Test case '2B' is identical to test case '2A', with the exception of the fault-path resistance.
The fault-path resistance, \( R_f \), is 25 Ohms in test case '2B'. A summary of the test case follows.

<table>
<thead>
<tr>
<th>Transmission line 'L1':</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutually coupled parallel line 'L2':</td>
<td>Enabled</td>
</tr>
<tr>
<td>Fault location:</td>
<td>0.5 per-unit (line midpoint)</td>
</tr>
<tr>
<td>Fault-path resistance ( R_f ):</td>
<td>25.0 Ohms</td>
</tr>
</tbody>
</table>

Two simulations are performed for each of the previous test cases. The first simulation models a phase-to-ground fault on phase 'a', as illustrated in figure 4.1. The second simulation models a phase-to-phase fault. This fault happens at the same fault location identified in figure 4.1, but it occurs between phases 'a' and 'b' through resistance \( R_f \), instead of between phase 'a' and ground, as shown. The reason both types of fault are simulated for each test case is so that the performance of each version of the fault locator equation, (3.1) and (3.2), may be demonstrated.

These test cases are by no means considered exhaustive -- they are merely a tool with which to illustrate the operation of the advanced distance relay. The results of comprehensive testing that was performed during the development phase of the advanced distance relay are described in a subsequent section.

### 4.2 Sample Test Case Results

This section discusses the results obtained from the application of the advanced distance relay to the four test cases described in the preceding section. As stated, both a phase-to-ground fault and a phase-to-phase fault are simulated for each test case. For the cases in which the parallel line is in service, the per-unit fault locator equation is applied without knowledge of the quantities associated with the parallel line. For a review of the implications of the lack of knowledge of the parallel line quantities, see chapter 3, section 3.1

**Test Case '1A' (Phase-To-Ground)**

To illustrate the performance of the algorithms discussed in chapter 3, the input current, mimic current, current phasor magnitude, and transient monitor status are plotted in figure 4.2. The plot begins with post-fault sample number '1', which corresponds to the first sample after the fault is detected by the transient monitor, and continues through post-fault sample number '48'. The 48 samples correspond to exactly 2 cycles of post-fault data.
Upon inspecting the figure, note that the input current has a slight DC offset, while the mimic current has no such offset. Also notice the phase difference between the input current and the mimic current. As discussed in chapter 3, the mimic current leads the input current by the mimic angle, which is equal to the positive-sequence impedance angle of the protected transmission line. Recall, however, that after the corresponding phasor is calculated from the mimic current, the mimic angle is subtracted from the calculated phasor angle so that the phasor represents the input current -- not the mimic current. The magnitude of the current phasor that corresponds to the input current is included in the figure.

In the advanced distance relay, the transient monitor status is indicated by a Boolean '1' or '0', denoting a "large" value of 'T' or a "small" value of 'T', respectively. In the test case result figures, however, the transient monitor status indicator is scaled so as to be more discernible. Notice that while the current phasor magnitude is in a state of transition from the pre-fault value to the post-fault value, the transient monitor status is "high". Once the current phasor accurately represents the post-fault current, the transient monitor status becomes "low". This process takes approximately 1/2 of a cycle since the phasor calculation algorithm employs a 1/2 cycle DFT, as explained in chapter 3.
Figure 4.3  Test case '1A' (phase-to-ground): calculated per-unit fault distance.

Figure 4.3 includes the calculated per-unit fault distance along with the scaled transient monitor status. The post-fault samples of figure 4.3 correspond to the post-fault samples of figure 4.2. The transient monitor status is included in both figures to make this correlation apparent. Notice that the per-unit fault distance has nearly reached a steady-state value at the instant that the transient monitor status becomes "low". This is due to the fact that the phasors utilized by the per-unit fault locator equation become an approximately accurate representation of the input waveforms at the instant that the transient monitor status indicator falls. This behavior is exemplified by the current phasor magnitude plot of figure 4.2, as previously indicated.

While figure 4.3 is a good illustration of the behavior of the calculated per-unit fault distance, the scale makes it difficult to determine the final value. This problem is rectified in figure 4.4. The post-fault samples of figure 4.4 correspond to those of figures 4.2 and 4.3, but the scale is changed so that the behavior of the calculated per-unit fault distance near its steady-state value is more readily apparent. Recall that the fault occurs at 0.5 per-unit of the transmission line. Note the correlation between the actual fault location, and the per-unit fault distance computed by the advanced distance relay.
Test Case '1A' (Phase-To-Phase)

The same three types of figures used to illustrated the performance of the advanced distance relay for the phase-to-ground fault simulation of test case '1A' are utilized by each of the other test case simulations as well. Just as in the previous discussion, 2 cycles of post-fault data are presented, beginning with the first sample after the fault is detected by the transient monitor.

Figure 4.5  Test case '1A' (phase-to-phase): internal relay quantities.

Figure 4.5 is very similar to figure 4.2 of the phase-to-ground simulation discussion. The same observations hold for both figures.

Moreover, figures 4.6 and 4.7 are quite similar to figures 4.3 and 4.4 of the phase-to-ground simulation discussion. Therefore, the same observations hold, respectively.
**Test Case '1B'**

Due to the similarities between the figures that have been discussed and the figures that have yet to be presented, the performance figures for the phase-to-ground fault simulation and phase-to-phase fault simulation of test case '1B' may be found in appendix 'B'. The following discussion highlights the only significant differences between the results of test case '1A' and test case '1B'.

There are three important points to note from the results of the simulations of test case '1B'. The first point is that the final steady-state value of the calculated per-unit fault distance is slightly greater than the true per-unit fault distance (see figures B.3 and B.6). The second point is that the per-unit fault distance in the phase-to-ground simulation of test case '1B', figure B.3, possesses a small oscillation around its steady-state value.
Finally, the third point is that it takes the calculated per-unit fault distance several more samples to reach a steady-state value than in case '1A' (see figures B.2 and B.5). All of these observations are the result of the magnitude of fault-path resistance (25 Ohms) present. However, even in light of these observations, the relay performance is quite good.

Test Case '2A'

As with test case '1B', the performance figures for the phase-to-ground fault simulation and phase-to-phase fault simulation of test case '2A' may be found in appendix 'B'. There are no significant differences between the results of test cases '1A' and '2A'.

Note that although no discernible variation is introduced by the mutually coupled parallel line in these phase-to-ground simulation results, this is seldom the case. Typically, as confirmed by separate, controlled experiments, mutual coupling introduces a slightly more noticeable deviation in the phase-to-ground version of the per-unit fault distance calculation as compared to the true per-unit fault distance. It is possible that the symmetry of the faulted system for the phase-to-ground fault simulation prevents the mutually coupled parallel line from having a profound effect upon the distance calculation. It is important to recall, however, that the phase-to-phase version of the fault locator equation is inherently unaffected by the presence of a mutually coupled parallel line. For more detailed information, see the chapter 3, section 3.1.

Test Case '2B'

Along with the other test case performance figures, the figures for the results of test case '2B' may also be found in appendix 'B'. Upon reviewing these figures, there are a few points worth noting.

As in the results of test case '2A', the mutual coupling does not appear to play a significant role. The reason for this is very likely the same as the reason presented in the discussion of the results of case '2A'.

There is, however, a noticeable difference between the steady-state value around which the calculated per-unit fault distance oscillates and the true per-unit fault distance. As cited in the discussion of the results of test case '1B', this difference is due to the magnitude of the fault-path resistance. While the computed fault distance is not totally accurate, it is very close in light of the magnitude of the associated fault-path resistance (25 Ohms). This is a great improvement over the error that would result from the utilization of the traditional distance calculation, discussed in chapter 2. In the case that the traditional distance calculation were used, the associated error would be of the same magnitude as the fault-path resistance.
Summary of Sample Test Case Results

The following are general observations based on the results of the previous test cases. These observations summarize the prior discussions of test case results.

Low fault-path resistance faults:
- The calculated per-unit fault distance is extremely accurate.
- The calculated per-unit fault distance reaches its steady-state value in approximately 1/2 of a cycle.

High fault-path resistance faults:
- The calculated per-unit fault distance is relatively close, but not exactly equal, to the true value.
- The calculated per-unit fault distance requires slightly more than 1/2 of a cycle to reach its steady-state value.
- The calculated per-unit fault distance possesses a slight oscillation about its steady-state value.

These observations suggest that the advanced distance relay performs exceptionally well for the sample test cases with low fault-path resistance. Although the performance is somewhat degraded for the test cases with high fault-path resistance, it is still quite good. By way of comparison, it is far better than the performance that could be expected from the traditional distance equations.

4.3 Comprehensive Developmental Testing

The simplified test cases presented previously are included solely to illustrate the performance of the advanced distance relay. They are not intended to provide extensive verification of the relay's performance. This verification is provided, however, by the more comprehensive tests described in this section.

The advanced distance relay was subjected to extensive testing during its development phase. To facilitate this testing, a 3 generator, 4 transmission line, power system featuring transformers, capacitor banks, mutual coupling, and loads was modeled in "EMTP". Then, 66 different fault scenarios were run to provide input data with which to test the advanced distance relay. See appendix 'C' for a chart summarizing these scenarios. The conclusions drawn from the results of these tests are presented below.
Before summarizing the results of the testing, an explanation of the manner in which the results are interpreted must be presented. Due to the inappropriate method in which the power system loads were represented in the "EMTP" program, the resulting distribution factor phase angle was incorrectly altered during high resistance faults so that it became quite different from the angle of zero that was assumed (refer to chapter 3, section 3.1). This introduced a distortion into the calculated per-unit fault distance that was solely an artifact of the method used to model the load. In this respect, had the relay been tested on a physical power system, the error would not have occurred.

The power system loads were incorrectly modeled with fixed impedances, which resulted in a distortion of the network distribution factor phase angle during high resistance faults. The loads should have been modeled as fixed current sources to more faithfully represent the true conditions of a physical power system. Had this been the case, the distribution factor phase angle (based on the 'X/R' ratio) would not have been incorrectly altered, and the per-unit fault locator equation would have produced more accurate results.

In order to evaluate the test results in light of the error introduced during high resistance faults by the inappropriate load model, it was decided to compare the calculated per-unit fault distance to the fault distance determined by the traditional distance equations. In this respect, the relative performance of the advanced distance relay could be gauged. The results of the testing follow.

**Summary of Comprehensive Test Results**

The results of the low resistance fault tests may be analyzed directly since the distortion of the distribution factor phase angle, due to the incorrect modeling of loads, occurs only during high resistance faults. The conclusions drawn from these low resistance tests are:

**Low fault-path resistance faults:**
- The calculated per-unit fault distance is extremely accurate.
- The calculated per-unit fault distance reaches its steady-state value in approximately 1/2 of a cycle.

As mentioned previously, the results of the high resistance fault tests must be analyzed in reference to the fault distance calculated by the traditional fault distance equations. Again, recall that this is necessary due to the incorrect modeling of the simulated loads. The conclusions drawn from the high resistance fault tests, referenced to the traditional fault distance equations, are:
High fault-path resistance faults:
- The calculated per-unit fault distance is always more accurate than the traditionally calculated fault distance.
- The calculated per-unit fault distance requires slightly more than 1/2 of a cycle to reach its steady-state value.
- The calculated per-unit fault distance possesses a slight oscillation about its steady-state value.
CHAPTER 5

CONCLUSIONS

5.1 Discussion

This document has explained the theory of operation of a next-generation, computer based distance relay. This advanced distance relay possesses many advantages over conventional distance relays which utilize traditional fault distance estimation methods. The use of a fault locator that is nearly immune to the effects of fault-path resistance gives the advanced distance relay a significant advantage. Also, the relay's ability to locate a fault in just over 1/2 of a cycle is extremely beneficial. In addition, the use of algorithms that make the relay resistant to the effects of DC current offset and CCVT subsidence transients yield a high degree of reliability, even during extreme system conditions. The result is a remarkably reliable, yet exceedingly secure high-performance distance relay.

5.2 Future Work

While the algorithms have been fully developed, as described in this document, there is much room for continued work on the actual prototype program. Two suggested areas for additional work are described below.

One possible area for continued work would be in the implementation and testing of the inverse CCVT routine that was described in chapter 3, section 3.3. Although some rudimentary testing was completed, there was not time to implement the routine in the final "C" language prototype.

A second possible area for continued work would be the inclusion of a protective scheme, possibly similar to stepped distance relaying, within the prototype program. The scheme could include multiple zones of protection, with the fastest zone operating 1/2 of a cycle following fault detection.
APPENDIX A

ADVANCED DISTANCE RELAY SOURCE CODE
/* Program: Computer Based Distance Relay */
/* Encoded By: David Smith */
/* Language: 'C' */

#include <math.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>

#define theta(pi/12.0)    /* Sampling Interval in Radians */
#define cos_phi cos(phi * (pi / 180))
#define sin_phi sin(phi * (pi / 180))
#define data_mvmove memmove((&data[1].data,(sizeof(data)-sizeof(record))))
#define mmvmove(r1,r2) memmove(r1,r2,(sizeof(mimic_i)-sizeof(mimic_i[1])))
#define vmvmove(r1,r2) memmove(r1,r2,(sizeof(vwin)+sizeof(vwin[1])))
#define cmvmove(r1,r2) memmove(r1,r2,(sizeof(cmcre)+sizeof(cmcre[1])))

struct record
{
    long int step;
    double time;
    double current[4];
    double voltage[7];
};

struct cmplxnum
{
    double r;
    double i;
};

struct memrec
{
    struct cmplxnum vphas[7];
    struct cmplxnum cphas[4];
    int tmonstal[4];
};

const int maxchrline = 150;
const double pi = 3.14159265359;

const float drift_correction = 1.878; /* Correction in degrees/cycle */
const float crisratio = 300.0; /* CT Ratio Format = n:1 */
const float vtratio = 2000.0; /* VT Ratio Format = n:1 (n=1*)
const float nomvoltToN = 115.0/sqrt(3.0); /* CCVT Output Voltage */
/* (Line-to-Neutral) */
/* Impedances Are Actual !Line Imp */
const cmplxnum zline0 = {1.770, 6.96}; /* 0 Seq Line Impedance (rec. form) */
const cmplxnum zline1 = {0.868, 4.09}; /* 0 Seq Line Impedance (rec. form) */
const cmplxnum zline2 = {0.868, 4.09}; /* 0 Seq Line Impedance (rec. form) */
/* + Seq Imped Angle for System(deg) */
const double phi = atan(zline1.i/zline1.r)*(180.0/pi);

char name[] = "input.in"; /* Input File Name */
char extname[] = "output.out"; /* Test Output File Name */

struct record data[3]; /* Most Recent data @ data[0] */
struct cmplxnum cphas[4]; /* Current Phasors */
struct cmplxnum vphas[7]; /* Voltage Phasors */
struct cmplxnum cmcre[25]; /* 1 Cycle of Memory */

struct cmplxnum zline0; /* Holds Secondary Reflected Impedance */
struct cmplxnum zline1; /* Holds Secondary Reflected Impedance */
struct cmplxnum zline2; /* Holds Secondary Reflected Impedance */
char line[maxchrline]; /* T-son up/down status: 1=up 0=down */
double mimic_i[13][4]; /* Storage Window for Mimic Output */
/* Most Recent data @ Mimic_i[12] */
double vwin[13][7]; /*Storage Window for Voltage Data*/
double 0:[13][4]; /*Most Recent data @ vwin[12]*/
FILE *filepointer; /*Input File Pointer*/
float fdistance = 0.0; /*Distance of nearest data @ r[12]*/
cmplxnum Badis, Ia, compdis, a, afdis, approxdis, prefl; /*Test Var*/

#include <complex.h>

/*CompAdd rec complex numbers*/
cmplxnum compadd_rec(cmplexnum a, cmplexnum b)
{
    cmplxnum result;
    result.r = a.r + b.r;
    result.i = a.i + b.i;
    return result;
}

cmplxnum compresult_rec(cmplexnum a, cmplexnum b)
{
    cmplxnum result;
    result.r = (a.r * b.r) - (a.i * b.i);
    result.i = (a.r * b.i) + (a.i * b.r);
    return result;
}

cmplxnum compsub_rec(cmplexnum a, cmplexnum b) /* a - b where a & b are complex*/
{
    cmplxnum result;
    result.r = a.r - b.r;
    result.i = a.i - b.i;
    return result;
}

APPENDIX A  58
```
cmplxnum compdiv_rec(complexnum a, complexnum b) {
    complexnum apol, bpol, pol_result, rec_result;
    apol = rec_to_pol(a);
    bpol = rec_to_pol(b);
    pol_result.r = (apol.r / bpol.r);
    pol_result.i = (apol.i - bpol.i);
    pol_result.i = pol_result.i*(pi/180.0);
    rec_result.r = pol_result.r*cos(pol_result.i);
    rec_result.i = pol_result.r*sin(pol_result.i);
    return rec_result;
}

complexnum rec_to_pol(complexnum a) {
    complexnum result;
    result.r = mag(a.r, a.i);
    result.i = deg_ang(a.r, a.i);
    return result;
}

complexnum pol_to_rec(complexnum a) {
    complexnum result;
    double temp;
    result.r = a.r * cos(a.i*(pi/180.0));
    result.i = a.r * sin(a.i*(pi/180.0));
    return result;
}

double rad_ang(double real, double imag) {
    double temp;
    temp = deg_ang(real, imag);
    temp = temp * (pi/180.0);
    return temp;
}

double deg_ang(double real, double imag) {
    double preangle;
    if (real==0.0) real=0.000000000001;
    preangle = ((atan(fabs(imag/real)))*(180.0/pi));
    if ((real>=0.0)&&(imag>=0.0))
        preangle = 360.0 - preangle;
    if ((real>=0.0)&&(imag<0.0))
        preangle = 180.0 - preangle;
    if ((real<0.0)&&(imag<0.0))
        preangle = 180.0 + preangle;
    return preangle;
}
```

double mag(double real, double imag)
{
    return (sqrt(real*real) + (imag*imag));
}

void compute_phasors( void )
{
    double delx;
    char c;
    static char kplus1 = 0;

    if ( kplus1 == 24 ) kplus1 = 0;
    ++kplus1;
    for ( c = 1 ; c <= 3 ; ++c )
    {
        delx = (sqrt(2.0) / 12.0) * (mimic_i[12][c] + mimic_i[0][c]);
        cphsr[c].r = cphsr[c].r + delx*cos(kplus1*theta);
        cphsr[c].i = cphsr[c].i - delx*sin(kplus1*theta);
    }
    for ( c = 1 ; c <= 6 ; ++c )
    {
        delx = (sqrt(2.0) / 12.0) * (vwin[12][c] + vwin[0][c]);
        vphsr[c].r = vphsr[c].r + delx*cos(kplus1*theta);
        vphsr[c].i = vphsr[c].i - delx*sin(kplus1*theta);
    }
    return;
}

void compute_residuals( void )
{
    char c;
    double delx, r1_old;

    for ( c = 1 ; c <= 3 ; ++c )
    {
        delx = (mimic_i[12][c] + raimic_i[0][c])/6.0;
        r1_old = r[1][c];
        r[1][c] = r[2][c] + cos(theta)*delx;
        r[2][c] = r[3][c] + cos(2.0*theta)*delx;
        r[3][c] = r[4][c] + cos(3.0*theta)*delx;
        r[4][c] = r[5][c] + 0.5*delx;
        r[5][c] = r[6][c] + cos(5.0*theta)*delx;
        r[6][c] = r[7][c] + 0.0*delx;
        r[7][c] = r[8][c] + cos(7.0*theta)*delx;
        r[8][c] = r[9][c] + 0.5*delx;
        r[9][c] = r[10][c] + cos(9.0*theta)*delx;
        r[10][c] = r[11][c] + cos(10.0*theta)*delx;
        r[11][c] = r[12][c] + cos(11.0*theta)*delx;
        r[12][c] = -1.9*r1_old + 5.0*delx;
    }
    return;
}

void initialize_phasors( void )
{
    char c;
    double root2div12;
    double cos1 = cos(theta);
    double cos2 = cos(2*theta);
    double cos3 = cos(3*theta);
    double cos4 = 0.5;
    double cos5 = cos(5*theta);
    double sin1 = sin(theta);
    double sin2 = 0.5;

APPENDIX A
double sin3 = sin(3*theta);
double sin4 = sin(4*theta);
double sin5 = sin(5*theta);

root2div12 = sqrt(2.0) / 12.0;
for (c = 1; c <= 3; ++c)
{
cphar[c].r = root2div12*(cos1*mimic_i[1][c] + cos2*mimic_i[2][c]
    + cos3*mimic_i[3][c] + cos4*mimic_i[4][c]
    + cos5*mimic_i[5][c] + 0.0*mimic_i[6][c]
    - cos5*mimic_i[7][c] - cos4*mimic_i[8][c]
    - cos3*mimic_i[9][c] - cos2*mimic_i[10][c]
    - cos1*mimic_i[11][c] - 1.0*mimic_i[12][c]);

cphar[c].i = root2div12*(-sin1*mimic_i[1][c] - sin2*mimic_i[2][c]
    - sin3*mimic_i[3][c] - sin4*mimic_i[4][c]
    - sin5*mimic_i[5][c] - 1.0*mimic_i[6][c]
    - sin5*mimic_i[7][c] - sin4*mimic_i[8][c]
    - sin3*mimic_i[9][c] - sin2*mimic_i[10][c]
    - sin1*mimic_i[11][c] - 0.0*mimic_i[12][c]);
}

for (c = 1; c <= 6; ++c)
{
vphar[c].r = root2div12*(cos1*vwin[1][c] + cos2*vwin[2][c]
    + cos3*vwin[3][c] + cos4*vwin[4][c]
    + cos5*vwin[5][c] + 0.0*vwin[6][c]
    - cos5*vwin[7][c] - cos4*vwin[8][c]
    - cos3*vwin[9][c] - cos2*vwin[10][c]
    - cos1*vwin[11][c] - 1.0*vwin[12][c]);

evphar[c].i = root2div12*(-sin1*vwin[1][c] - sin2*vwin[2][c]
    - sin3*vwin[3][c] - sin4*vwin[4][c]
    - sin5*vwin[5][c] - 1.0*vwin[6][c]
    - sin5*vwin[7][c] - sin4*vwin[8][c]
    - sin3*vwin[9][c] - sin2*vwin[10][c]
    - sin1*vwin[11][c] - 0.0*vwin[12][c]);
}

return;

void initialize_residuals(void)
{
    char c;
    double cos1 = cos(theta);
    double cos2 = cos(2*theta);
    double cos3 = cos(3*theta);
    double cos4 = 0.5;
    double cos5 = cos(5*theta);

    for (c = 1; c <= 3; ++c)
    {
        r[1][c] = (1.0/6.0)*(5.0*mimic_i[1][c] - cos1*mimic_i[2][c]
            - cos2*mimic_i[3][c] - cos3*mimic_i[4][c]
            - cos4*mimic_i[5][c] - cos5*mimic_i[6][c]
            + 0.0*mimic_i[7][c] + cos5*mimic_i[8][c]
            + cos4*mimic_i[9][c] + cos3*mimic_i[10][c]
            + cos2*mimic_i[11][c] + cos1*mimic_i[12][c]);

        r[2][c] = (1.0/6.0)*(-cos1*mimic_i[1][c] + 5.0*mimic_i[2][c]
            - cos1*mimic_i[3][c] - cos2*mimic_i[4][c]
            - cos3*mimic_i[5][c] - cos4*mimic_i[6][c]
            - cos5*mimic_i[7][c] + 0.0*mimic_i[8][c]
            + cos5*mimic_i[9][c] + cos4*mimic_i[10][c]
            + cos3*mimic_i[11][c] + cos2*mimic_i[12][c]);
    }
}
\[ r[3][c] = (1.0/6.0) * ( \cos^2 \text{mimic}_i[1][c] - \cos^2 \text{mimic}_i[2][c] \\
+ 5.0 \text{mimic}_i[3][c] - \cos \text{mimic}_i[4][c] \\
- \cos^2 \text{mimic}_i[5][c] - \cos \text{mimic}_i[6][c] \\
- \cos^4 \text{mimic}_i[7][c] - \cos^5 \text{mimic}_i[8][c] \\
+ 0.0 \text{mimic}_i[9][c] + \cos^5 \text{mimic}_i[10][c] \\
+ \cos^4 \text{mimic}_i[11][c] + \cos^3 \text{mimic}_i[12][c];) \]

\[ r[4][c] = (1.0/6.0) * ( \cos^3 \text{mimic}_i[1][c] - \cos^2 \text{mimic}_i[2][c] \\
- \cos^2 \text{mimic}_i[3][c] - \cos^2 \text{mimic}_i[4][c] \\
- \cos^3 \text{mimic}_i[5][c] - \cos^2 \text{mimic}_i[6][c] \\
- \cos^3 \text{mimic}_i[7][c] - \cos^4 \text{mimic}_i[8][c] \\
- \cos^5 \text{mimic}_i[9][c] + 0.0 \text{mimic}_i[10][c] \\
+ \cos^5 \text{mimic}_i[11][c] + \cos^4 \text{mimic}_i[12][c];) \]

\[ r[5][c] = (1.0/6.0) * ( \cos^4 \text{mimic}_i[1][c] - \cos^3 \text{mimic}_i[2][c] \\
- \cos^2 \text{mimic}_i[3][c] - \cos^2 \text{mimic}_i[4][c] \\
+ 5.0 \text{mimic}_i[5][c] - \cos \text{mimic}_i[6][c] \\
- \cos^2 \text{mimic}_i[7][c] - \cos^2 \text{mimic}_i[8][c] \\
- \cos^4 \text{mimic}_i[9][c] - \cos^5 \text{mimic}_i[10][c] \\
+ 0.0 \text{mimic}_i[11][c] + \cos^5 \text{mimic}_i[12][c];) \]

\[ r[6][c] = (1.0/6.0) * ( \cos^5 \text{mimic}_i[1][c] - \cos^4 \text{mimic}_i[2][c] \\
- \cos^3 \text{mimic}_i[3][c] - \cos^2 \text{mimic}_i[4][c] \\
- \cos^2 \text{mimic}_i[5][c] - \cos^2 \text{mimic}_i[6][c] \\
+ 5.0 \text{mimic}_i[7][c] - \cos \text{mimic}_i[8][c] \\
- \cos^2 \text{mimic}_i[9][c] - \cos^5 \text{mimic}_i[10][c] \\
- \cos^4 \text{mimic}_i[11][c] + \cos^5 \text{mimic}_i[12][c];) \]

\[ r[7][c] = (1.0/6.0) * (0.0 \text{mimic}_i[1][c] - \cos^5 \text{mimic}_i[2][c] \\
- \cos^4 \text{mimic}_i[3][c] - \cos^3 \text{mimic}_i[4][c] \\
- \cos^2 \text{mimic}_i[5][c] - \cos^2 \text{mimic}_i[6][c] \\
+ 5.0 \text{mimic}_i[7][c] - \cos \text{mimic}_i[8][c] \\
- \cos^2 \text{mimic}_i[9][c] - \cos^5 \text{mimic}_i[10][c] \\
- \cos^4 \text{mimic}_i[11][c] - \cos^5 \text{mimic}_i[12][c];) \]

\[ r[8][c] = (1.0/6.0) * (\cos^5 \text{mimic}_i[1][c] + 0.0 \text{mimic}_i[2][c] \\
- \cos^5 \text{mimic}_i[3][c] - \cos^4 \text{mimic}_i[4][c] \\
- \cos^3 \text{mimic}_i[5][c] - \cos^2 \text{mimic}_i[6][c] \\
- \cos^2 \text{mimic}_i[7][c] - 5.0 \text{mimic}_i[8][c] \\
- \cos^2 \text{mimic}_i[9][c] - \cos^2 \text{mimic}_i[10][c] \\
- \cos^2 \text{mimic}_i[11][c] - \cos^4 \text{mimic}_i[12][c];) \]

\[ r[9][c] = (1.0/6.0) * (\cos^2 \text{mimic}_i[1][c] + \cos^5 \text{mimic}_i[2][c] \\
+ \cos^5 \text{mimic}_i[3][c] - \cos^5 \text{mimic}_i[4][c] \\
- \cos^4 \text{mimic}_i[5][c] - \cos^3 \text{mimic}_i[6][c] \\
- \cos^2 \text{mimic}_i[7][c] - \cos^5 \text{mimic}_i[8][c] \\
+ 5.0 \text{mimic}_i[9][c] + \cos^5 \text{mimic}_i[10][c] \\
- \cos^2 \text{mimic}_i[11][c] - \cos^3 \text{mimic}_i[12][c];) \]

\[ r[10][c] = (1.0/6.0) * (\cos^3 \text{mimic}_i[1][c] + \cos^4 \text{mimic}_i[2][c] \\
+ \cos^5 \text{mimic}_i[3][c] + 0.0 \text{mimic}_i[4][c] \\
- \cos^5 \text{mimic}_i[5][c] - \cos^4 \text{mimic}_i[6][c] \\
- \cos^3 \text{mimic}_i[7][c] - \cos^2 \text{mimic}_i[9][c] \\
- \cos^2 \text{mimic}_i[9][c] + 5.0 \text{mimic}_i[10][c] \\
- \cos^2 \text{mimic}_i[11][c] - \cos^2 \text{mimic}_i[12][c];) \]

\[ r[11][c] = (1.0/6.0) * (\cos^2 \text{mimic}_i[1][c] + \cos^3 \text{mimic}_i[2][c] \\
+ \cos^4 \text{mimic}_i[3][c] + \cos^3 \text{mimic}_i[4][c] \\
+ 0.0 \text{mimic}_i[5][c] - \cos^5 \text{mimic}_i[6][c] \\
- \cos^4 \text{mimic}_i[7][c] - \cos^3 \text{mimic}_i[8][c] \\
- \cos^2 \text{mimic}_i[9][c] - \cos^1 \text{mimic}_i[10][c] \\
+ 5.0 \text{mimic}_i[11][c] - \cos^1 \text{mimic}_i[12][c];) \]

APPENDIX A
r[12][c][1]=(1.0/6.0) cos(1*mimic_i[1][c] + cos2*mimic_i[2][c]
    + cos3*mimic_i[3][c] + cos4*mimic_i[4][c]
    + cos5*mimic_i[5][c] + 0.0*mimic_i[6][c]
    - cos5*mimic_i[7][c] - cos4*mimic_i[8][c]
    - cos3*mimic_i[9][c] - cos2*mimic_i[10][c]
    - cos1*mimic_i[11][c] - 5.0*mimic_i[12][c])**;

return;
}

void mimic_routine(struct record *d0, struct record *d1, struct record *d2)
{
    unsigned char c1;

    nimientove(mimic_i,&mimic_i[1]); /* Moves Floating Mimic Window */
    for (c1 = 1; c1 <= 3; ++c1) /* Mimic Eqn for Each Phase */
    {
        mimic_i[12][c1] = d1->current[c1] * cos_phi +
                        (theta/sin(theta)) * (1.0/(2.0/theta)) * sin_phi +
                        (d0->current[c1] - d2->current[c1]);
    }

    vWiz.Move(vwin,&vwin[1]); /* Fills Most Recent Element of */
    for (c1 = 1; c1 <= 3; ++c1) /* The Voltage Window -- Vwin */
    {
        vwin[12][c1] = d1->voltage[c1];
    }

    return;
}

void parse(char in[], struct record *d)
{
    long int step;
    double time;,
    double c[4];
    double v[7];

    if (!scanf(in,"%d%d%d%d%d%d%d",&step,&time,&c[1],&c[2],&c[3],
               &v[1],&v[2],&v[3],&v[4],&v[5],&v[6]) || 11)
    {
        printf("There was an error while parsing the input \n");
        exit(1);
    }

    d->step = step;
    d->time = time;
    d->current[1] = c[1];
    d->current[2] = c[2];
    d->current[3] = c[3];
    d->voltage[1] = v[1];
    d->voltage[2] = v[2];
    d->voltage[3] = v[3];
    d->voltage[4] = v[4];
    d->voltage[5] = v[5];
    d->voltage[6] = v[6];

    return;
}

void openfile(char name1[],char name2[])
{
    extern FILE *filepointer;
    int i;
    char header[maxchline];

APPENDIX A

63
if ((filepointer = fopen(nme,"r")) == NULL)
    { 
        printf("An error has occurred while attempting to open the file in\n");
        exit(1);
    }

fout = fopen(nme2,"w");                              /*Open Test Output File*/
for (i = 1 ; i <= 0 ; ++i)                              /*Read & Skip File Header*/
    { 
           
            gets(header,maxchsize,filepointer);
        if (feof(filepointer) || error(filepointer))
            { 
                    printf("There has been an error while reading the file header\n");
                    exit(1);
            }
        } 

        return;
    }

void sym_comp(complex Qa,complex Qb,complex Qc,complex *Qa0,complex *Qa1,complex *Qa2)
    { 
        /*Symmetrical Components of the Phasor Passed to Qa are */
        /*Returned by Qa0 (zero sequence) Qa1 (pos. sequence), */
        /*And Qa2 (neg. sequence)*/
        complex temp1,temp2;
        complex op = {0.5,0.8660254};                /*Rotation Operator (120 deg)*/
        complex opxop = {0.5,0.8660254};            /*Rotation Operator squared*/
        Qa0->r = (Qa.r + Qb.r + Qc.r)/3.0;
        Qa0->i = (Qa.i + Qb.i + Qc.i)/3.0;
        temp1 = compmult_rec(op,Qb);
        temp2 = compmult_rec(opxop,Qc);
        Qa1->r = (Qa.r + temp1.r + temp2.r)/3.0;
        Qa1->i = (Qa.i + temp1.i + temp2.i)/3.0;
        temp1 = compmult_rec(op,Qc);
        temp2 = compmult_rec(opxop,Qb);
        Qa2->r = (Qa.r + temp2.r + temp1.r)/3.0;
        Qa2->i = (Qa.i + temp2.i + temp1.i)/3.0;
        return;
    }

void cyclemem(vid)
    { 
        char phase;
        complex null = {0,0,0,0};
        for (phase = 1 ; phase <= 3 ; ++phase)
            { 
            cyclemem[0].vphr[phase].r = vphr[phase].r;
            cyclemem[0].vphr[phase].i = vphr[phase].i;
            cyclemem[0].cphr[phase].r = cphr[phase].r;
            cyclemem[0].cphr[phase].i = cphr[phase].i;
            cyclemem[0].tmonstat[phase] = tmonstat[phase];
            }
    }
for (phase = 4; phase <= 6; ++phase)
{
    cycmem[0].vphsr[phase].r = vphsr[phase].r;
    cycmem[0].vphsr[phase].i = vphsr[phase].i;
}

cycmemmove(&cycmem[1], cycmem);

for (phase = 1; phase <= 3; ++phase)
{
    cycmem[0].vphsr[phase] = null;
    cycmem[0].cphsr[phase] = null;
    cycmem[0].tmomat[phase] = 0;
}

for (phase = 4; phase <= 6; ++phase)
    cycmem[0].vphsr[phase] = null;

return;
}

cmploxnum faultdirection(cmploxnum prela0, cmploxnum prela1, cmploxnum prela2, cmploxnum preVa1)
{
    cmploxnum               dirV, deltaI;
    cmploxnum               postVa0, postVa1, postVa2;
    cmploxnum               postla0, postla1, postle2;
    cmploxnum               temp;

    sym_comp(cycmem[1].vphsr[1], cycmem[1].vphsr[2], cycmem[1].vphsr[3],
             &postVa0, &postVa1, &postVa2);
    sym_comp(cycmem[1].cphsr[1], cycmem[1].cphsr[2], cycmem[1].cphsr[3],
             &postla0, &postla1, &postle2);

    dirV = postVa1;
    Eadis = postVa1;

    /*Memory Voltage Statement*/
    if (mag(dirV, dirV, i) < ((0.05*nomvoltLtoN)) dirV = preVa1;

    deltaI = compsh_rec(postla1, prela1);
    return (compdiv_rec(dirV, deltaI));
}

void ifault_then_direction(void)
{
    /*JUST IMPLEMENTED FOR PHASE A CURRENTLY*/

    static int                 numsamplepoints = 0;
    const char                 phase < 1; /*Phase AV*/
    static int                 samples = 0;
    static char                faultflag = 0;
    static char                dirdscaleflag = 0;
    static cmploxnum           prela0, prela1, prela2, prefaultV;
    static cmploxnum           preVa0, preVa1, preVa2, prefaultV;
    static char                 times = 0;

    if (((cycmem[2].tmomat[phase] == 0) && (tmomat[phase] == 1))
    {
        times++;
        faultflag = 1;
        samples = 0;
        /*dirdscaleflag = 0;*/
    }
}
if ((cymem[2].monstat[phase] == 1) && (monstat[phase] == 0))
{
    faultflag = diridiscalcfalg = 0;
    direction_r = 0.0;
    direction_i = 0.0;
    fdistance = 0.0;
} /*
if (faultflag == 1) ++samples;
if (((faultflag == 1) && (times <= 2)) || diridiscalcfalg == 1;
if (diridiscalcfalg) numsamplepoints;

if (faultflag && diridiscalcfalg)
{
    /* Get Prefault Voltage and Current (1/2 Cycle Before fault)*/
    sym_comp(cymem[12].cphsr[1],cymem[12].cphsr[2],cymem[12].cphsr[3],
        &prela0,&prela1,&prela2);
    prefaultI = cymem[12].cphsr[phase];

    prefaultI = rec_to_pol(prefaultI);
    prefaultI_i = prefaultI_i * phi - (drift_correction/2);
    prefaultI = pol_to_rec(prefaultI);

    sym_comp(cymem[12].vphsr[1],cymem[12].vphsr[2],cymem[12].vphsr[3],
        &preVa0,&preVa1,&preVa2);
    prefaultV = cymem[12].vphsr[phase];
}

if (((diridiscalcfalg == 1) && (times <= 2)) && (numsamplepoints <= 64))
{
    /*CALCULATE DIRECTION OF FAULT*/
    direction = faultdirection(prela0,prela1,prela2,preVa1);
    direction = rec_to_pol(direction);
    /*CALCULATE FAULT DISTANCE*/
    fdistance = distance(phase,&prefaultI);
}

return;

double distance(char phase, cmplxnum *nonminpreflt)
{
    char count;
    double num, denom, dist;
    cmplxnum lcomp, lcompa0, lcompa1, lcompa2, deltI;
    cmplxnum m;

    cmplxnum nonmincphsr[4]; /*Current Phasors w/ Mism Angle Removed*/

    m = compsub_rec(rzline0, rzline1);
    m = compdiv_rec(m, rzline1);

    for (count = 1; count <= 3; ++count)
    {
        nonmincphsr[count] = rec_to_pol(cphsr[count]);
        nonmincphsr[count] = nonmincphsr[count] * phi; /*Subtract Mism Angle*/
        nonmincphsr[count] = pol_to_rec(nonmincphsr[count]);
    }

    sym_comp(nonmincphsr[1], nonmincphsr[2], nonmincphsr[3], &is0, &i1a, &is2);
    lcomp = compadd_rec(nonmincphsr[phase], compmult_rec(m, lcompa0));
    approxdis = compdiv_rec(Eadis, lcomp);

    /*Note: Eqn for phase-to-ground faults only*/
    deltI = compsub_rec(nonmincphsr[phase], *nonminpreflt);

    num = (vphsr[phase] * deltI - (vphsr[phase] * deltI));
    denom = (rzline1 * (lcomp * deltI + lcomp * deltI) - (rzline1 * (lcomp * deltI + lcomp * deltI)));
dist = (num / denom);

fprint(fout,"%d%e%e%e\n",tmonstat[1],dist,approxdisr,approxdisi);

*nonmimprefilt = rec_to_pol(*nonmimprefilt);
nonmimprefilt->i = nonmimprefilt->i - (drift_correction/24);
*nonmimprefilt = pol_to_rec(*nonmimprefilt);
return dist;
}

void readdata(FILE *fp)
{
    char         count;
    static char  initial = 0; /*Initialization Flag*/

    for (count = 1; count <= 2; ++count) /*Get 1st 2 Lines of Input*/
    {
        fgets(line,maxchrline,fp);
        if (feof(fp) || ferror(fp))
        {
            printf("There has been an error while reading the file.in");
            exit(1);
        }
        datawinmove;
        parse(line,&data[0]);
    }

    while (!feof(fp)) /*****Main Loop*****/
    {
        fgets(line,maxchrline,fp);
        if (feof(fp))
        {
            printf("There has been an error while reading the file.in");
            exit(1);
        }
        datawinmove;
        parse(line,&data[0]);
        mimic_routine(&data[0],&data[1],&data[2]);
        compute_residuals();
        compute_phasors();
        if ((initial == 0) && ((mimic_1[1][1] == 0.0)) /*Initialize Residual*/
        {
            initialize_residuals(); /*Matrix and Phasor */
            initialize_phasors(); /*Matrix After First */
            initial = 1; /*12 Mimize Currents */
        }
        transient_mon();
        cyclenm();
        if(fault_thendirection());
        return;
    }

void transient_mon(void)
{
    const float threshold = .50; /* Tmonsum value that enables*/
    const float factr = 4.0; /* Dynamic "tmonup/down" checking*/
    const float factr = 4.0; /* Dynamic "tmonup/down" checking*/
    static double tmonmin[4] = {0.0,threshold,threshold,threshold};
    static double tmonmax[4] = {0.0,0.0,0.0,0.0};
    char c;
    char phase;
    double tmonsum[4] = {0.0,0.0,0.0,0.0};

    for (phase = 1; phase <= 3; ++phase)
    {
        for (c = 1; c <= 12; ++c)
        { tmonsum[phase]+=fabs(r[c][phase]);
            if (tmonsum[phase]>tmonmax[phase]) tmonmax[phase] = tmonsum[phase];
            if (tmonsum[phase]<tmonmin[phase]) tmonmin[phase] = tmonsum[phase];
        }
    }
}
if ((tmonsum[phase] > threshold) && (tmonnum[phase] > (factr * tmonmin[phase])))
{
    tmonstat[phase] = 1;
    tmonnum[phase] = tmonsum[phase];
}
if (((tmonstat[phase] == 1) && (tmonsum[phase] < threshold)) || (tmonsum[phase] < (tmonmax[phase] * factr)))
{
    tmonstat[phase] = 0;
    tmonnum[phase] = tmonsum[phase];
}
}
return;
}

main()
{
    rzline0.r = zline0.r * (cratio/vratio);
    rzline0.i = zline0.i * (cratio/vratio);
    rzline1.r = zline1.r * (cratio/vratio);
    rzline1.i = zline1.i * (cratio/vratio);
    rzline2.r = zline2.r * (cratio/vratio);
    rzline2.i = zline2.i * (cratio/vratio);

    openfile(zame, outname);
    readdata(filepointer);
    return 0;
}
APPENDIX B

RESULTS OF SAMPLE TEST CASES
'1B', '2A', '2B'
Test Case '1B': Phase-To-Ground

Figure B.1  Test case '1B' (phase-to-ground): internal relay quantities.

Figure B.2  Test case '1B' (phase-to-ground): calculated per-unit fault distance.

Figure B.3  Test case '1B' (phase-to-ground): calculated per-unit fault distance.
Test Case '1B': Phase-To-Phase

Figure B.4  Test case '1B' (phase-to-phase): internal relay quantities.

Figure B.5  Test case '1B' (phase-to-phase): calculated per-unit fault distance.

Figure B.6  Test case '1B' (phase-to-phase): calculated per-unit fault distance.
Test Case '2A': Phase-To-Ground

**Figure B.7** Test case '2A' (phase-to-ground): internal relay quantities.

**Figure B.8** Test case '2A' (phase-to-ground): calculated per-unit fault distance.

**Figure B.9** Test case '2A' (phase-to-ground): calculated per-unit fault distance.
Test Case '2A': Phase-To-Phase

Figure B.10  Test case '2A' (phase-to-phase): internal relay quantities.

Figure B.11  Test case '2A' (phase-to-phase): calculated per-unit fault distance.

Figure B.12  Test case '2A' (phase-to-phase): calculated per-unit fault distance.
Test Case '2B': Phase-To-Ground

Figure B.13  Test case '2B' (phase-to-ground): internal relay quantities.

Figure B.14  Test case '2B' (phase-to-ground): calculated per-unit fault distance.

Figure B.15  Test case '2B' (phase-to-ground): calculated per-unit fault distance.
Test Case '2B': Phase-To-Phase

Figure B.16  Test case '2B' (phase-to-phase): internal relay quantities.

Figure B.17  Test case '2B' (phase-to-phase): calculated per-unit fault distance.

Figure B.18  Test case '2B' (phase-to-phase): calculated per-unit fault distance.
APPENDIX C

INDEX OF COMPREHENSIVE ADVANCED DISTANCE RELAY TEST CASES
<table>
<thead>
<tr>
<th>Case No.</th>
<th>L1 Fault Location (%)</th>
<th>L2 Fault Location (%)</th>
<th>L1,L2 Length (miles)</th>
<th>Fault Type</th>
<th>Fault Resistance (Ohms)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>--</td>
<td>5</td>
<td>A-GND</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>--</td>
<td>120</td>
<td>A-GND</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>--</td>
<td>5</td>
<td>A-GND</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>--</td>
<td>5</td>
<td>A-GND</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>85</td>
<td>--</td>
<td>5</td>
<td>A-GND</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>--</td>
<td>5</td>
<td>A-GND</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>--</td>
<td>5</td>
<td>A-GND</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>25</td>
<td>--</td>
<td>5</td>
<td>A-GND</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>--</td>
<td>0</td>
<td>5</td>
<td>3 PHASE</td>
<td>ra=rb=rc=0.2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>70</td>
<td>--</td>
<td>5</td>
<td>A-GND</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>--</td>
<td>100</td>
<td>5</td>
<td>3 PHASE</td>
<td>ra=rb=rc=0.2</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>--</td>
<td>100</td>
<td>5</td>
<td>A-B-GND</td>
<td>ra=rb=rg=0.2</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>100</td>
<td>--</td>
<td>5</td>
<td>3 PHASE</td>
<td>ra=rb=rc=0.2</td>
<td>Parallel Line Open</td>
</tr>
<tr>
<td>14</td>
<td>100</td>
<td>--</td>
<td>5</td>
<td>A-GND</td>
<td>0.2</td>
<td>Parallel Line Open</td>
</tr>
<tr>
<td>15</td>
<td>--</td>
<td>--</td>
<td>5</td>
<td>--</td>
<td>--</td>
<td>Energize Bus Capacitors</td>
</tr>
<tr>
<td>16</td>
<td>--</td>
<td>0</td>
<td>5</td>
<td>A-GND</td>
<td>0.4</td>
<td>Sudden Reversal</td>
</tr>
<tr>
<td>17</td>
<td>25</td>
<td>--</td>
<td>5</td>
<td>A-GND</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>Case No.</td>
<td>L1 Fault Location (%)</td>
<td>L2 Fault Location (%)</td>
<td>L1,L2 Length (miles)</td>
<td>Fault Type</td>
<td>Fault Resistance (Ohms)</td>
<td>Notes</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------</td>
<td>-----------------------</td>
<td>----------------------</td>
<td>------------</td>
<td>------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>18</td>
<td>85</td>
<td>--</td>
<td>5</td>
<td>A-GND</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>85</td>
<td>--</td>
<td>5</td>
<td>A-B-GND</td>
<td>ra=rb=0.2; rg=0</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>100</td>
<td>--</td>
<td>5</td>
<td>A-B-GND</td>
<td>ra=rb=0.2; rg=0</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>--</td>
<td>5</td>
<td>3 PHASE</td>
<td>ra=rb=rc=0.2</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>--</td>
<td>0</td>
<td>5</td>
<td>3 PHASE</td>
<td>ra=rb=rc=0.2</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>85</td>
<td>--</td>
<td>5</td>
<td>A-B-GND</td>
<td>ra=rb=0.2; rg=0</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>85</td>
<td>--</td>
<td>5</td>
<td>3 PHASE</td>
<td>ra=rb=rc=0.2</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>100</td>
<td>**</td>
<td>5</td>
<td>B-GND</td>
<td>0.2</td>
<td>Simultaneous Fault</td>
</tr>
<tr>
<td></td>
<td>**</td>
<td>100</td>
<td>5</td>
<td>A-GND</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>**</td>
<td>5</td>
<td>B-GND</td>
<td>0.2</td>
<td>Simultaneous Fault</td>
</tr>
<tr>
<td></td>
<td>**</td>
<td>0</td>
<td>5</td>
<td>A-GND</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>85</td>
<td>**</td>
<td>5</td>
<td>B-GND</td>
<td>0.2</td>
<td>Simultaneous Fault</td>
</tr>
<tr>
<td></td>
<td>**</td>
<td>90</td>
<td>5</td>
<td>A-GND</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>70</td>
<td>--</td>
<td>5</td>
<td>A-B-GND</td>
<td>ra=rb=0.2; rg=20</td>
<td>Evolving Fault</td>
</tr>
<tr>
<td>29</td>
<td>85</td>
<td>--</td>
<td>5</td>
<td>A-B-GND</td>
<td>ra=rb=0.2; rg=20</td>
<td>Evolving Fault</td>
</tr>
<tr>
<td>30</td>
<td>--</td>
<td>50</td>
<td>5</td>
<td>A-GND</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>--</td>
<td>10</td>
<td>5</td>
<td>A-B-GND</td>
<td>ra=rb=0.2; rg=10</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>85</td>
<td>--</td>
<td>120</td>
<td>A-GND</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>70</td>
<td>--</td>
<td>120</td>
<td>A-GND</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Case No.</td>
<td>L1 Fault Location (%)</td>
<td>L2 Fault Location (%)</td>
<td>L1, L2 Length (miles)</td>
<td>Fault Type</td>
<td>Fault Resistance (Ohms)</td>
<td>Notes</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------</td>
<td>-----------------------</td>
<td>-----------------------</td>
<td>------------</td>
<td>------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>34</td>
<td>70</td>
<td>--</td>
<td>120</td>
<td>A-GND</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>25</td>
<td>--</td>
<td>120</td>
<td>A-GND</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>25</td>
<td>--</td>
<td>120</td>
<td>A-B-GND</td>
<td>ra=rb=0.2; rg=20</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>--</td>
<td>0</td>
<td>120</td>
<td>3 PHASE</td>
<td>ra=rb=rc=0.2</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>--</td>
<td>0</td>
<td>120</td>
<td>3 PHASE</td>
<td>ra=rb=rc=0.2</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>--</td>
<td>--</td>
<td>120</td>
<td>--</td>
<td>--</td>
<td>Energize Bus Capacitors</td>
</tr>
<tr>
<td>40</td>
<td>85</td>
<td>--</td>
<td>120</td>
<td>A-B-GND</td>
<td>ra=rb=0.2; rg=0</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>85</td>
<td>--</td>
<td>120</td>
<td>A-B-GND</td>
<td>ra=rb=0.2; rg=30</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>100</td>
<td>--</td>
<td>120</td>
<td>A-GND</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>100</td>
<td>--</td>
<td>120</td>
<td>A-B-GND</td>
<td>ra=rb=3; rg=10</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0</td>
<td>--</td>
<td>120</td>
<td>A-GND</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>0</td>
<td>--</td>
<td>120</td>
<td>A-B</td>
<td>ra=rb=3</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>0</td>
<td>--</td>
<td>120</td>
<td>A-B-GND</td>
<td>ra=rb=rg=1</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>0</td>
<td>--</td>
<td>120</td>
<td>3 PHASE</td>
<td>ra=rb=rc=0.2</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>70</td>
<td>--</td>
<td>120</td>
<td>A-B-GND</td>
<td>ra=rb=3; rg=20</td>
<td>Evolving Fault</td>
</tr>
<tr>
<td>49</td>
<td>--</td>
<td>10</td>
<td>120</td>
<td>3 PHASE</td>
<td>ra=rb=rc=1</td>
<td>Evolving Fault</td>
</tr>
<tr>
<td>50</td>
<td>--</td>
<td>50</td>
<td>120</td>
<td>A-GND</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>100</td>
<td>--</td>
<td>40</td>
<td>A-GND</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>100</td>
<td>--</td>
<td>40</td>
<td>3 PHASE</td>
<td>ra=rb=rc=1</td>
<td>Evolving Fault</td>
</tr>
<tr>
<td>53</td>
<td>85</td>
<td>--</td>
<td>40</td>
<td>A-B-GND</td>
<td>ra=rb=1; rg=4</td>
<td>Evolving Fault</td>
</tr>
<tr>
<td>Case No.</td>
<td>L1 Fault Location (%</td>
<td>L2 Fault Location (%)</td>
<td>L1,L2 Length (miles)</td>
<td>Fault Type</td>
<td>Fault Resistance (Ohms)</td>
<td>Notes</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------</td>
<td>-----------------------</td>
<td>----------------------</td>
<td>------------</td>
<td>------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>54</td>
<td>100</td>
<td>--</td>
<td>40</td>
<td>A-GND</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>70</td>
<td>--</td>
<td>40</td>
<td>A-B-GND</td>
<td>ra=r_b=2; r_g=4</td>
<td>Evolving Fault</td>
</tr>
<tr>
<td>56</td>
<td>25</td>
<td>--</td>
<td>40</td>
<td>A-B-GND</td>
<td>ra=r_b=4; r_g=30</td>
<td>Evolving Fault</td>
</tr>
<tr>
<td>57</td>
<td>0</td>
<td>--</td>
<td>40</td>
<td>A-B-GND</td>
<td>ra=r_b=1; r_g=7</td>
<td>Evolving Fault</td>
</tr>
<tr>
<td>58</td>
<td>0</td>
<td>--</td>
<td>40</td>
<td>3 PHASE</td>
<td>ra=r_b=r_c=1</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>0</td>
<td>--</td>
<td>40</td>
<td>A-GND</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>100</td>
<td>--</td>
<td>40</td>
<td>A-GND</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>--</td>
<td>0</td>
<td>40</td>
<td>3 PHASE</td>
<td>ra=r_b=r_c=0.2</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>--</td>
<td>10</td>
<td>40</td>
<td>3 PHASE</td>
<td>ra=r_b=r_c=0.2</td>
<td>Evolving Fault</td>
</tr>
<tr>
<td>63</td>
<td>--</td>
<td>0</td>
<td>40</td>
<td>3 PHASE</td>
<td>ra=r_b=r_c=0.2</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>--</td>
<td>50</td>
<td>40</td>
<td>3 PHASE</td>
<td>ra=r_b=r_c=2</td>
<td>Evolving Fault</td>
</tr>
<tr>
<td>65</td>
<td>--</td>
<td>90</td>
<td>40</td>
<td>A-B</td>
<td>ra=r_b=3</td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>--</td>
<td>100</td>
<td>40</td>
<td>A-GND</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>
REFERENCES


7. Provided by A. G. Phadke.

VITA

The author was born on April 12, 1972, and grew up near Covington, Virginia. He graduated from Alleghany High School first in his class in 1990. Subsequently, he attended Virginia Tech to study electrical engineering, graduating in 1994 in the top 15% of his class. In the summers of 1992, '93, and '94, he interned with the electrical department of Westvaco Corporation in Covington, Virginia. In the fall of 1994, the author enrolled in the graduate school at Virginia Tech as a graduate research assistant, and received an M. S. degree in electrical engineering in 1996.

David M. Smith