A FEASIBILITY STUDY OF AN ADAPTIVE RECLOSING RELAY

by

Sundararaman Vaidyanathan

Thesis submitted to the faculty of the

Virginia Polytechnic Institute and State University

in partial fulfilment of the requirements for the degree of

Master of Science

in

Electrical Engineering

APPROVED:

Arun G.Phadke, Chairman

J.De La Ree Lopez

Kwasur Tam

October 1988

Blacksburg, Virginia

A FEASIBILITY STUDY OF AN ADAPTIVE RECLOSING RELAY

by

Sundararaman Vaidyanathan

Committee Chairman: Arun G. Phadke

Electrical Engineering

(ABSTRACT)

Logic for an adaptive reclosing relay has been developed. The relay works correctly in a wide number of fault cases. The relay has the following distinguishing characteristics :

(a) Reclosing into a three phase fault is avoided under all circumstances.

(b) The logic is applicable only for circuit breakers which have reclosing on individual phases.

(c) The relay works correctly in the case of both (shunt) compensated and uncompensated lines.

The logic of the relay and its associated algorithms were tested on several EMTP simulations of transmission line faults. In each of the cases simulated, six voltages (on either side of the Circuit Breaker) and the three currents in each of the phases of a three phase transmission line were filtered using a Finite Impulse Response digital filter. Positive sequence phasors of currents and voltages at various instants of time were calculated by the use of Discrete Fourier Transforms. The reclosing relay makes its decisions using these filtered phasors and symmetrical components as its input. The speed at which the decision is made is a little more than 1 cycle of the power frequency in most cases. The reclosing relay can be viewed as an example of an Adaptive Protection System.

ACKNOWLEDGEMENTS

The preparation of this thesis would not have been possible without the support given by a number of people. My sincere thanks to my advisor Prof. Arun G.Phadke who has guided me throughout this research. I would also like to thank and his students in the Department of Electrical Engineering at the University of Minnesota, who have put up with me during the time I have been in their lab, running EMTP simulations on their computer. Last but surely not the least, my infinite gratitude to my brother and his wife for, if it had not been for their constant encouragement, this thesis would have been non-existent.

TABLE OF CONTENTS.

CHAPTER 1

INTRODUCTION

1.1 STATEMENT OF THE PROBLEM

The objective of this thesis is to do a feasibility study of an Adaptive Reclosing Relay. A Computer Based Reclosing Relay was considered an excellent subject for this study since it has the following characteristics :

(a) A logic capability, provided by the algorithms running in the computer.

(b) A memory, provided by the storage peripherals of the computer.

(c) A communication capability, provided by the digital peripherals of the computer.

The above characteristics help a Computer Based Reclosing Relay in initiating the tripping or closing commands, based on the prevalent system conditions. The use of a Computer Based Relay becomes especially attractive when we also consider the fact that computers have taken a dominant role in recent years in the functioning of a Power System.

1.2 HISTORICAL BACKGROUND

To understand the work done by the author it would be helpful to present an account of some recent developments that have taken place in the area of digital (computer) based power system protection. Early attempts in this area were limited to the application of minicomputers which were available at the time for process control applications [1], [2], [3]. Recent developments in the area of microcomputer based protection systems utilizing digital technology, have made what once was an exercise in determining the feasibility of 'computer relaying' to a serious study of an engineering alternative to conventional methods of power system protection [4] .

While computer based relays do not by themselves signify a departure from conventional relaying practices, they do so through their communication capability, extensive storage capacity and self-diagnostic capabilities. These features also make way for a system wide integration which could have a far reaching influence on power system engineering as a whole and lead to a considerable improvement in the protection system's performance as well.

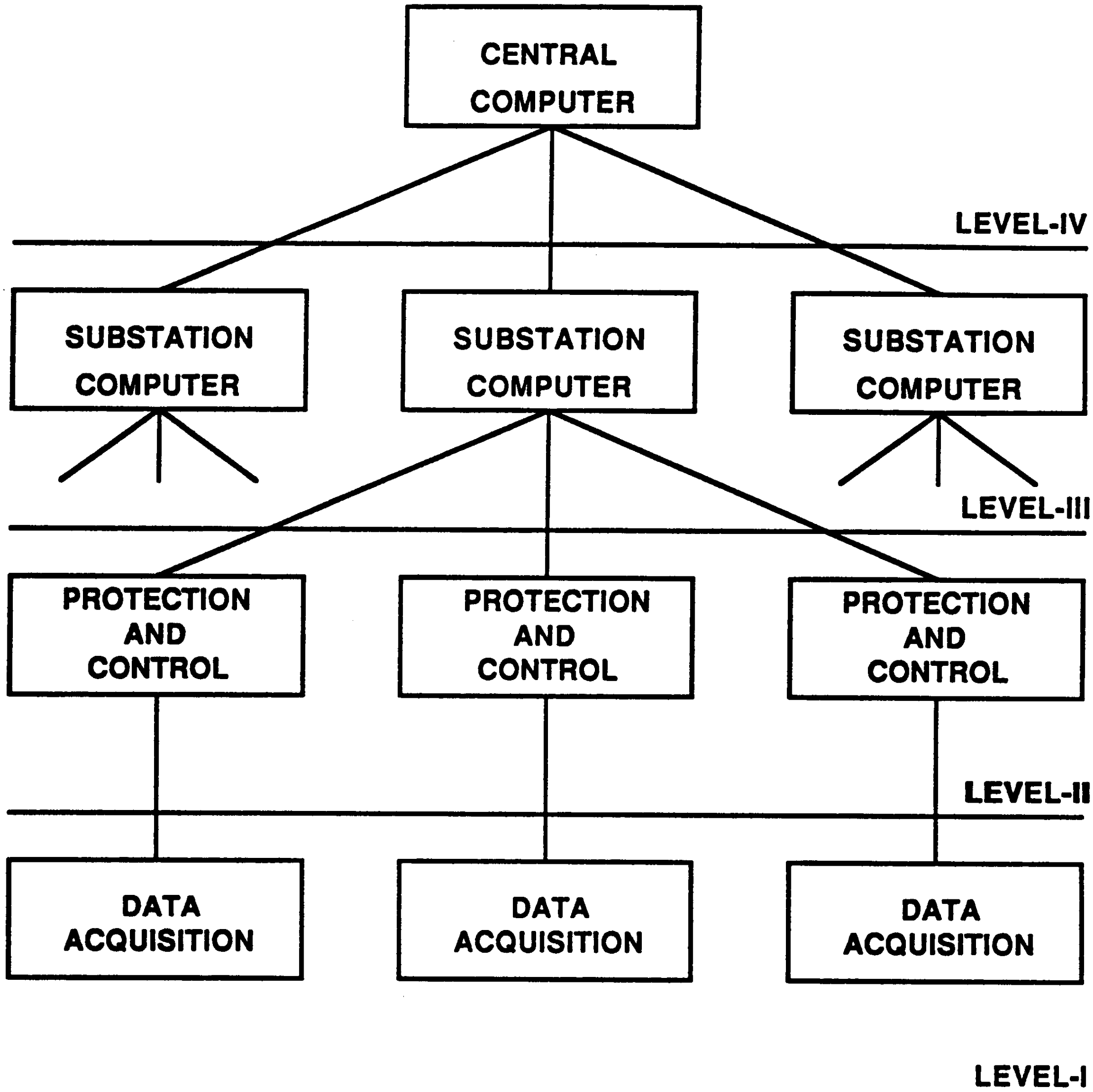


FIG 1.1 :STRUCTURE OF THE HEIRARCHICAL INTEGRATED PROTECTION AND CONTROL SYSTEM

A structural diagram of an integrated control and protection system of which the designed Reclosing Relay is assumed to be a part is shown in fig 1.1. It shows a four level hierarchy, starting at the bottom at Level-I, which is the data acquisition level (interfaced with transducers: for example, current and voltage transformers) in the switch-yard. Level-II consists of protection and control computers. These are microcomputer-based protection systems capable of performing relaying functions even if the communication links to the higher hierarchies fail. The communication links between Level-I and Level-II are high speed links. Level-III computers are host computers within the substation. They serve to collect and transmit data from the protection computers to the center and to communicate control decisions or queries from the center to the protection devices. The communication links between Level-II and Level-III are usually of moderate speeds. Level-III computers have access to substation-wide data and can hence participate in some analytical tasks and also contribute to the Adaptive Protection function through Level-II computers. Computers at Level-III are connected to the Level-IV computers at the system center through available mass- communication channels.

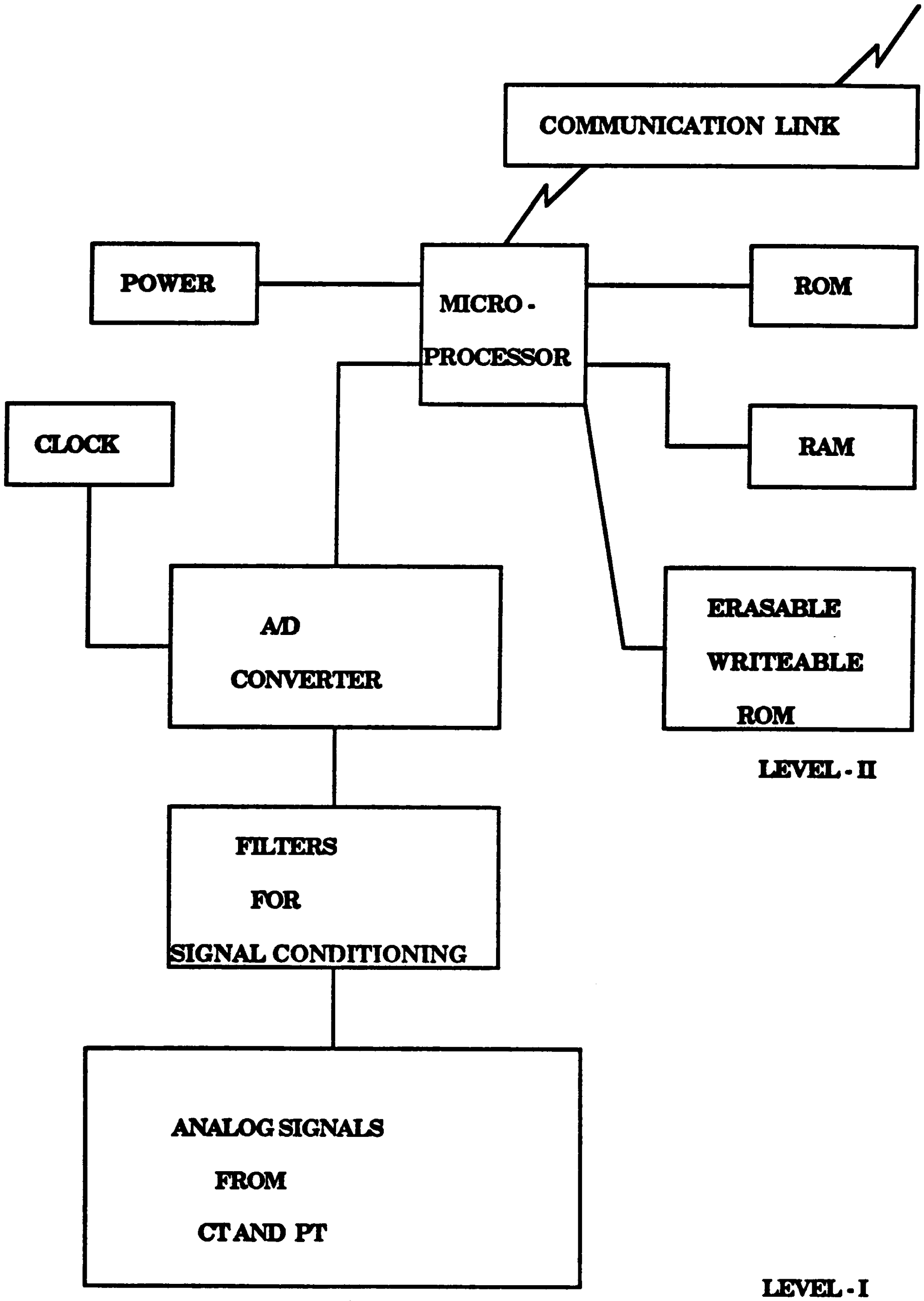


FIG 1.2 : EXAMPLE OF RELAYING COMPUTER-SUBSYSTEMS

Computers at this level have access to system wide data, and control actions (including many adaptive changes to protective functions) originate at this level.

Fig 1.2 shows a computer relay architecture which is a part of such a hierarchy. Only Level-I and Level-II subsystems are shown. At the bottom is level-I where the analog signals are filtered and reduced to suitable levels (generally, ± 10 volts maximum). These are then sampled by the A/D converter at a clock frequency determined by a programmable clock. At Level-II is the microprocessor and its storage peripherals. The fixed relaying program resides in the Read Only Memory (ROM) and its changeable settings reside in the Erasable writeable ROM. The transient data and intermediate results reside in RAM. The microprocessor can communicate with the the Level-III substation microprocessor over a 'communication link' [13].

These new systems could, using their self-diagnostic capabilities, provide alarms upon detection of relay failures through the utilization of their communication capabilities and also allow changes to be made in the relay settings to adapt to prevalent system conditions [6], [7]. The concepts of Adaptive relaying are elaborated in chapter 2.

In order for the computer based relays to find out what is going on in the transmission line (or any other apparatus they have to protect) they have to sample various parameters of the system (example, voltages and currents at a line terminal) at prescribed time intervals and arrive at decisions based on the effective processing of these samples. In this thesis, an Adaptive Computer Based Reclosing Relay has been designed. The logic of the relay and its associated algorithms were tested in an off-line environment on several EMTP simulations of transmission line faults. One of the 'associated algorithms' is a digital anti-aliasing filter which samples and filters the output of the EMTP simulation. The filtered output is then processed into phasors by another of the 'associated' algorithms. The reclosing relay uses the phasors to determine whether there is a fault on the system or not.

The reclosing relay and associated algorithms would run in the microcomputer at Level-II of the hierarchy as would the other normal substation protection and control algorithms. The reclosing relay logic would be activated only when the circuit breaker has tripped due to a fault or for maintenance.

1.3 METHOD ADOPTED

The primary purpose of reclosing circuit breakers following a fault is to return the transmission system to its normal configuration with a minimum time of outage and use of manpower. At the present time, automatic reclosing and manual reclosing are the principal ways in which reclosing is done.

Positive sequence voltage phasor at a power system bus is a parameter with a lot of importance. The collection of all positive sequence voltage phasors constitutes the state vector of a power system. The phasors could also be used for either fault detection, measurement of the local frequency or the rate of change of frequency. In addition, accurate measurement of phasor voltages in real time may lead to development and implementation of adaptive control procedures which can be based on direct measurement of key system variables [5].

Until a few years ago, an entirely satisfactory practical method of measuring positive sequence phasors did not exist. A new development in recent years, is to obtain the phasors of either the line voltages or currents using samples which are then processed using Digital Fourier Transforms. The samples are obtained after digitally filtering the signals so as to band-limit the frequency components in the signal to half the sampling frequency ' f_s'. In a field installation of such a system, analog anti-aliasing filters are generally used.

The author has proposed an Adaptive Reclosing Relay which utilizes the phasors obtained in the manner stated above for fault detection in the post fault period. This thesis contains a description of the logic used in the relay as also its associated algorithms. The testing of the logic was done using off-line fortran programs. The logic for the reclosing relay was tested on simulations of faults staged on a 765 kv transmission line on a computer using EMTP. The implementation of the system in the above form was also done by the author.

The results of the various EMTP simulations and the corresponding testing of the reclosing relay logic are summarized in chapter 7. The Fortran programs as well as selected samples of the phasors and plots of the filtered outputs are given in the appendix.

CHAPTER 2

ADAPTIVE TRANSMISSION LINE PROTECTION

2.1 INTRODUCTION

Adaptive Relaying is not a new concept. For example, Time Delay Over - current Relays adapt their operating time to the operating current magnitude and Directional Relays adapt to the direction in which the fault current is flowing. All of these however, are permanent characteristics of a relay or protection system and are included as part of the original design or installation to perform a given function. These do not make use of a much greater sophisticated capability of "Adaptiveness" which is achievable with the utilization of the present "state of the art" technology.

DEFINITION: Adaptive protection is a protection philosophy which permits and seeks to make adjustments to various protection functions in order to make them more attuned to prevailing power system conditions [6].

The following salient points should be noted with reference to the definition:

(a) The recognition that some of the settings of a protection system are based on the assumed system conditions and that if the system conditions change, then for better functioning, a change in these settings is desirable.

(b) An ability to change or adapt exists within the protection system.

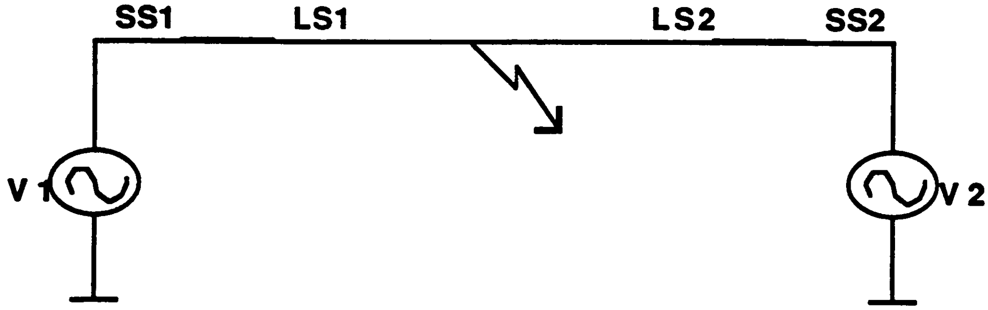
(c) We assume that the relays are computer based.

(d) If communication facilities are available, then the changes could be introduced through remote commands. The implementation of such a system would be made possible by the use of a Microcomputer Based Protective system.

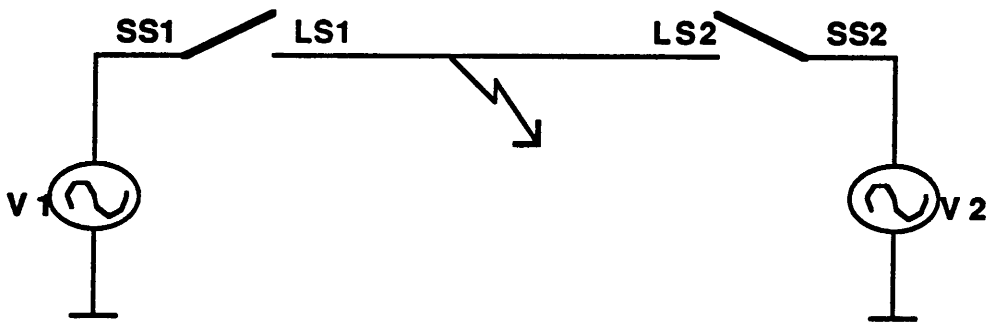
Adaptive techniques have been developed for the protection of equipment such as transmission lines, buses, transformers and for control and monitoring at the substation, regional and system levels. Due to the fact that there is not a lot of experience with the use of digital systems for the protection, control and monitoring the primary function of relaying is first perfected before proceeding to more sophisticated enhancements.

Most of the historic experience has been to make Adaptive changes by human intervention. There is a limited amount of Adaptive capability in some electromechanical and solid state analog systems. But, these are limited due to their small logic and memory capabilities. Digital Technology, on the other hand, is inherently programmable and and has for all practical purposes, unlimited logic and memory capabilities. They also have an infinitely enhanced capability as regards data transfer, analysis, communication between individual modules (both local and remote). These expanded capabilities make adaptive functions much more feasible in a digital system. With the use of digital technology in a microprocessor Based system, the implementation of the Adaptive concept becomes very straightforward.

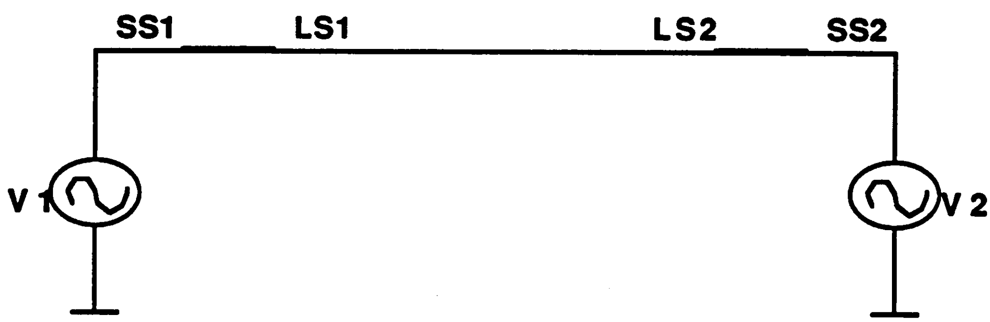
The use of the microprocessor also provides the system with the additional self-checking and diagnostic capabilities which enhance the reliability of the protective scheme considerably. Though manual periodic testing of the equipment is the most complete method of uncovering failures, there is a considerable risk of a hidden failure during the long period for which the equipment is unavailable, making this method suspect too. Automatic testing on the other hand introduces additional equipment which reduces the reliability of the system. Even Monitoring, which has the advantage of simplicity, has the limitation of not being able to detect subtle, potential failures [7].



(a) The system with a fault on it



(b) With the circuit breakers opened after the detection of the fault



(c) After the reclosing of the circuit-breakers following return to normalcy .

FIG 2.1: REPRESENTATION OF A SIMPLE SINGLE-PHASE A.C. SYSTEM

2.2 AUTOMATIC LINE RECLOSING

The primary objective of Automatic reclosing in circuit breakers, following a fault, is to return the transmission system to its normal configuration with a minimum outage time of the affected equipment and the least expenditure of manpower. The rapid restoration of the system to its normal configuration provides these additional benefits:

(a) Restoring the power transfer capability of the system to its former level in a minimum amount of time , and

(b) Reducing the service interruption time to the customers fed directly from the transmission system. Fig 2.1 shows the representation of a single phase a.c. system. In fig 2.1 (a), the system is shown with a fault on it. In fig.2.1 (b), the circuit breakers open following the detection of the fault by the protective relays. In fig 2.1(c), the circuit breakers reclose following the detection by the reclosing relays on either side of the line that the fault is no longer present and the system returns to normalcy.

Though Automatic reclosing practices are established to cover a wide range of possible events and configurations, they may not reflect the exact situation at the time a closing command is initiated. Also, reclosing could be considered as a godsend or a bane depending upon which element of the power system we are looking at:

(a) Stability: Many users prefer to forego the advantages of HSR due to the risk of it adversely affecting stability (if unsuccessful), though this decision would be determined by other factors too. For example, stability may be affected by three phase faults and not by single phase faults, or in certain areas alone and not throughout the system. In this research, reclosing is considered only in the case of single phase and phase-phase faults.

(b) In recent years, considerable study has focused on the possibility of stresses to the shafts and other components of a turbine-generator set following reclosing.

(c) Reclosing is not recommended at places where transformers are connected unless the protective zones can clearly rule out the possibility of an internal transformer fault.

(d) In the case of 'single pole switching,' only the faulted phase is deenergized on either side. In this case, the remaining energized phases tend to reignite the faulted arc after the primary arc has been interrupted due to the deenergization of the faulted phase. In this case, HSR would be unsuccessful. It is possible then to simply delay the reclose till the secondary arc extinguishes itself, as is done in Europe. This may be risky in instances where a sufficiently large current may sustain the secondary arc or the amount of time delay could exceed the permissible time for stability, voltage or negative sequence considerations [7] .

As is generally known, close to 80 to 90 % of all faults on overhead Transmission lines are transient. These result mainly from flashover of insulators induced by either lightning or wind caused conductor contact or conductor tree contact. Hence, by making the deenergization time long enough for the fault source to pass and the fault arc to be deenergized, service can be restored by automatically reclosing the breaker.

Reclosing can be done in the following ways:

(a) High speed (HSR) or time-delayed,

(b) one shot or multiple shot.

(c) supervised or unsupervised.

(d) with or without synchronizing check.

In the case of HSR, the first attempt is with no significant delay, where the Circuit Breaker's 'closing coil' is energized very shortly after its 'trip coil' has been energized by the protection relays. Hence the Breaker contacts are opened for a very short time (values range from 20-30 cycles). This delay is a function of the system normal voltage. Hence, a HSR cannot be programmed (or controlled) according to necessity. A time delayed automatic reclosing relay is controllable through the manipulation of the time delay after which the 'closing coil' is energized, though much slower than HSR.

'Synchronising-check' relays are used for energizing or restoring lines to service after an outage and in interconnecting preenergized parts of the system. Here, before closing, the phase angles between the voltage phasors at the two ends of the circuit breaker are checked. The circuit breaker is closed if the phase angle difference between the two ends is smaller than the relay-setting. For example, circuits used to transfer power between stations require reclosing of both terminals to restore service. HSR can be used in this case only if there are sufficient parallel ties in the network to provide an exchange of power between the two parts of the system. This is necessary to hold the source voltages in synchronism and essentially in phase during the line closing period. Reclosing could be done with synchronizing check in such cases. After their cycle of operation, if reclosures are successful at any point in time, the reclosing relay 'resets' after a preset time interval. If not, it moves to a 'lock-out' position so that no further reclosing is possible. When the circuit breaker has been closed manually after maintenance work, after a time delay , the relay resets.

Many stations and substations are left unattended most of the time. Hence, the reclosing requirements have to be programmed for the automation and control, either at the despatch center or into a local computer. It is the latter that is assumed for this report [8].

2.3 EXAMPLES OF ADAPTIVE RELAYING

Power system control is exercised at the generating stations through exciter and prime mover control functions, and in the substations through circuit breaker operations in order to reconfigure the network. Here, we will consider three system protection and control functions which can be enhanced through adaptive relaying:

(1) On-line detection of power system instability

(2) Load shedding and restoration

(3) Out-of-step blocking and tripping

2.3.1 On-line detection of instability: Detection of instability is not a traditional protection function. Instability is presently determined in planning studies of various contingency conditions and the results from the studies are then used in determining appropriate relay settings. Some relay settings, as in load shedding relays and out of step relays, depend upon the premise that given the current state of the power system, it will remain stable following a transient oscillation. On the other hand the out of step tripping relay is based upon the premise that the post-disturbance system will be unstable. If a developing transient could be judged to be stable or unstable, then the operation of such relays could be made to adapt to this judgment. It is for this reason that successful instability detection is a prerequisite for certain relaying function.

Real time detection of instability for a general multi-machine power system is an as yet unsolved problem. What we need in this case is advance (faster than real- time) determination of instability, which is an even more difficult problem. However, when systems behave like a two machine system, some progress can be made in this direction [6].

2.3.2 Load shedding and restoration: Load shedding relays are used to preserve the power system integrity after it suffers a loss of generation. If a power system forms a relatively small island, the frequency decays until the generation and load come to a balance point at a new reduced frequency. If the frequency at balance point is lower than a value at which the remaining generators can be operated safely, the generators must trip, causing a collapse of the 'island system.' Load shedding relays attempt to reduce system load in a controlled manner, so that the system frequency decay is arrested before the generator operating limit is reached. When the system is a small island, the frequency itself is a measure of the amount of load that must be shed. Load shedding relays can hence be set to operate at graduated frequency settings. On the other hand if the system is connected to neighboring systems, the frequency does not decay significantly even after a loss of generation, and load shedding must be achieved through supervisory control. Load restoration relays return the load to the system after the generation deficiency has been made up. Load restoration may be performed automatically with relays using frequency set points or with supervisory control. It should be done in steps such that:

(a) there are no pumping oscillations between load

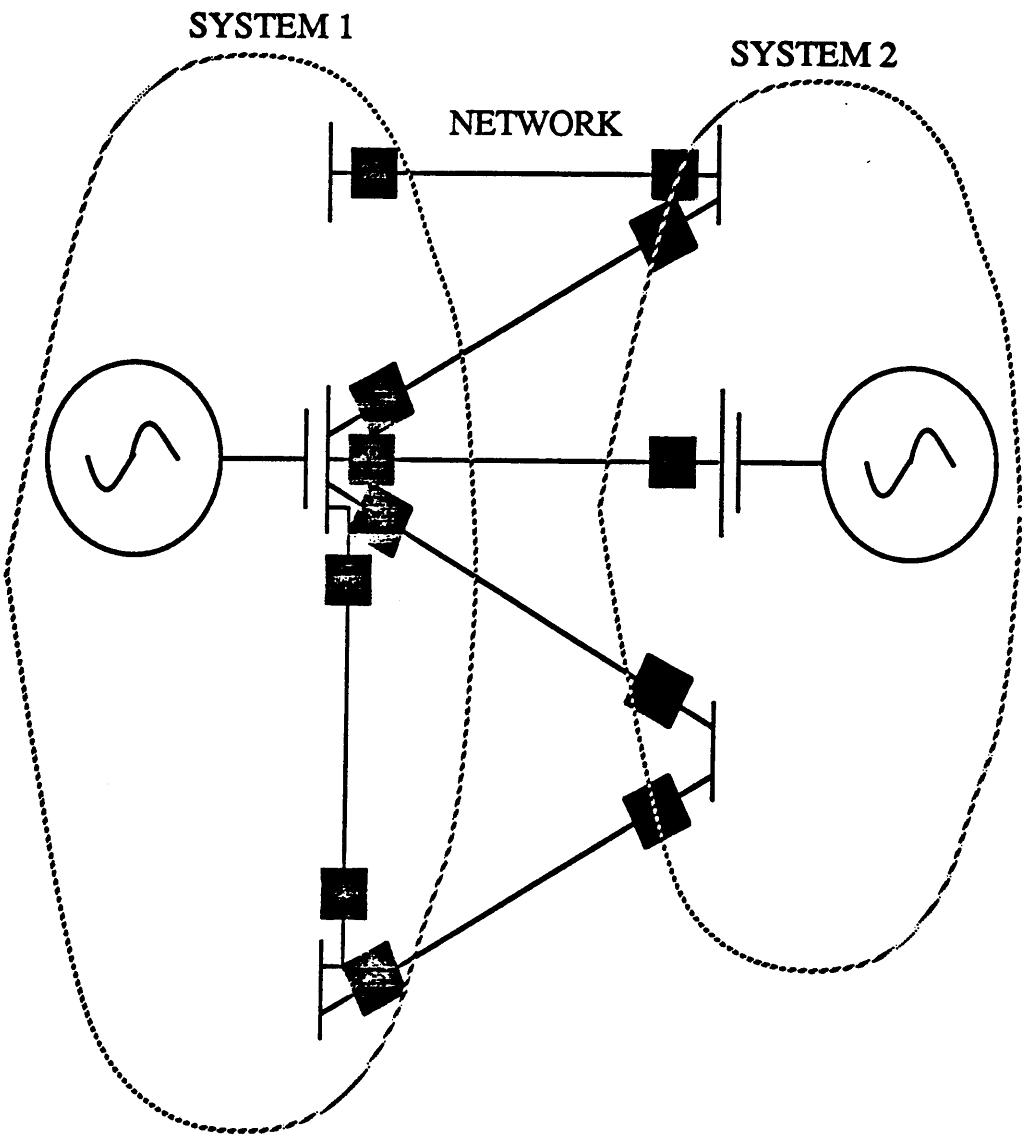


FIG 2.2: TWO EQUIVALENT MACHINES CONNECTED BY A LARGE NETWORK

shedding and load restoration relays, and

(b) no overload of tie-lines occurs.

In all cases, the generation deficiency could be measured accurately by forming a measure such as Area Control Error (ACE) which is used in Automatic Generation Control (AGC). The computer relays can measure frequency and power flows in tie lines accurately and with a fast response to changes. This information may be transmitted to a central location, where generation-load imbalance could be determined. A centrally directed (adaptive) load shedding (and restoration) program could then be carried out without waiting for the system frequency to actually decay [16].

2.3.3 Out-of-step blocking and tripping: The object of out-of-step relaying as it is applied to generators and systems is to eliminate the possibility of damage to a generator as a result of an out of step function; and, in the case of the power system, to supervise the operation of various relays such that when a system separation is imminent it should take place along boundaries which will form islands with matching load and generation. Generator out-of step condition can be detected easily from its terminal voltages and currents. In the case of a power network, determining appropriate boundaries of separation in order to form self sufficient islands is not an easy task.

The starting point for adaptive out-of-step relaying is the online determination of system instability. As mentioned in section 2.3.2, this can be done in the case of some simple power systems. Assuming that this type of analysis can be performed in a larger system as well, the out-of-step relaying then reduces to finding desirable separation points on the system with approximate match between load and generation. In fig 2.2 is shown a power system in which two islands (j=1,2) are formed with matched load and generation when its is determined that system instability is imminent. Islanding is achieved by tripping breakers $b_1..b_4$. This task must be performed at the control center where state estimation output is available [6].

2.4 ADAPTIVE FEATURES OF RECLOSING RELAYS

At the present time, Automatic reclosing represents Adaptive Reclosing control as it exists in the power system. The additional use of digital devices greatly enhances and simplifies the implementation of a system with Adaptive-reclosing capability. Given below are some of the aspects which could be enhanced using a digital relay:

(a) Capability of adapting to the clearing or otherwise of the fault: If there happens to be a permanent fault on a line, it is desirable for the first pole which closes to close at the maximum of the driving voltage, so that the d.c offset of the short circuit current could be minimized. If after the first pole has reclosed, no fault is detected on the line by the reclosing-relay then, the subsequent poles could be reclosed at the minimum of the driving voltage across the circuit breaker [7].

(b) The capability of Adapting to the type and severity of the fault: With the ability of digital relays to identify not only the location of a fault but also its type, it is a very minor task to include this information in the reclosing relay's control logic. In this research, the capability of determining the type and severity of the fault is incorporated into the reclosing relay algorithm in the form of a simple 'Distance relay'. The information about the type of the fault is then used to determine the pole on which the reclosing is to start. Chapter 6 gives details on the implementation of the Adaptive reclosing relay as it pertains to this research.

(c) 'Hard' and 'soft' Breaker operation: 'Hard' operations are those circuit breaker operations which stress the circuit breaker's insulation or mechanical structure. In the case of 'soft operations', there is no stress since, precisely controlling the closing and tripping impulse to the breaker reduces the stresses on it. This has the advantage of reducing the maintenance work on the circuit breakers.

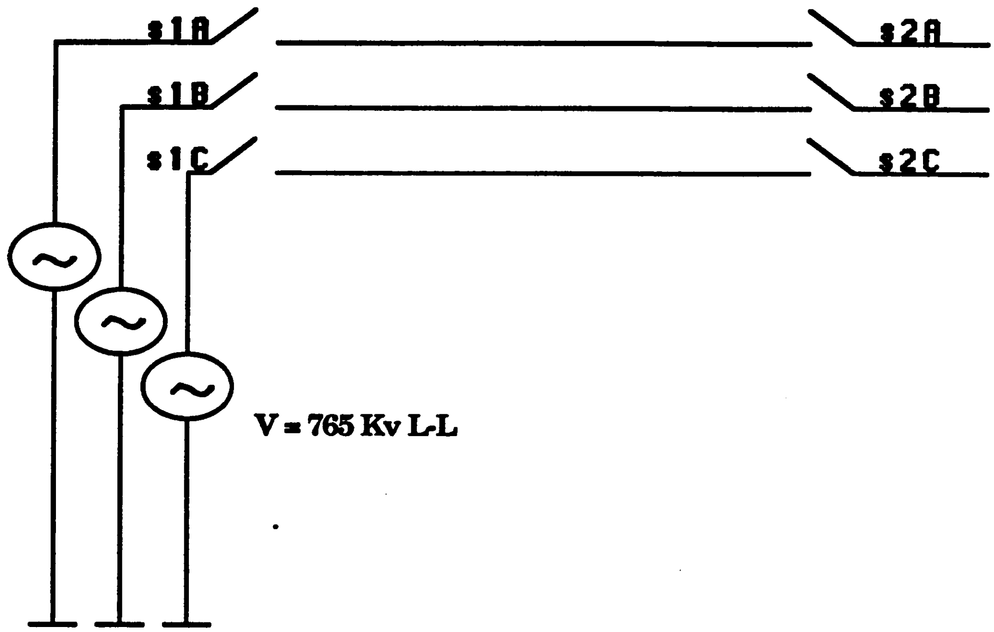
CHAPTER 3

SIMULATION RESULTS AND TECHNIQUE

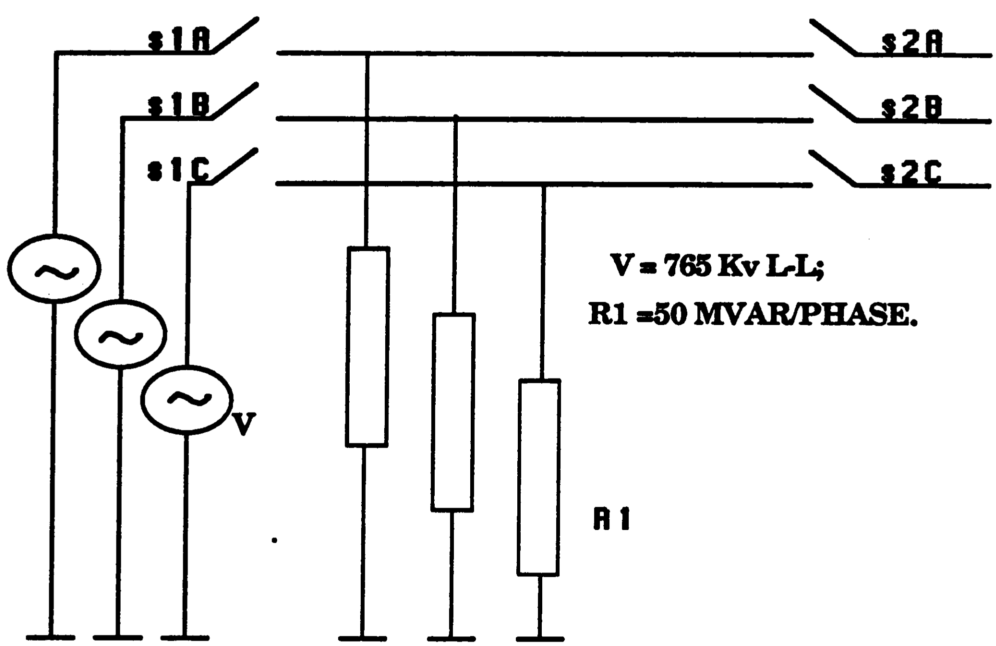
3.1 OBJECTIVES OF EMTP SIMULATIONS:

The main objective of the EMTP simulations was to simulate various faults in order to study the response of the relaying algorithms to these faults. A simple and yet typical system was first set up in order to carry out these simulations. During a simulation, the EMTP computed the system voltages (and currents) of selected nodes (and branches) at discrete times. This could be considered a very good approximation of the sampled data obtained from the power system in a real time situation.

In this thesis, the phasors of the voltages and currents in all the three lines were obtained at selected intervals of time. The values of the phasors on all the phases for various sequences of pole-closings (closing into a faulted line,closing into an unfaulted line, closing of two poles one after the other), permanent and temporary faults (L-g and L-L), closing on the remote end as well as various combinations of energisations were obtained. The reclosing relay algorithm that has been



(a) Single side energisation (uncompensated line)



(b) Single side energisation (shunt compensated line)

FIG. 3.1: BASIC CONFIGURATION

designed in this thesis, was tested on the output of the various EMTP- simulations.

When the first pole reclosed and it was determined that there was no fault in that phase, the voltage phasors on the remaining, unenenergized phases were examined to determine whether they were faulted or not. In certain cases, a second pole had to be reclosed before it could be determined whether the remaining phases are faulted or otherwise, as is mentioned in section 3.2.

3.2 DESCRIPTION OF EMTP SIMULATIONS

A reader who is not familiar with EMTP should be able to get a good knowledge of it in a quick manner by reading Ref. [9]. In figure 3.1(a) and 3.1(b), the configurations which were common to the cases simulated on EMTP are shown. Figure 3.1(a) shows an uncompensated three phase line which is energized from one end alone and Figure 3.1(b) shows shunt compensated three phase line with shunt reactors connected to all the three phases [10]. For the Three phase transmission line, both transposed and untransposed models were used. In various cases of EMTP simulations different types of faults were applied to the transmission line, in order to test the reclosing relay logic. In addition to cases with single side energization, fault cases were also run with:

(a) energization on both sides of the transmission line.

(b) Double circuit lines.

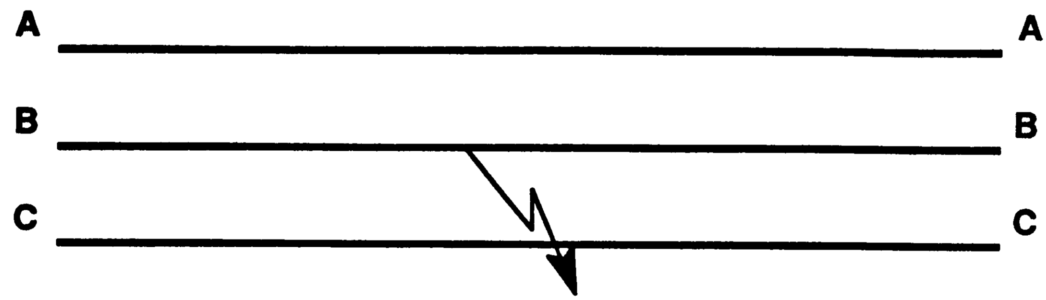
In all the cases, EMTP outputs three source side voltages, three line side voltages and the three line currents in the line to which the reclosing relay is connected. These outputs are then filtered by an anti-aliasing filter. A sample-listing of an EMTP data file is given in appendix-I. The plots of the line side voltages and currents in some selected cases are given in appendix.-II.

The following salient points should be mentioned regarding the configurations:

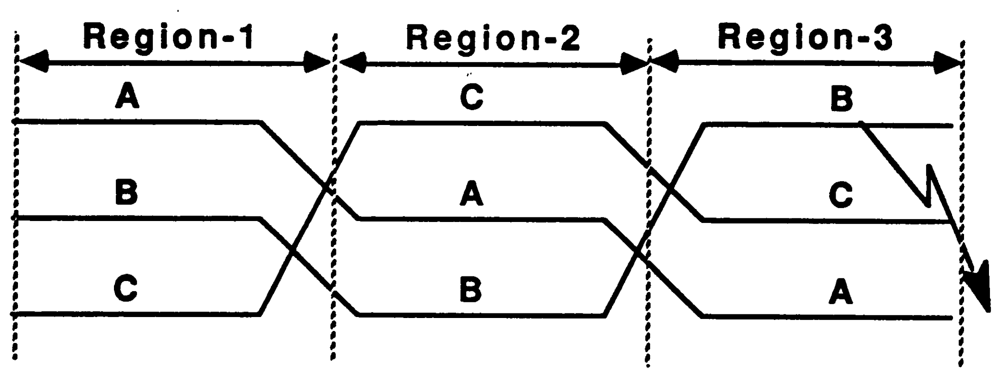
1. The faults which were simulated were phase to ground (Ø - G) and phase to phase (Ø - Ø) faults. No three phase faults were simulated since it was decided that there was to be no reclosing following a three phase fault.

2. It was assumed that each pole of the circuit breakers connected to the line (uncontrolled switches for EMTP simulations), was capable of individual control for reclosing.

3. The simulation was done for a total time period of



(a) UNTRANSPOSED LINE



(b) TRANSPOSED LINE

FIG 3.2 : GROUND FAULT ON B ON THE TWO TYPES OF TRANSMISSION LINES

0.2 seconds.

4. In each simulation, the switches on all three phases were opened after about 0.05 seconds from the beginning, as soon as the current passing through the particular phase reached its zero. This is taken care of by EMTP.

5. When a Ø - G fault was being simulated, the switch on the phase farthest from the fault was reclosed.

6. In each case, the time interval at which the output was received from the EMTP was 0.462 ms.

7. The equivalent source which has been used behind 'END-1' had:

(a) L-L Voltage Base =765 Kv ;

(b) Power Base = 100 MVA ;

(c) $Z_1 = (0.1878364 \times 10^{1} + j\, 0.563361 \times 10^{2})\ \Omega$.

(d) $Z_0 = (0.13963 \times 10^{2} + j\, 0.64498 \times 10^{2})\ \Omega$.

8. In the case of a compensated line, a 50 mvar per phase reactor was connected to 'END-1' in each phase of the transmission line.

9. The model used for the transmission line was:

(a) For an untransposed line : A distributed parameter travelling-wave propogation model of an untransposed line of length 56.8 mi., as shown in figure

3.2(a).

(b) For a transposed line: Three sections of untransposed transmission- lines, using the same model as the untransposed line, each of length 37.87 mi. connected to each other,thus making up a transposed line of length 113.6 mi., as shown in figure 3.2 (b).

In each of the above cases,

$$Z_1 = (0.022005 + j\ 0.59198)\ \Omega\ /\text{mi.}\ ;$$

$$Z_0 = (0.51631 + j\ 1.5891)\ \Omega\ /\text{mi.}\ ;$$

$$C_1 = 7.36\ \text{mhos/mi.}\ ;$$

$$C_0 = 5.302\ \text{mhos/mi.}\ ;$$

3.3 LIMITATIONS OF EMTP SIMULATION

The Circuit Breakers shown in fig 3.1, were simulated on EMTP using simple switches which were uncontrolled. The opening and closing times of these switches were determined by the author based on his judgment. This amounted to manual control. Hence, the testing of the reclosing was not done in an environment resembling real-time, where, the reclosing relay would be in control of these decisions. This was due to the fact that EMTP is itself an offline simulation

TABLE 3.1 SUMMARY OF EMTP SIMULATIONS

CASE	REACTOR PRESENT (Y/N)	PHASORS AFTER ONE POLE RECLOSES: LINE SIDE VOLTAGES (PER UNIT) VA	VB	VC	LINE CURRENTS (PER UNIT) IA	IB	IC	TRANS OR UNTRANS (T/U)	TYPE	SAFE CLOSING TIME CYCLES)
NF1	N	1.0	0.18	0.08	1.15	0.0	0.0	U	1	1.0
NFR1	Y	1.0	VAR	VAR	VAR	0.0	0.0	U	1	1.0
BG1	N	1.0	0.0	0.05	1.2	0.0	0.0	U	1	1.0
BGR1	Y	1.0	0.0	VAR	VAR	0.0	0.0	U	1	1.0
BC1	N	1.0	0.13	0.13	1.2	0.0	0.0	U	1	1.0
BCR1	Y	1.0	VAR	VAR	VAR	0.0	0.0	U	1	1.0
NF1	N	1.0	0.11	0.11	1.35	0.0	0.0	T	1	*
NFR1	Y	1.0	VAR	VAR	VAR	0.0	0.0	T	1	*
BG1	N	1.0	0.01	0.04	1.65	0.0	0.0	T	1	1.0
BGR1	Y	0.98	0.0	VAR	VAR	0.0	0.0	T	1	1.0
BC1	N	1.0	0.11	0.11	1.34	0.0	0.0	T	1	*
BCR1	Y	0.98	VAR	VAR	VAR	0.0	0.0	T	1	*
NF2	N	1.0	0.18	0.18	1.36	0.0	0.0	U	2	1.0
NFR2	Y	1.0	VAR	VAR	VAR	0.0	0.0	U	2	1.0
BG2	N	1.0	0.01	0.05	1.65	0.0	0.0	U	2	1.0
BGR2	Y	1.0	0.01	VAR	VAR	0.0	0.0	U	2	1.0
BC2	N	1.0	0.13	0.13	1.65	0.0	0.0	U	2	1.0
BCR2	Y	1.0	VAR	VAR	VAR	0.0	0.0	U	2	1.0
NF3	N	1.0	0.18	0.08	2.45	0.0	0.0	U	3	1.0
NFR3	Y	1.0	VAR	VAR	VAR	0.0	0.0	U	3	1.0
BG3	N	1.0	0.0	0.05	2.50	0.0	0.0	U	3	1.0
BGR3	Y	1.0	0.01	VAR	VAR	0.0	0.0	U	3	1.0
BC3	N	1.0	0.01	0.05	2.50	0.0	0.0	U	3	1.0
BCR3	Y	1.0	0.01	VAR	VAR	0.0	0.0	U	3	1.0

CASE:
NF: NO FAULT
BG: L-G FAULT ON B
BC: L-L FAULT ON B-C.
R: WITH REACTOR PRESENT

TYPE:
1= ENERGISATION FROM ONE END.
2= ENERGISATION FROM BOTH ENDS.
3= DOUBLE-CIRCUIT LINE.

* - DEPENDS ON THE CLOSING TIME OF THE SECOND POLE.

VAR: VARIES (DOES NOT HAVE A CONSTANT VALUE THROUGH OUT THE SIMULATION).

package which cannot be controlled by an external program. EMTP communicates with the external world through files. This can be considered a major disadvantage in utilizing EMTP for testing the logic of the reclosing relay. This is not to say that the method is wrong. It is just that the reclosing relay program had to be modified a little to output appropriate information to show us that it was making decisions appropriate to the case.

3.4 DISCUSSION OF EMTP RESULTS:

The results of the cases simulated and the testing of the reclosing relay are summarized in table 3.1. The cases that have been listed are:

Untransposed line (U) and Transposed line (T) with:

(1) No Fault (NF) without compensation; with reactive compensation (NFR).

(2) B-G fault (BG) without compensation; with reactive compensation (BGR).

(3) B-C fault (BC) without compensation; with reactive compensation (BCR).

These were simulated with energization on a single end (type=1); simulation on both ends (type= 2) or simulations done on a simple double circuit line (type=3).

In the case where there is no fault on the system (that is, in the case of NFx or NFRx, x=1..3), the reclosing program assumes that the line opens only when the the line requires maintenance. If the flag for maintenance is not up (that is, 'Y' in the case of the reclosing program), the program gives an error message and shuts off.

The last column in table 3.1 lists the 'safe closing time' (in cycles), that is, the time after which the decision made by the reclosing program could be considered reliable. This time is evaluated with reference to the time at which the distance relay detects a fault on the line. As can be seen from the table, the time is 1.0 cycles in all the cases where the time could be evaluated by the closing of a second pole. This is the minimum time for a reliable evaluation of the phasor from the samples obtained after reclosing on one pole.

In the cases where the reclosing program cannot make a decision without closing on another pole of the circuit breaker, the 'safe closing time' cannot be determined since this time depends on the time taken for the second pole to close. This time is dependent on the design of the circuit breaker and hence may not be the same at different times. This is the

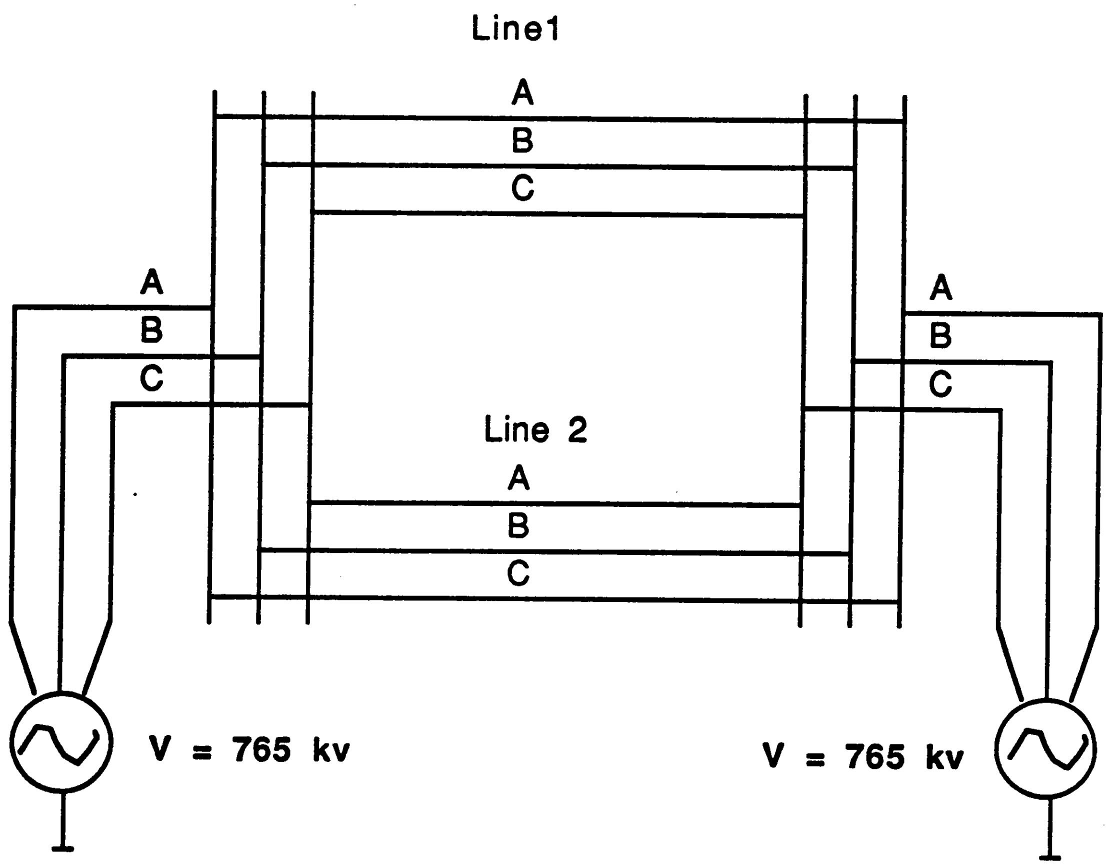


Figure 3.3: Double circuit line simulated on EMTP

reason that this time is marked with an asterisk (*).

It can be seen from Table 3.1 that the relay that has been designed works equally well in both the compensated and uncompensated cases. Also, as Table 3.1 shows, the values of the phasors of the signals in the case of the double-circuit lines is very similar to the case of single circuit lines. Hence, it is hardly surprising that the time intervals in this case are the same as in the single-circuit lines. Fig 3.3 illustrates the simple double-circuit line simulated in this thesis.

The reclosing algorithm also made correct decisions when it acted on various cases simulated on EMTP other than those listed in table 3.1. These cases were only slightly different from the cases listed in the table (for example, a combination of two of the cases listed as in the case of a temporary phase to ground fault, which is a combination of the no fault case and ø-g case) and were specifically designed to test the versatility of the algorithm.

CHAPTER 4
ANTI-ALIASING FILTER

4.1 INTRODUCTION:

The period of interest for a high speed computer based reclosing relay is of the order of a cycle following reclosing. The voltage and current waveforms during this period as also at the time of occurrence of a fault, contain significant amounts of transient components in addition to the fundamental frequencies. The calculation of the phasors of the signals outputted from EMTP assumes that the input signals in such cases must be band-limited to satisfy the 'Nyquist criterion', that is, filtered by a low pass filter for which the cutoff is half the sampling frequency 'f_s'. The reason for this is to avoid aliasing, that is, mistaking the frequency of the input. The next section explains in greater detail the necessity for band-limiting the signals. In the present case, the input signals from the EMTP were sampled and the sampling frequency used for calculating the phasor was 'f_s' = 720 Hz.

Hence, we have to use a filter whose cutoff frequency is $f_s/2$ = 360 Hz before calculating phasors of the waveforms. This must be done without introducing excessive delay in the fault detection process. In conventional relays, this would be achieved using analog filters. In the research described in this thesis, the filtering has been achieved using a 'Finite Impulse Response' digital filter [2,3]. The FIR filter makes use of Digital Fourier Transforms (DFT). The DFT is described in sect 4.3.

4.2 SAMPLING OF WAVEFORMS:

Consider fig 4.1(a). If h(t) is continuous at t = T, then a sample of h(t) at time T can be expressed as

$$h(t) = h(t)\ \delta(t-T) = h(T)\ \delta(t-T) \qquad (4.1)$$

that is, the impulse which occurs at the time T has the amplitude equal to the function at the time.

If h(t) is continuous at t = nT for n = 0, ± 1, ± 2,... then,

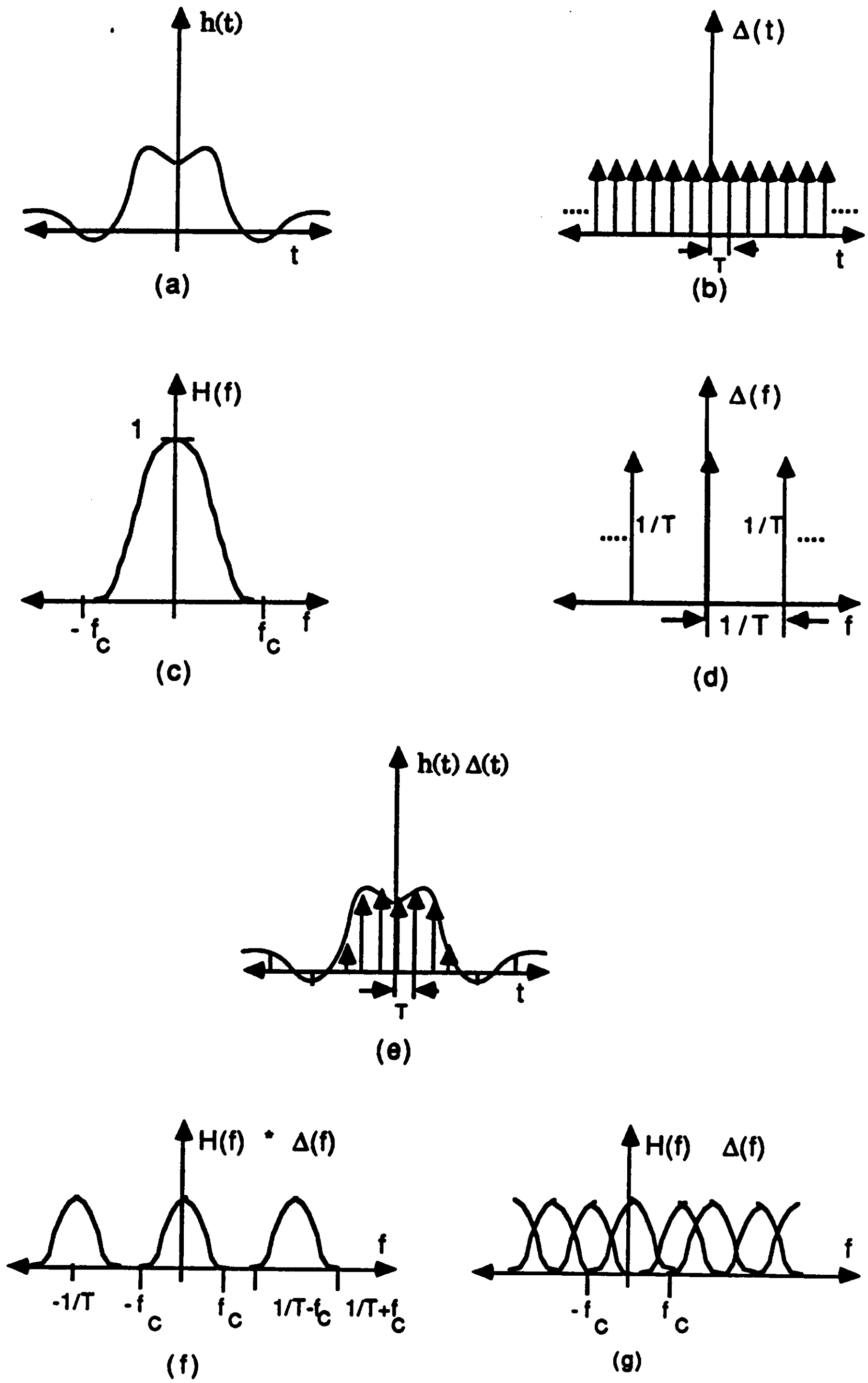


FIG 4.1: GRAPHICAL DEVELOPMENT OF THE FOURIER - TRANSFORM OF A SAMPLED WAVEFORM

$$h(t) = \sum_{n=-\infty}^{\infty} h(nT)\ \delta(t-nT)$$

$$= h(t) \sum_{n=-\infty}^{\infty} \delta(t-nT)$$

$$= h(t)\Delta(t) \qquad \text{--------------------} \qquad (4.2)$$

as seen in fig 4.1(e) and is a sampled waveform of h(t) with a sampling interval T. Since equation 4.2 is the product of the continuous function h(t) and the sequence of impulses (that is, the function $\Delta(t)$ shown in fig 4.1(b)), we can employ the frequency convolution theorem to obtain the Fourier Transform as $H(f)*\Delta(f)$, that is, the convolution of H(f) with $\Delta(f)$.

The Fourier Transform of the sampled waveform **h(t)** is then a periodic function where, one period is equal to the Fourier Transform of the continuous function h(t) (as seen in fig 4.1(f)). This is valid only if the sampling interval is sufficiently small since, if T were very large, the spacing between the impulses of $\Delta(f)$ decrease so that the convolution results in an overlapping function (as seen in fig 4.1(g)). This distortion of the desired Fourier Transform of a sampled function is the 'aliasing' mentioned in the previous section. How does one guarantee oneself that the Fourier Transform of a sampled function is not aliased? The 'sampling theorem' comes to our help here.

Sampling theorem: The sampling theorem states that if the Fourier Transform of a function h(t) is zero for all frequencies greater than a certain frequency 'f_c' (that is, the Fourier Transform is band-limited at the frequency f_c), then the continuous function h(t) can be uniquely determined from a knowledge of its sampled values,

$$h(t) = \sum_{n=-\infty}^{\infty} h(nT)\,\delta(t-nT)$$

if $T = 1 / 2f_c$ (this can be verified from figs 4.1(c) and 4.1(d)). The frequency $1/T = 2f_c$ is known as the 'Nyquist Sampling Rate' [15].

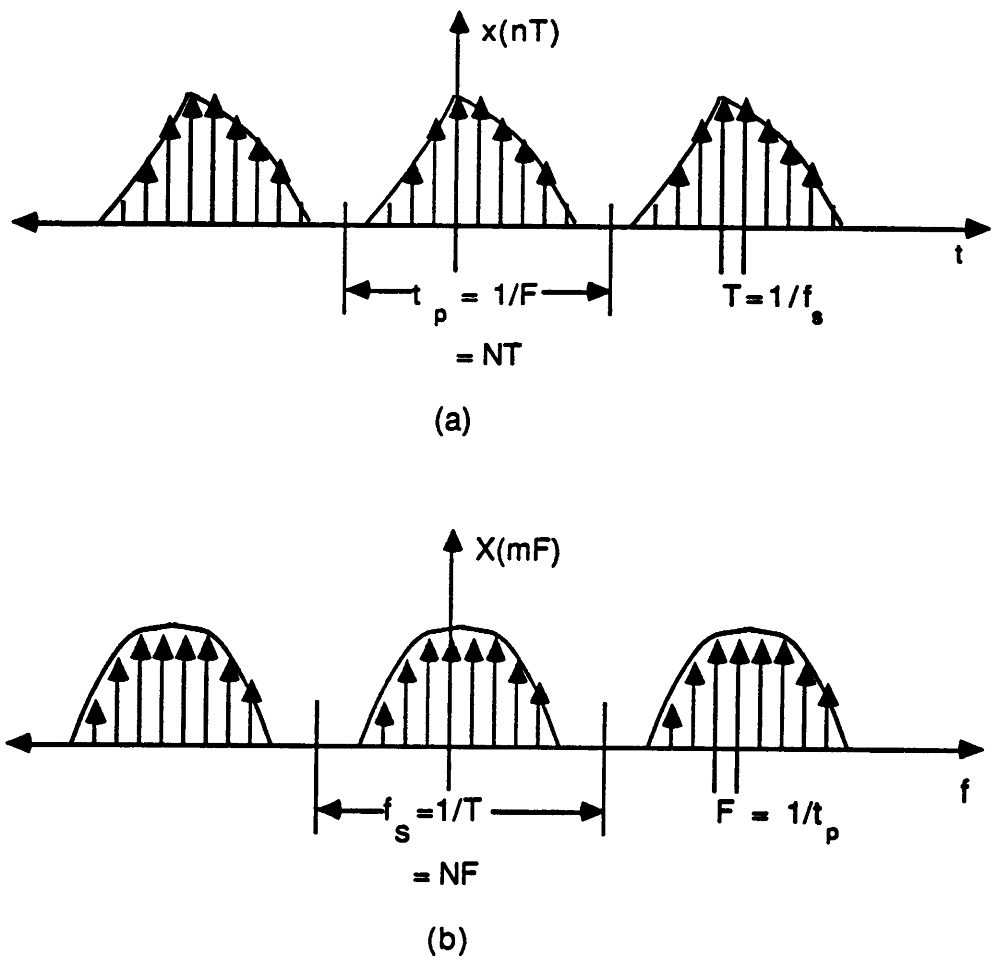


FIG 4.2: GRAPHIC ILLUSTRATION OF A DISCRETE-FOURIER TRANSFORM (DFT)

4.3 DISCRETE FOURIER TRANSFORM (DFT) :

The DFT has been used in this thesis not only in the anti-aliasing filter but also for the calculation of phasors (described in chap 5). Hence, it would be helpful to spend some time on it.

Let,

t_p = effective period of the time domain representation of any given function;

f_s = sampling rate or frequency with which the time domain representation is being sampled;

T = Sampling period in the time domain $=1/f_s$;

F = Sampling period in the frequency domain $=1/t_p$;

N = No. of samples per period.

As seen from fig 4.2, $f_s = NF$ and $t_p = NT$.

Hence,

$FT=1/N$

Let us define the quantity, $W_N = e^{-j2\pi FT}$

$$\therefore\ W_N = e^{-j(2\pi/N)} \qquad (4.3)$$

Then the DFT pair can be written as,

$$\text{DFT}: \quad X(mF) = \sum_{n=0}^{N-1} x(nT)\ (W_N)^{mn} \text{-----------------} \quad (4.4)$$

$$\text{IDFT}: \quad x(nT) = (1/N) \sum_{n=0}^{N-1} X(mF)\ (W_N)^{-mn} \text{-----------------} \quad (4.5)$$

A more compact relation is obtained if we let,

$X_m = X(mF)$ and $x_n = x(nT)$

Then, equations 4.4 and 4.5 can be represented as,

$$X_m = \sum_{n=0}^{N-1} x_n\ (W_N)^{mn} \text{-----------------} \quad (4.6)$$

$$x_n = (1/N) \sum_{n=0}^{N-1} X_m\ (W_N)^{-mn} \text{-----------------} \quad (4.7)$$

We can express these operations in symbolic form as,

$$X_m = D[\,x_n\,] \text{ or } x_n = D^{-1}[\,X_m\,]$$

We will now briefly state some of the properties of DFT:

(a) LINEARITY: Given that,

$$D[x_{1n}] = X_{1m} \text{ and}$$

$$D[x_{2n}] = X_{2m}$$

Then, $D\,[ax1n+bx_{2n}] = aX_{1m}+bX_{2m}$

(b) SHIFTING: $D\,[x_{n-k}] = (W_N)^{km}X_m$

(c) MODULATION: $D\,[(W_N)^{kn}x_n] = X_{m-k}$

(d) CONVOLUTION: $D\,[x_n*h_n] = D\,[\,\sum_{n=0}^{N-1} x_n h_{n-k}\,]$

$$= X_m H_m \text{-----} \qquad (4.8)$$

Convolution is the property that has been used in the anti- aliasing filter as explained in sect 5.4 [11].

Now consider a pure sinusoidal signal:

$$v(t) = \sqrt{2}\ V\cos(wt+\phi)$$

If this signal is to be sampled at the rate of N samples a cycle, then equation 4.6 yields for the fundamental frequency phasor of v(t)

$$V_1 = (N/\sqrt{2})Ve^{j\phi}$$

The fundamental frequency phasor associated with an arbitrary signal x(t) would be,

$$X_1 = (\sqrt{2}/N) \sum_{n=0}^{N-1} x_n (W_N)^n \qquad (4.9)$$

The above equation is utilized for generating the phasors of EMTP output signals after they have been filtered. The calc-

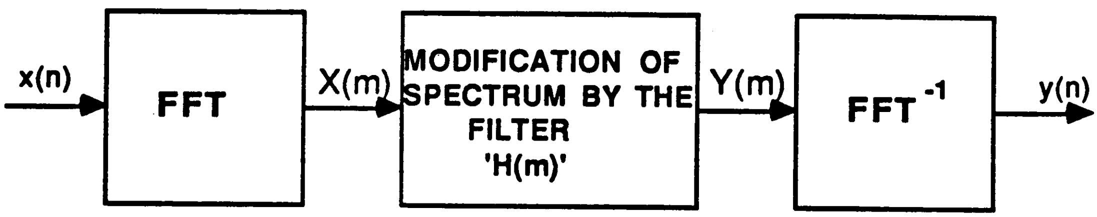


FIG 4.3: DIGITAL FILTERING USING THE FAST-FOURIER TRANSFORM (FFT)

ulation of phasors is examined in detail in chap 5 [16].

4.3.1 Fast Fourier Transforms (FFT): The 'Fast Fourier Transforms' are a set of algorithms that permit implementation of the DFT with considerable savings in computational time. It can hence be seen that the FFT is not a different transform from the DFT. One of the most important applications of the FFT is that of high speed convolution, a process made possible by the simplicity of the corresponding relationships in the transform domain. If x_n and h_n are relatively long functions, the number of computations required to perform a direct convolution become excessively large. However, according to equation 4.9, this operation is equivalent to multiplying the DFTs of the two signals. This property makes FFT very useful in digital filtering. Fig 4.3 illustrates a digital filter using FFT. We will now briefly look into an example of a FFT algorithm.

Consider equation 4.6. It can be expressed as,

$$X_0=x_0W^0 + x1W0 + \dots\dots\dots\dots + X_{N-1}W^0$$

$$X_1=x_0W^0+ x_1W^1 + \dots\dots\dots\dots + X_{N-1}W^{N-1}$$

$$X_2 = x_0 W^0 + x_1 W^2 + \ldots\ldots\ldots\ldots\ldots + X_{N-1} W^{2(N-1)} \quad --- \qquad (4.10)$$

$$\vdots$$

$$X_{N-1} = x_0 W^0 + x_1 W^2 + \ldots\ldots\ldots\ldots\ldots + X_{N-1} W^{(N-1)(N-1)}$$

This can be expressed in matrix form as ,

$$\mathbf{X} = [\mathbf{W}]\, \mathbf{x} \quad ---------------------------- \qquad (4.11)$$

where,

X = The column vector defining the transform;

x = The column vector defining the discrete time signal; and

$$[W] = \begin{bmatrix} W^0 & W^0 & \cdots & W^0 \\ W^0 & W^1 & \cdots & W^{N-1} \\ W^0 & W^2 & \cdots & W^{2(N-1)} \\ \vdots & \vdots & & \vdots \\ W^0 & W^{N-1} & \cdots & W^{(N-1)(N-1)} \end{bmatrix}$$

It can be seen from equations 4.10 that the number of operations to be performed to calculate the Transform is of the order of N^2. It has been shown [17] that if N were to be chosen as a multiple of 2 and a quantity L defined as,

$$L = \log_2 N \text{ --------------------------------- } \quad (4.12)$$

Then, [W] can be factored into L matrices in the form,

$$[W] = [W_1]\,[W_2] \,\ldots\ldots\ldots\, [W_L] \text{ -------------- } \quad (4.13)$$

Where, each row of a matrix $[W_k]$ has the property that it contains only two non-zero terms- W^0 and W^k where k is an integer, with $1 \le k \le L$. We then have,

$$X_s = [W_1]\,[W_2] \,\ldots\ldots\ldots\, [W_L]\, x \text{ ------------- } \quad (4.14)$$

The subscript 's' in X_s is to represent the fact that the order of the terms may have changed as compared to X. The operation for finding X_s takes $N\log_2 N$ complex operations in the worst case and hence is a significant improvement in the case of applications where the values of N are large. In the cases where N is small, the difference between N^2 and $N\log_2 N$ is not large enough to be of any significance [11].

In relaying applications, N is usually small (from 4 to 20 for most algorithms). Also, generally, only the fundamental frequency component (that is, m=1) is used in impedance relaying while a few harmonics are used in transformer algorithms. Hence, FFT has found little application in digital relaying [16].

4.4 DESCRIPTION OF THE FILTER ALGORITHM

A digital filter may be described as a computational process or algorithm that converts one sequence of numbers representing an input signal into another sequence of numbers representing an output signal and in which the character of the input signal is modified in a manner prescribed by the filter response stored in the memory of the computer [2]. As mentioned in sect 4.1, the kind of filter used in this thesis was a 'Finite Impulse Response' filter. A FIR may be realized in one of three ways namely,

(a) Nonrecursive (Direct Convolution) Realization: Here, the present value of the output depends only on the present and past values of the input.

(b) Recursive Realization: Here, the present output depends both on the input (present and/or past) and the previous values of the output.

(c) FFT Realization: Here, the input signal is first transformed using FFT, filtered as desired and then inverse transformed.

The method that has been used in this thesis is the nonrecursive (direct convolution) Realization.

A Finite Response Filter (FIR) is one in which the impulse response h_n is limited to a finite number of samples defined over the range $n_1 \leq n \leq n_2$ where n_1 and n_2 are both finite. The impulse response h_n can be expressed as

$$h_n = a_n \quad 0 \leq n \leq k \qquad \text{------------------} \qquad (4.15)$$
$$= 0 \quad \text{otherwise}$$

This can also be expressed in the form,

$$h_n = \sum_{i=0}^{k} a_i \, \delta_{n-i} \qquad \text{------------------} \qquad (4.16)$$

The transfer function (in the z-domain), corresponding to equation 4.16 can be expressed as,

$$H(z) = \sum_{m=0}^{k} a_m z^{-m} \text{ ------------------ } \quad (4.17)$$

$$= a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots\ldots\ldots + a_k z^{-k}$$

$$= a_0 + a_1 e^{-sT} + a_2 e^{-2sT} + \ldots\ldots\ldots + a_k e^{-ksT}$$

since, $z = e^{sT}$.

The integer k represents the order of the function. The output of the filter with the above impulse response can be shown to be,

$$y_n = \sum_{i=0}^{k} h_i x_{n-i} \text{ ------------------ } \quad (4.18)$$

Equation 4.18 is a convolution in the time domain and describes a non-recursive Realization for the FIR transfer function H(z) [11].

The implementation of the filter in the thesis consisted of two parts,

(a) A program to obtain the impulse response of the filter: The order of the filter chosen was two, it had poles at s_1 and s_2, a cutoff frequency of 360 Hz and a transfer function of,

$$H(z) = A\ [\ s_1 s_2 \ /\ (z + s_1)(z + s_2)\ ] \quad \text{-------------} \quad (4.19)$$

If we write this as a sum of partial fractions, we have the time domain impulse response,

$$h(t) = c_1\, e^{-s_1 t} + c_2\, e^{-s_2 t} \quad \text{-------------} \quad (4.20)$$

where,

$$c_1 = G_{60}\ [s_1 s_2 \ /\ (s_2 - s_1)]\ ;$$

$$c_2 = G_{60}\ [s_1 s_2 \ /\ (s_1 - s_2)]\ ;$$

$$G_{60} = A = \text{The inverse of the gain at 60 Hz}$$

$$= \sqrt{\ }\ [((s_2)^2 + (377)^2)\ ((s_2)^2 + (377)^2)]\ /\ (s_1\ s_2)$$

The reason for choosing the gain of the filter A equal to G_{60} was to make the gain of the filter equal to unity at the fundamental frequency of 60 Hz. The response is sampled at

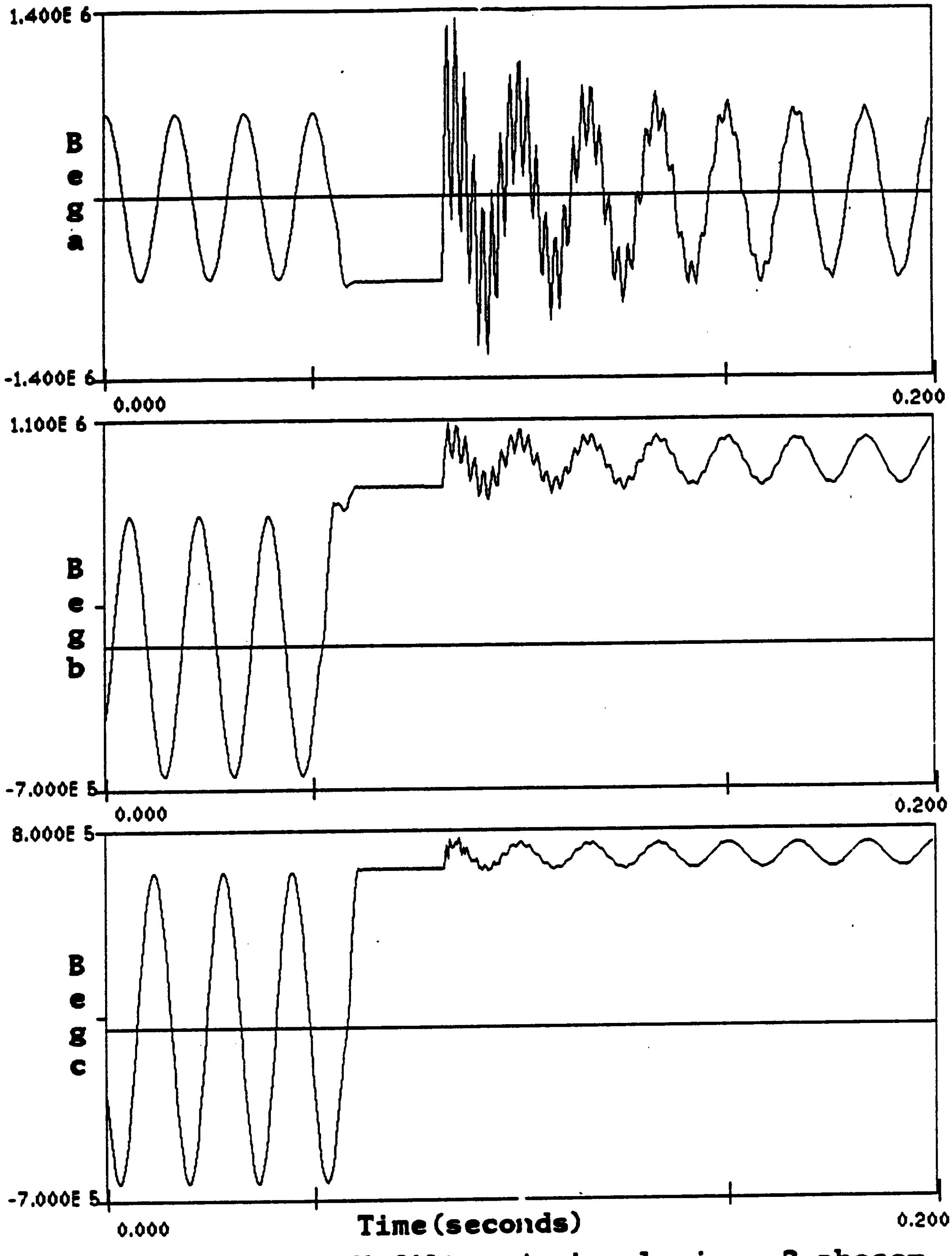


Figure 4.4: Unfiltered signals in a 3 phase-transmission line

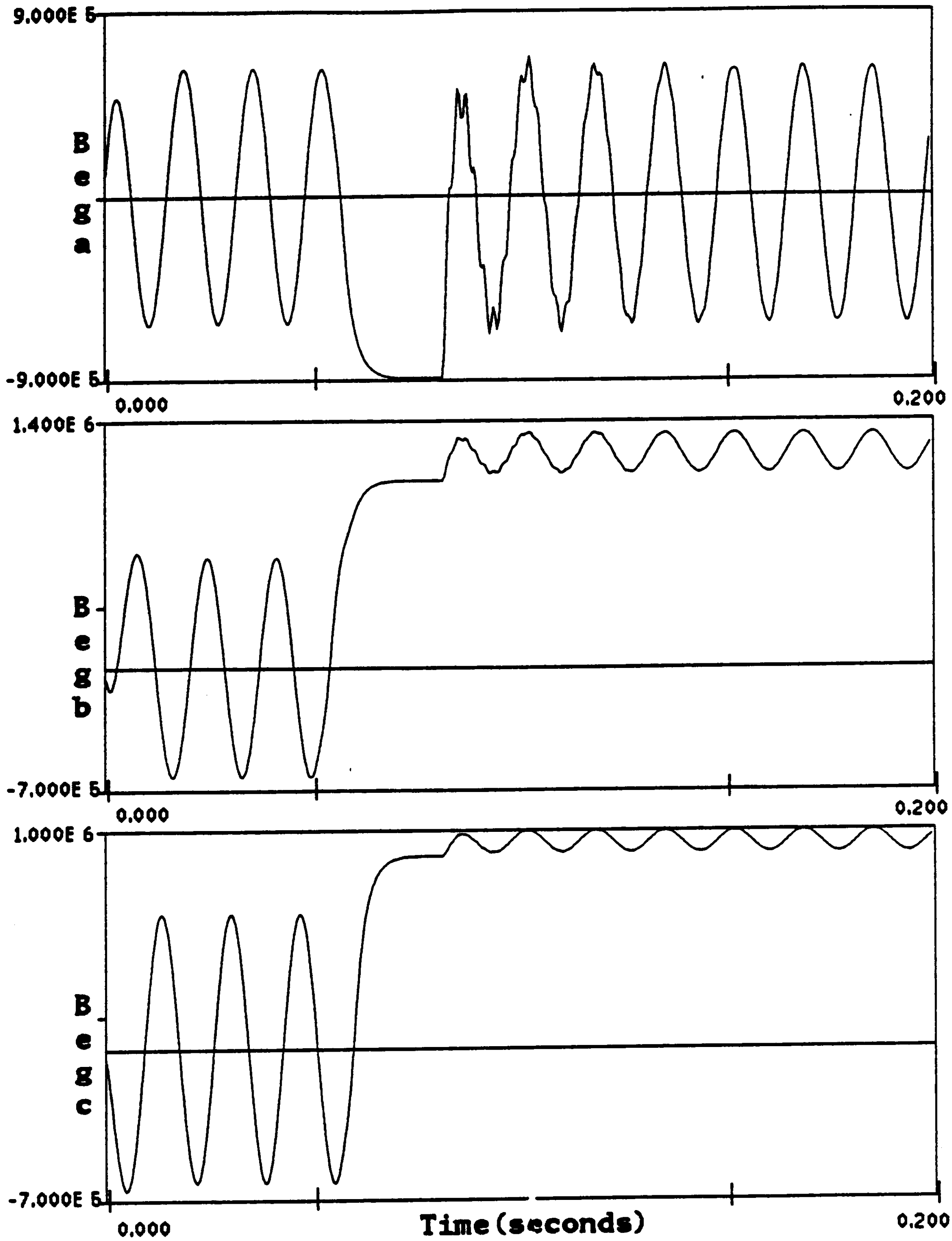


Figure 4.5: Filtered waveforms of signals given in figure 4.4

the sampling rate of 2160 Hz and communicated to the second program (in the case of the thesis, through a file) [16].

(b) A program which uses the logic of equations 4.16, 4.17 and 4.18. The order of the filter used is two (that is k=2) which was not large and hence, the convolution shown in equation 4.18 was done by direct enumeration using the formula rather than through the use of the FFT.

Listings of the above programs are given in Appendix-I. Figure 4.4 shows plots of unfiltered voltage signals from a 3Ø transmission line. Figure 4.5 shows the corresponding filter outputs.

CHAPTER 5

PHASOR CALCULATION

5.1 POSITIVE SEQUENCE MEASURING SYSTEMS

Positive sequence voltage phasor at a power system bus is a parameter of vital significance. The collection of positive sequence voltage phasors at all buses in a power system constitutes the state vector of the system. It is also one of the output variables in Load Flow and State Estimation programs. It should hence be a matter of concern that many of the phase angle measurement schemes in literature, measure the magnitude and phase on only one phase. In the presence of system unbalance, such a measurement is an approximation at best. Furthermore, almost all of these schemes measure the phase angle of the voltage by timing the zero crossing instant of the voltage waveform with respect to a standard reference pulse. Since the zero crossing instant of a voltage wave is affected by the harmonics and other noise in the wave, this is yet another source of error as far as the measurement of positive sequence phasors is concerned. The method used in this thesis utilizes Digital Fourier Transforms for obtaining the phasors of the various signals.

5.2 SUMMARY OF DERIVATIONS FOR POSITIVE SEQUENCE PHASORS :

Consider a sinusoidal input signal of frequency 'ω', ie,

$$x(t) = \sqrt{2}\ X\ \sin(\omega t+\phi) \text{ ------------------} \qquad (5.1)$$

The corresponding phasor is given by

$$\mathbf{X} = Xe^{j\phi} \text{ --------------------------------} \qquad (5.2)$$

$$= X\ (\cos\phi + j\ \sin\phi)$$

Assuming that x(t) has been sampled 'N' times per cycle of the 60 Hz. waveform to produce a sample set { x_k }, such that,

$$x_k = \sqrt{2}X\ \ \sin((2\pi/N)K+\phi) \text{ -----------------} \qquad (5.3)$$

where,

X = Magnitude of the input signal;

ø = phase of the signal with respect to a reference;

N = no of samples per cycle

K = an integer

In general, the function x(t) is not a pure sinusoid, during transient conditions. Assuming that such a non-sinusoidal signal x(t) is sampled at t = KT with { K = 0,1,...}. Hence,

$x_k = x(KT)$, where T= sampling interval.

For an impure signal x(t), only the optimum estimate of the phasor X_1 can be obtained. When the signal has been suitably 'band-limited' using an anti-aliasing filter as described in chapter 4, and the sample set { x_k} spans a multiple of one half the fundamental frequency period, the optimum estimate X_1, becomes the Fourier transform of x(t)

$$X_1 = (2/N) \sum_{K=0}^{N-1} [\, x_k e^{-j\,(2\pi T/To)k}]$$

where,

To is the period of the fundamental frequency wave.

Now,

To/T= N.

Hence,the Discrete Fourier Transform of { x_k } contains a fundamental frequency component given by,

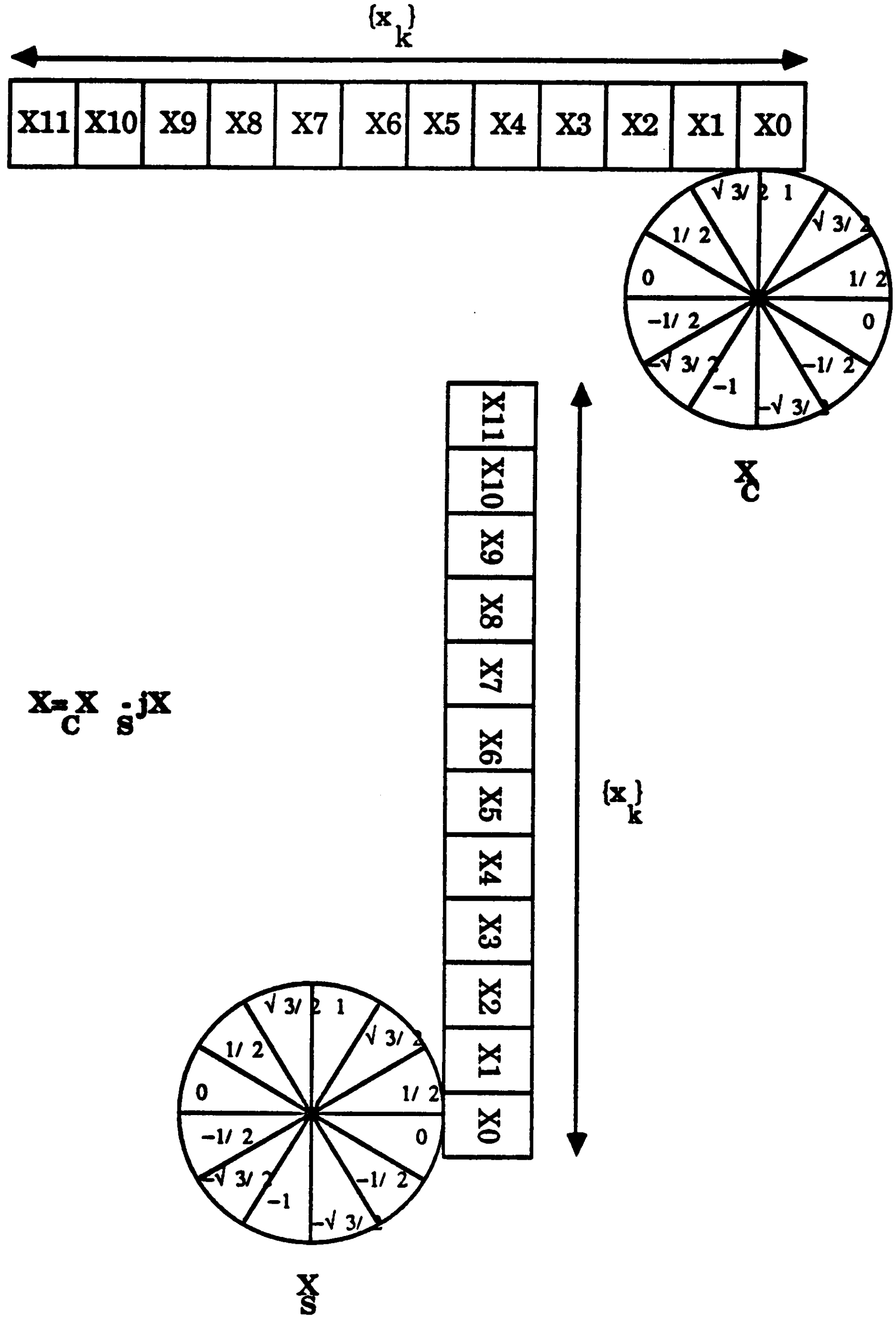


FIG 5.1 : CALCULATION OF PHASORS FROM SAMPLES

$$X_1 = (2/N) \sum_{K=0}^{N-1} [\, x_k e^{-j(2\pi/N)k}]$$

$$= (2/N) \sum_{K=0}^{N-1} [\, x_k \ (\cos(2\pi/N\ )k\ -j\sin(2\pi/N\ )k)] \text{ -----} \qquad (5.4)$$

In the present case,a 1 cycle sample window,with 12 samples per cycle is being used for the calculations.Hence, $N = 12$.

Let,

$$X_1 = X_c - jX_s \text{ --------------------------} \qquad (5.5)$$

Hence, from equations 5.3, 5.4 and 5.5 ,

$$X_c = \sqrt{2} x \sin\phi$$

$$X_s = \sqrt{2} x \cos\phi \text{-------------------------} \qquad (5.6)$$

From equations 5.2, 5.4 and 5.6, then the fundamental phasor representation, the phasor X is related to the fundamental frequency component of its D.F.T, X_1 by

$$X = (1/\sqrt{2})\, jX_1 = (1/\sqrt{2})\, (\, X_s + jX_c)\text{----------} \qquad (5.7)$$

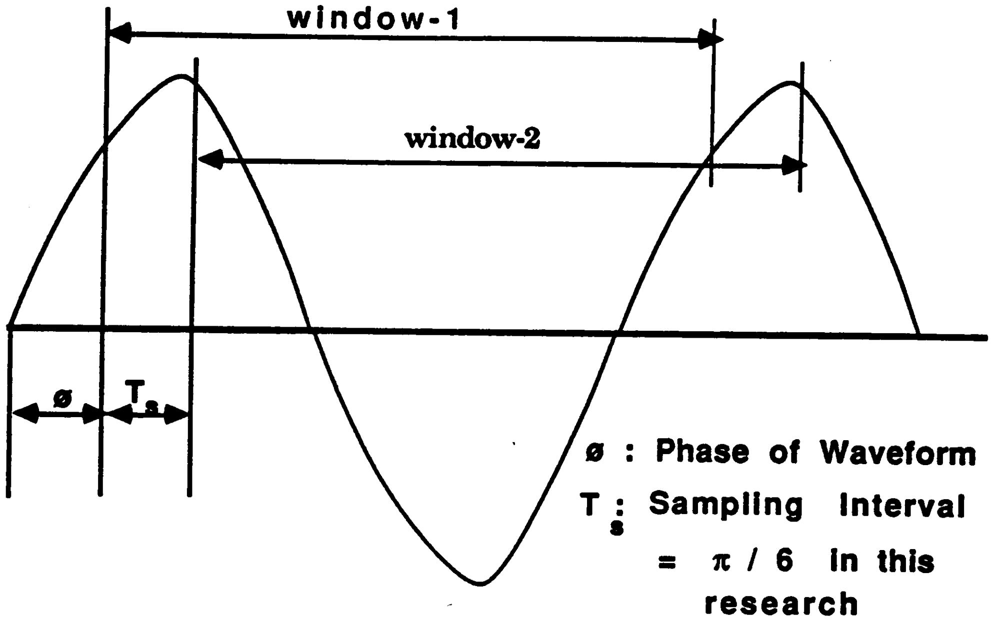


FIGURE 5.2: DATA WINDOWS USED IN THE CALCULATION OF THE PHASOR.

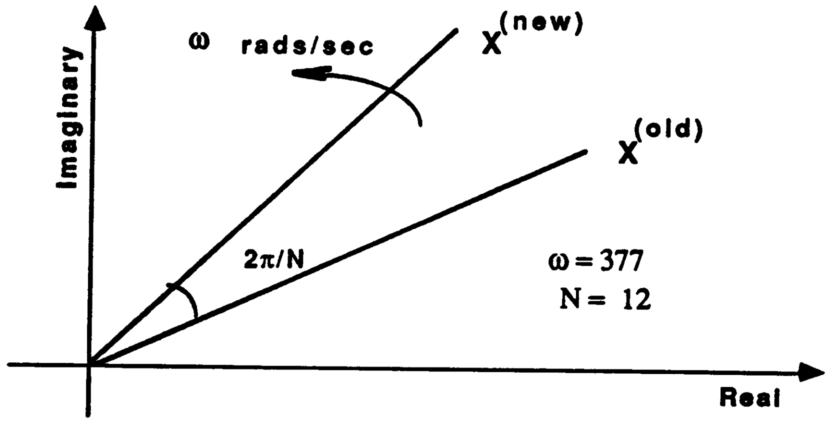


FIG 5.3 : PHASORS FROM DIFFERENT DATA WINDOWS

Equations 5.6 and 5.7 are used in a Fortran program which calculates the phasors in the first sample window. Fig 5.1 represents this calculation diagramatically. The same method could be used to calculate the phasors in the subsequent windows too. But, considerable speed improvement is obtained by the use of recursion. The following section looks into this aspect in detail.

5.3 RECURSIVE vs NON-RECURSIVE CALCULATIONS

In fig 5.2 the signal of equation 5.2.1 is given. 'Window-1' produces the set of samples $\{x_k,\ k=0..11\}$. The phasor representation for this window is given by equation 5.2.7. A new sample is obtained after an interval $T_s = 2\pi/N = \pi/6$ radians and the data window moves by this interval to 'window-2' with the sample set $\{x_k,\ k=1,..N\}$. The phasor computations using data window-2 are performed using equations 5.1, 5.2 and 5.7 as follows:

$$x(t)= \sqrt{2}\ \tilde{X}\ \sin(wt +\phi+2\pi/N) \text{------------------} \qquad (5.8)$$

$$\therefore\ X^{(new)}= X\ e^{j(\phi+2\pi/N)}$$

$$= X^{(old)}\, e^{j2\pi/N} \text{------------------------} \qquad (5.9)$$

The use of Equation 5.9 for calculating the filtered phasor of a signal produces a phasor which rotates in a counter clockwise direction in the complex plane by the sampling angle of $2\pi/N$ rads. The phasor rotates at an angular velocity of $120\pi = 377$ rads/sec in the case of a 60 Hz. system. This has been represented in fig 5.3.

In general, when sample data from the r^{th} window is used we have,

$$X_c^{(r)} = 2/N \sum_{k=0}^{N-1} [x_{k+r-1} \cos(2\pi/N)k] \text{------------------} \qquad (5.10)$$

$$X_s^{(r)} = 2/n \sum_{k=0}^{N-1} [\, x_{k+r-1} \sin(2\pi/N)k] \text{ ----------------} \qquad (5.11)$$

$$X^{(r)} = 1/\sqrt{2}\ (\, X_s^{(r)} + X_c^{(r)}\,) \text{ ---------------------} \qquad (5.12)$$

The procedure as given above, for the calculation of $X^{(r)}$ is non-recursive, and requires 2N multiplications and 2(N-1) additions. However, it should be noted that in progressing from one data window to the next, only one sample is discarded and one sample added in its stead. It would be hence advantageous to develop a technique that retains 2(N-1) multiplications and 2(N-1) sums corresponding to that portion of the data which is common to both the old and new data windows.

It can be shown [1] that the phasor corresponding to the r^{th} window is related to that corresponding to the $(r-1)^{th}$ window by

$$X^{(r)} = X^{(r-1)} + j\,(1/6)\,(1/\sqrt{3})\,[x_{N+r} - x_r] \qquad (5.13)$$

This is a recursive relationship and only two multiplications need be performed at each new sample time, making this a very efficient computational algorithm.

In this thesis, the objective is to obtain positive sequence phasors in each of the phases of a three phase transmission line of the voltages on the line-side and source side, as well as the currents flowing in each phase at specified instants of time to be input to a Computer based reclosing relay. As given in section 3.1, the EMTP outputs the values of the signals at the rate of once every 4.62 milli- seconds. The program written to calculate the phasors, samples the output at the rate of 12 times in a cycle, ie, $N = 12$ [5] .

The program listing is given in appendix-I. Examples of some phasor calculations in selected EMTP simulation cases are given in Table 3.1.

CHAPTER 6

THE RECLOSING ALGORITHM

6.1 CIRCUIT BREAKER RECLOSING LOGIC

The logic which was used in the reclosing of the Circuit Breaker after it was opened in the event of a fault made the following assumptions :

(a) There was to be no reclosing on to a three phase fault.

(b) The circuit breaker is capable of closing individually in each phase.

(c) In the case of a line to ground fault, reclosing is started (that is, the 'close coil' is energized first) on the pole in the phase farthest from the phase on which the fault was detected. The reason for this being that the first fault produces an arc which might spread to the nearby conductors, involving them in the fault.

(d) In case both of the unfaulted phases are at the same distance from the faulted phase in a line to ground fault then, reclosing could be started at either of these phases.

(e) In the case of a line to line fault, reclosing starts on the remaining unfaulted phase.

(f) Voltage phasors are used to detect whether a fault is present on the transmission line. The phasors are obtained in an associated program which uses the logic described in the previous chapter.

(g) The phasors were obtained from filtered samples of EMTP simulation outputs.

(h) If the reclosing algorithm detects any fault on the line, reclosing is aborted in all the poles of the circuit breaker, temporarily for 'multiple shot' and permanently for a 'one shot' reclosing relay.

The method of deciding the phase on which reclosing is to start depends on whether the line is transposed or untransposed.

6.1.1 Reclosing logic for Untransposed lines: In this case, the lines are all at equal distances with respect to each other from end to end, as shown in fig 3.2 (a). Hence, if a line to ground fault were to occur in phase B, reclosing could be started in either A or C. But if the fault occurs on phase A then, according to the logic given above, reclosing would start on phase C and not on phase B.

6.1.2 Reclosing logic for Transposed lines: The case of transposed lines is different. Here, at various points along the transmission line, the distance between lines (phases) is deliberately made to vary, as shown in fig 3.2 (b). Hence, for a L-G fault on phase B, it does not matter whether we first close on phase A or C if the fault occurs in region-1. But, since in region-2, A and C are at different distances from B, the closing sequence could make a difference.

This means that we would need to know not only the type of fault but also its distance from a given end in the case of transposed lines. This would be the case in a real time situation since the 'Protection Relay' would be supplying the information to the reclosing relay. But since the algorithm described has not been written for a real time case, a simple 'Distance Relay' has been incorporated into the program which obtains the distance and the type of the fault. A flow chart of this algorithm is shown in fig 6.6.

6.2 DISTANCE RELAY FORMULAE

Let the transmission line have a positive sequence impedance of 'Z1' and a zero sequence impedance of 'Z0'. Then the positive sequence impedances between the various phases to neutral from the end at which the voltages and currents are being measured are given by

$$ZA = (VA*Z1) / [Z1*IA+((Z0-Z1)*I0]$$

$$ZB = (VB*Z1) / [Z1*IB+((Z0-Z1)*I0] \quad \text{----------------} \ (6.1)$$

$$ZC = (VC*Z1) / [Z1*IC+((Z0-Z1)*I0]$$

The positive sequence impedance between the various phases to each other are given by,

$$ZAB = ((VA-VB)*Z1) / [Z1*(IA-IB)]$$

$$ZAC = ((VA-VC)*Z1) / [Z1*(IA-IC)] \quad \text{-----} \ (6.2)$$

$$ZBC = ((VB-VC)*Z1) / [Z1*(IB-IC)]$$

Therefore, we have the distance of any phase to neutral (ground) connection on the line on a phase 'p' given by,

$$DISTP = ZPM/ Z1M= MAG\ (\ VP/ [IP+M*I0]\ )\text{----------}(6.3)$$

and that of any phase to phase connection between phases 'p1' and 'p2' given by

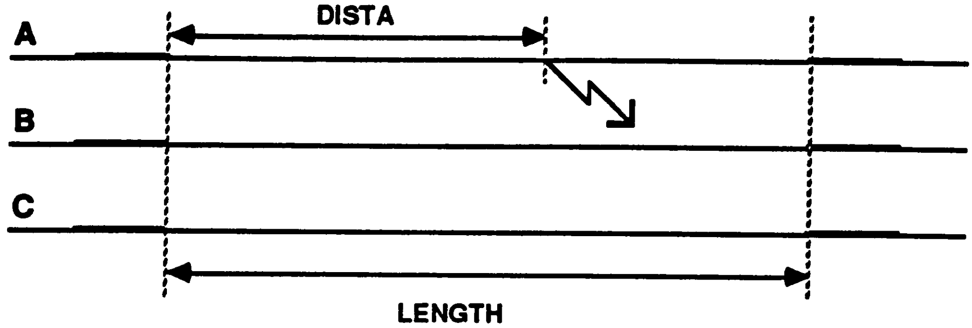
$$DISTP1P2 = ZP1P2M/ Z1M$$

$$= MAG(\ (VP1-VP2) / [(IP1-IP2)]\ \text{----}(6.4)$$

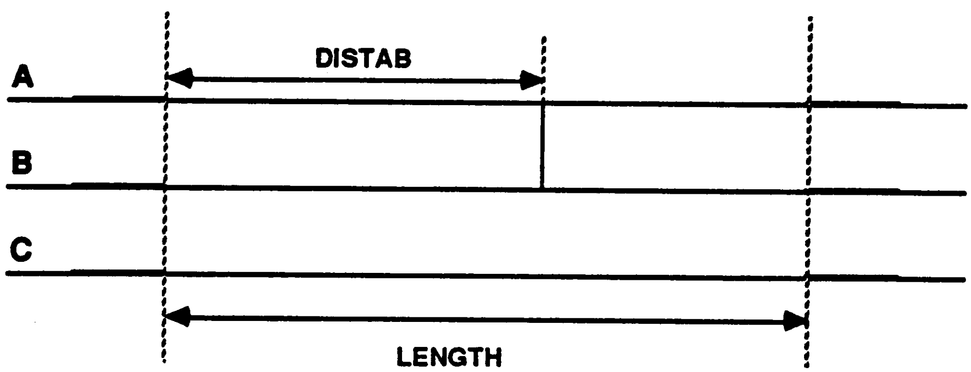
where,

$$M=(Z0-Z1) / Z1$$

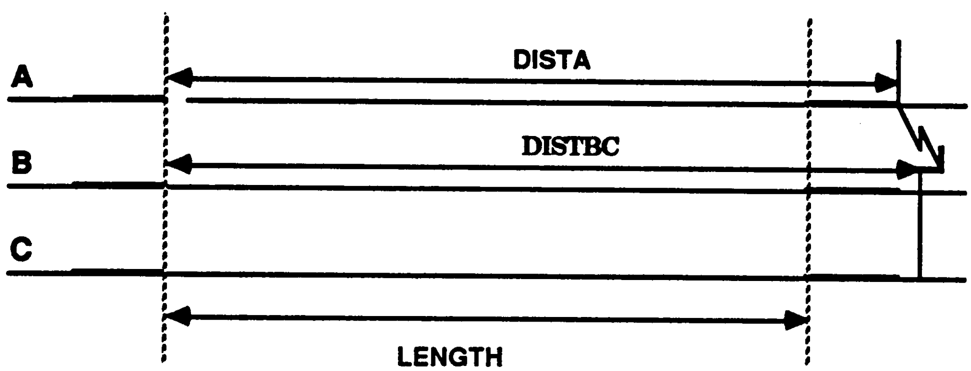
In the simulations which were made in this research,



(a) L-G fault on A



(b) L-L fault on A-B



(c) NO FAULT ON THE LINE

Fig 6.1: ILLUSTRATION OF WORKING OF DISTANCE RELAY

$Z1 = (0.022 + j\,0.59198)\ \Omega = 1.01224 \times 10^{-4} \angle 87.87$ p.u.

$Z0 = (.51631 + j1.5891)\ \Omega = 2.85 \times 10^{-4} \angle 71.95$ p.u.

$M = 1.879 \angle -23.1.$

We would be able to detect that there has been a fault from p-g or p1-p2 in the transmission line under consideration, if DISTP or DISTP1P2 respectively, were less than the length of the line which is in the zone of control of the reclosing relay as illustrated in fig 6.1 (a) and fig 6.1 (b). If on the other hand, DISTP and DISTP1P2 were greater than the length of the line in the case of all the phases, as shown in fig.6.1(c), then we can come to the conclusion that no fault exists on the line under consideration. A fault may exist in the next zone of protection but, that will be taken care of by the reclosing relay within whose zone of protection it occurs.

6.3 METHOD OF DETECTING FAULTS IN THE POST RECLOSE PERIOD

The logic used in the actual reclosing relay algorithm is very simple. The reclosing relay obtains the information on the type of the fault at the time before tripping from the distance relay(which in a real-time situation could be a Symmetrical Component Distance Relay (SCDR) [12], [14]). Using this information, the relay starts reclosing on a pole of the circuit breaker based on the rules given in sect 6.1. If reclosing takes place on to a faulted phase, the current flowing through the phase is larger than a preset amount and the closing is aborted on all phases (permanently, in case of one-shot relays and temporarily in case of multiple shot relays). If it is determined that the reclosing phase is normal then, the following logic is used to determine whether the rest of the system is faulted or normal:

6.3.1 Untransposed lines: In the case of a L-G fault, the faulted phase has a voltage phasor of magnitude close to 0.0 volts p.u. whereas, the other unenergized phase has a voltage of magnitude around 0.03-0.04 volts p.u.(in the case where the line is not compensated). Hence if we detect such a condition, we would know that the fault still persists. In the case of 'one shot' relays where we are allowed only a single attempt at reclosing, we would completely abort reclosing at this stage whereas, in 'multiple shot' relays we would abort reclosing for the time being and go through the process enumerated above after a previously fixed time has elapsed. We would continue in this manner till either the system returns to normal or the prescribed number of attempts (shots) have been completed, A 1.0

B 0.18

C 0.08

(a) NORMAL CASE

A 1.0

B 0.0

C 0.04

(b) GROUND FAULT ON B

A 1.0

B 0.11

C 0.11

(c) L-L FAULT ON B-C

FIG 6.2 : SIMULATIONS ON AN UNTRANSPOSED
3 PHASE TRANSMISSION LINE

whichever is earlier.

In the case of a L-L fault, the difference between the phasors on both the faulted lines is close to 0.0 and this would be an indication for either temporarily or permanently aborting reclosing. In the case where the other two phases are normal, their voltages are 0.8- 0.18 volts p.u. The voltage relationships are for an untransposed line are illustrated in fig 6.2. for the case where phase A alone is energized.

6.3.2 Transposed lines: The logic for detection of L-G faults is the same in this case as in the case of untransposed lines. But in the case of L-L faults, the detection method becomes a little different. In the case of transposed lines, when one of the poles is reclosed , the other two phases have the same magnitude as well as phase angle irrespective of whether the system is normal or whether a L-L fault is present. The voltage relationships for a transposed line are illustrated in fig.6.3 for the case where phase A alone is energized. Hence, to detect a L-L fault, we wait for the closing of one more pole at which time if:

(a) The phasors measured on the previously faulted lines are different then,we know that the fault has cleared

A 1.0

B 0.11

C 0.11

(a) NORMAL CASE

A 1.0

B 0.0

C 0.05

(b) GROUND FAULT ON B

A 1.0

B 0.11

C 0.11

(c) L-L FAULT ON B-C

FIG 6.3 : SIMULATIONS ON A TRANSPOSED 3 PHASE TRANSMISSION LINE

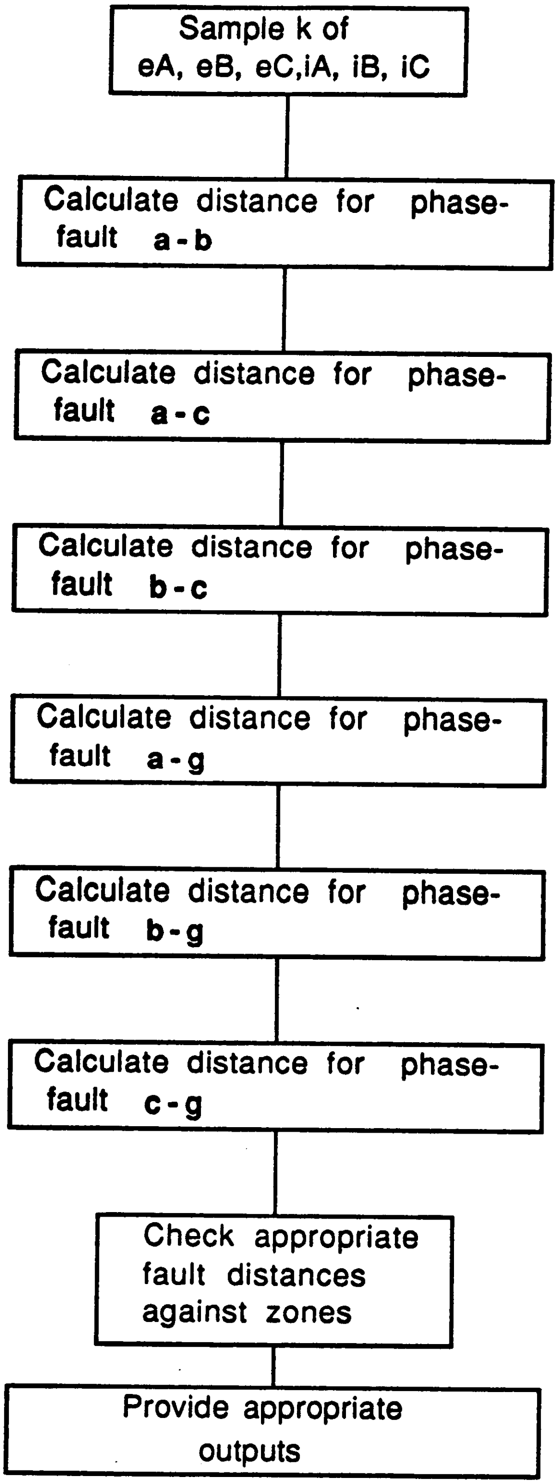


fig 6.4 : FLOW CHART FOR A SIMPLE DISTANCE RELAY

(b) If the phasors are still equal then the fault still persists.

Due to the fact that for the the detection of the 'No Fault' and L-L fault cases, we have to wait for another pole to close, the time necessary in these cases cannot be predetermined as the time needed for the second pole to reclose would depend on the design of the circuit breaker (this is noted in table 3.1 also).

The program which uses the logic described above is given in the Appendix-I. It has been written to be as general as possible and covers the cases which have been discussed thus far and those listed in table 3.1 . In a normal online setting, the relay engineer would input parameters, thus giving the relay, information about the system (whether the line is transposed or untransposed, whether the line has been taken out for maintenance, the maximum allowable fault current etc.). In the present case, the program being off-line, it queries the terminal a predetermined set of questions. A summary of

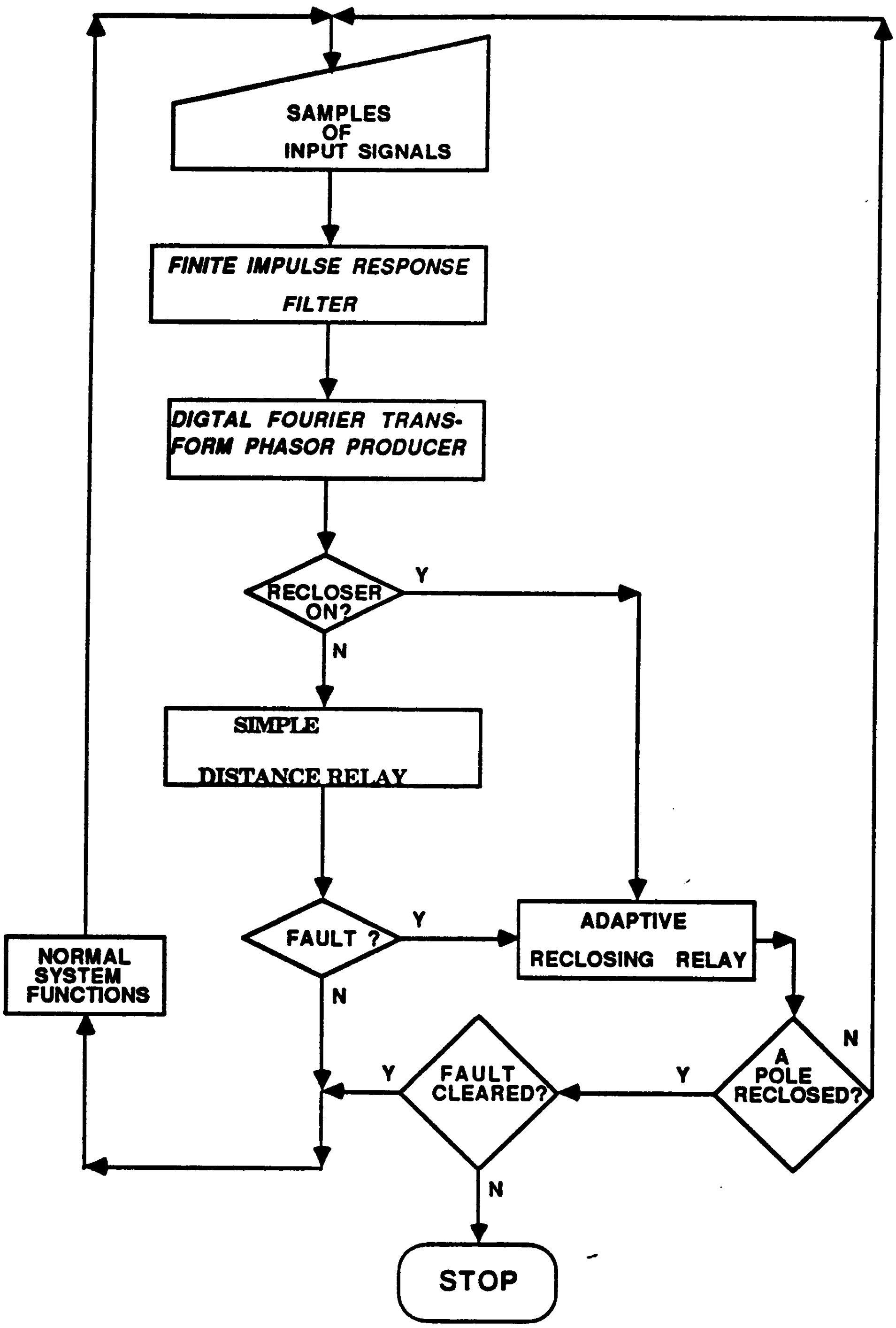


FIG 6.5: FLOWCHART OF THE PROTECTIVE RELAY SYSTEM

the program, in flowchart form has been illustrated in Fig 6.5.

CONCLUSIONS

(1) In recent years, there have been a spate of papers in the literature on the topic of Adaptive Relaying, an issue that has been mentioned in Section 1.2. Integration of a Computer Reclosing Relay with the rest of the components of an 'Adaptive Protective System', becomes a very attractive idea. This Thesis is at least a step towards showing that the idea is not only attractive but, feasible too.

(2) The systems which have been simulated may seem simple in their basic form (figure 3.1). But it is general in the sense that we have what could be considered as the Thevenin equivalent of the system connected to either end. It can be inferred from the significant number of successful cases which have been studied in this Thesis in which the reclosing relay algorithm has detected the presence of a fault after the circuit breaker has reclosed that a Computer Reclosing Relay is very feasible.

(3) Due to the simplicity of the reclosing program, the speed of the recloser is a little greater than a cycle except in the 'No Fault' and 'L-L' cases, where the total decision time depends on the design of the reclosing relay and the circuit breaker. The only reason why the speed is not higher is because the phasor calculation utilizes a one cycle window. This means that the recloser has to wait for a period of 1 cycle (or 12 samples) which is when the phasor calculation becomes reliable. The length of the program code (listing given in the appendix) itself is due to the fact that the program had to be made as general as possible. In actual practice, the length could be reduced depending on the system conditions.

(4) In this thesis, Computer Reclosing Relaying as it pertains to transmission lines is the only area of the power system which has been studied. The author would suggest that adaptive relaying in other areas, for example, transformer protection, could be good areas for future research.

REFERENCES

1. G.D.Rockefeller, High Speed Distance Relaying using a digital computer, **IEEE PAS-91**, pp. 1244-1258, May, 1972.

2. A.G.Phadke, T.Hlibka, M.Ibrahim, A Digital Computer Relay for EHV systems, **IEEE PAS-95**, pp. 291-301, January/February, 1976.

3. W.D.Breingan, M.M.Chen, Laboratory Investigation of a Digital system for protection of transmission lines, **IEEE PES**,Winter meeting,1977.

4. A.G.Phadke, J.S.Thorp, M.G.Adamiak, Microprocesssor based Ultra High Speed Relay, **IEEE PAS-100**, Vol.4, pp. 2026-2032, April 1981.

5. A.G.Phadke, J.S.Thorp, M.G.Adamiak, A new measuring technique for tracking voltage phasors, local system frequency and rate of change of frequency., **IEEE PAS- 102**, Vol.5, pp. 1025-1034, May1983.

6. A.G.Phadke, J.S.Thorp, S.H.Horowitz, Impact of Adaptive Protection on Power System Protection and control, Report on Research sponsored by the U.S Department Of Energy.

7. A.G.Phadke, J.S.Thorp, S.H.Horowitz, Adaptive Transmission System Relaying, **IEEE PES,** Summer meeting, 1987.

8. J.Lewis Blackburn, Protective Relaying Principles And Applications (Book), Marcel Dekker,1987, pp. 491-492.

9. Electromagnetic Transients program (EMTP) Primer, EPRI final report, September 1985.

10. William D.Stanley, Gary R.Dougherty, Ray Dougherty, Digital Signal Processing (Book), Prentice Hall, 1984, pp. 82-83.

11. A.G.Phadke, M.G.Adamiak, T.Hlibka, M.Ibrahim, A Microcomputer symmetrical component distance relay, **Proceedings of PICA,1979.**

12. K.R.Shah, E.R.Detjen, A.G.Phadke, Feasibilty of an Adaptive Distribution Protection System using computer over- current

relaying concepts, Research sponsored by U.S Department Of Energy.

13. A.G.Phadke, T.Hlibka, M.Ibrahim,Fundamental Basis For Distance Relaying with symmetrical components, **IEEE, PAS-96**,Vol.no. 2, pp. 635-643, March/April,1977.

14. E.Oran Brigham, Fast Fourier Transforms (book), Prentice-Hall, 1974, pp. 80-86.

15. A.G.Phadke, J.S.Thorp, Computer relaying for Power Systems (book), RSP,John Wiley & Sons,1988, pp. 1-270.

16. J,W. Cooley, J.W. Tukey, An algorithm for the machine computation of complex, Math computation, 1965, pp. 297-301.

APPENDIX-I
PROGRAMS

```
C_______________________________________________________________________
C    THIS PROGRAM FILTERS AN INPUT WAVE FORM USING SAMPLED-DATA.
C         THE WAVEFORM IS SAMPLED ONCE EVERY 10 DEGREES
C                FORTRAN PROGRAM LISTING
C_______________________________________________________________________
      PARAMETER NFMAX=3600
C NFMAX IS THE MAXIMUM LENGTH OF THE FILTER
      PARAMETER LFAC=100
C LFAC IS THE NO. OF TENTHS OF A DEGREE BETWEEN SAMPLES IN INPUT
      PARAMETER FSAMP=2164.5
C FSAMP IS THE INPUT SAMPLING FREQUENCY
      PARAMETER NSIZE=2000
C NSIZE IS THE MAXIMUM LENGTH OF THE INPUT DATA STRING
      DIMENSION HFIL(NFMAX),ZTD1(NFMAX),ZTD2(NFMAX),ZTD3(NFMAX),

     1          ZTD4(NFMAX),ZTD5(NFMAX),ZTD6(NFMAX),ZTD7(NFMAX),
     1          ZTD8(NFMAX),ZTD9(NFMAX)
      REAL TERA(NSIZE),TERB(NSIZE),TERC(NSIZE),BEGA(NSIZE),BEGB(NSIZE)
      REAL BEGC(NSIZE),IA(NSIZE),IB(NSIZE),IC(NSIZE)
      CHARACTER*1 JUNK(6)
      DATA N0,ZOUT1,ZOUT2,ZOUT3,ZOUT4,ZOUT5,ZOUT6,ZOUT7,ZOUT8,
     1                                      ZOUT9/10*0.0/
C       BEGIN MAIN PROGRAM
C THE FOLLOWING IS NECESSARYTO FILTER THE RELEVENT DATA FROM THE REST
C   OF THE JUNK IN THE EMTP OUTPUT
C     BEGIN JUNKING
        DO 10 I =1,100
        READ(5,30) (JUNK(J),J=1,6)
        IF(JUNK(1).EQ.'+'.AND.JUNK(2).EQ.'M'.AND. JUNK(3).EQ.'I'.
     *  AND. JUNK(4).EQ.'S'.AND.JUNK(5).EQ.'C'.AND.JUNK(6).EQ.'.')
     *  GO TO 20
10      CONTINUE
20      BACKSPACE 5
30      FORMAT(6A1)
        DO 50 I =1,1000
        READ(5,30) (JUNK(J),J=1,6)
        IF(JUNK(1).EQ.' '.AND.JUNK(2).EQ.' '.AND. JUNK(3).EQ.'S'.
     *  AND. JUNK(4).EQ.'T'.AND.JUNK(5).EQ.'E'.AND.JUNK(6).EQ.'P')
     *  GO TO 60
50      CONTINUE
60      BACKSPACE 5
        READ(5,30) (JUNK(J),J=1,6)
        READ(5,30) (JUNK(J),J=1,6)
C   END JUNKING
        READ(7,*)NA,NB
C THE DESIRED IMPULSE RESPONSE OF THE FILTER IS STORED IN THE FILE WITH
C  UNIT NO.= 7
        IF(NB.LE.NFMAX) GO TO 6
        WRITE(*,*) 'DECIMATION FILTER TOO LONG'
        STOP
6       NBF=NB/LFAC
        IF(NB.EQ.NBF*LFAC) GO TO 105
        WRITE(*,*)'FILTER LENGTH INDIVISIBLE BY LFAC'
        STOP
105     READ(7,*)(HFIL(JJ),JJ=1,NB)
C       ITIME GIVES WRITE TOTAL NO. OF SAMPLES TO BE PROCESSED
        ITIME=432
        DO 11 JJ=1,ITIME
150     CONTINUE
        READ(5,100,ERR=150,END=1) TERA(JJ),TERB(JJ),TERC(JJ),BEGA(JJ),

     1          BEGB(JJ), BEGC(JJ),IA(JJ),IB(JJ),IC(JJ)
```

```
11      CONTINUE
100     FORMAT(15X,9E13.6)
1       IPTR=1
C       DRATE GIVES ENTER DESIRED PROCESSING RATE
31      DRATE=2164.5
        MFAC=IFIX(FSAMP*LFAC/DRATE)
        IF(MFAC*DRATE.EQ.FSAMP*LFAC) GO TO 40
        WRITE(*,*)'RATE IS UNACHIEVABLE-TRY AGAIN'
        GO TO 31
40      WRITE(*,*)'INTERPOLATION FACTOR=',LFAC
        WRITE(*,*)'DECIMATION FACTOR=',MFAC
        WRITE(9,*)'              FILTERED SAMPLE-DATA'
        WRITE(10,*)'             FILTERED SAMPLE-DATA'
        WRITE(11,*)'             FILTERED SAMPLE-DATA'
        WRITE(9,*)
        WRITE(10,*)
        WRITE(11,*)
        WRITE(9,*)' TIME         BEGA          BEGB          BEGC'
        WRITE(10,*)' TIME         TERA          TERB          TERC'
        WRITE(11,*)' TIME         IA            IB            IC'
        DO 500 I=1,ITIME
        DT=(I-1)/FSAMP
        WRITE(2,110) DT,X1,X2,X3,X4,X5,X6,X7,X8,X9
110     FORMAT(F8.5,9(1X,E12.5))
        X1=TERA(IPTR)
        X2=TERB(IPTR)
        X3=TERC(IPTR)
        X4=BEGA(IPTR)
        X5=BEGB(IPTR)
        X6=BEGC(IPTR)
        X7=IA(IPTR)
        X8=IB(IPTR)
        X9=IC(IPTR)
        IPTR=IPTR+1
        DO 120 J=1,NBF-1
        INDX=NBF+1-J
        ZTD1(INDX)=ZTD1(INDX-1)
        ZTD2(INDX)=ZTD2(INDX-1)
        ZTD3(INDX)=ZTD3(INDX-1)
        ZTD4(INDX)=ZTD4(INDX-1)
        ZTD5(INDX)=ZTD5(INDX-1)
        ZTD6(INDX)=ZTD6(INDX-1)
        ZTD7(INDX)=ZTD7(INDX-1)
        ZTD8(INDX)=ZTD8(INDX-1)
120     ZTD9(INDX)=ZTD9(INDX-1)
        ZTD1(1)=X1
        ZTD2(1)=X2
        ZTD3(1)=X3
        ZTD4(1)=X4
        ZTD5(1)=X5
        ZTD6(1)=X6
        ZTD7(1)=X7
        ZTD8(1)=X8
        ZTD9(1)=X9
        N0=N0+LFAC
        IF(N0.LT.MFAC) GO TO 500
        N0=N0-MFAC
        DO 130 J=1,NBF
        INDX=J*LFAC-N0
        ZOUT1=ZOUT1+HFIL(INDX)*ZTD1(J)
```

```
      ZOUT2=ZOUT2+HFIL(INDX)*ZTD2(J)
      ZOUT3=ZOUT3+HFIL(INDX)*ZTD3(J)
      ZOUT4=ZOUT4+HFIL(INDX)*ZTD4(J)
      ZOUT5=ZOUT5+HFIL(INDX)*ZTD5(J)
      ZOUT6=ZOUT6+HFIL(INDX)*ZTD6(J)
      ZOUT7=ZOUT7+HFIL(INDX)*ZTD7(J)
      ZOUT8=ZOUT8+HFIL(INDX)*ZTD8(J)
130   ZOUT9=ZOUT9+HFIL(INDX)*ZTD9(J)
      ZOUT1=ZOUT1/FSAMP
      ZOUT2=ZOUT2/FSAMP
      ZOUT3=ZOUT3/FSAMP
      ZOUT4=ZOUT4/FSAMP
      ZOUT5=ZOUT5/FSAMP
      ZOUT6=ZOUT6/FSAMP
      ZOUT7=ZOUT7/FSAMP
      ZOUT8=ZOUT8/FSAMP
      ZOUT9=ZOUT9/FSAMP
C ZOUTX STANDS FOR THE FILTERED OUTPUT OF INPUT NO.= X
      WRITE(3,110)DT,ZOUT1,ZOUT2,ZOUT3,ZOUT4,ZOUT5,ZOUT6,ZOUT7,
     1                                            ZOUT8,ZOUT9
      WRITE(9,140)DT,ZOUT1,ZOUT2,ZOUT3
      WRITE(10,140)DT,ZOUT4,ZOUT5,ZOUT6
      WRITE(11,140)DT,ZOUT7,ZOUT8,ZOUT9
140   FORMAT(F8.5,3(3X,E12.5))
500   CONTINUE
      STOP
      END
```

FILTER RESPONSE

```
      DIMENSION H(3600)
C     MAIN PROGRAM
      S1=394.
      S2=2620.
      G60=(SQRT((S1**2+(377.)**2)*(S2**2+(377.)**2)))/(S1*S2)
      C1=G60*S1*S2/(-S1+S2)
      C2=G60*S1*S2/(S1-S2)
      WRITE(7,*) 1,3600
      DO 100 I=1,3600
      DT=(I-1)/216000.
      H(I)=C1*EXP(-DT*S1)+C2*EXP(-DT*S2)
      WRITE(7,*) H(I)
100   CONTINUE
      STOP
      END
```

```
C
C  THE FOLLOWING FORTRAN PROGRAM CALCULATES THE PHASOR OF THE VOLTAGES AND
C    CURRENTS OBTAINED AFTER FILTERING THE EMTP OUTPUT
C
C MAIN PROGRAM
      REAL TERA(0:11),TERB(0:11),TERC(0:11),BEGA(0:11),BEGB(0:11)
      REAL BEGC(0:11),IA(0:11),IB(0:11),IC(0:11),TIME(0:11)
      REAL TERAR,TERAIM,TERBR,TERBIM,TERCR,TERCIM,BEGAR
      REAL BEGAIM,BEGBR,BEGBIM,BEGCR,BEGCIM,IAR,IAIM,IBR,IBIM,ICR,ICIM
      REAL BEGAP,BEGAM,BEGBP,BEGBM,BEGCP,BEGCM,IAP,IAM,IBP,IBM,ICP
      REAL ICM,TERAM,TERAP,TERBM,TERBP,TERCP,TERCM,NTIME
      REAL DTERA,DTERB,DTERC,DBEGA,DBEGB,DBEGC,DIA,DIB,DIC
      INTEGER OLD
      CHARACTER*132 GARB
      COMMON THETA,N
C BEGIN MAIN PROGRAM
        N=11
        THETA=0.5235987
C NDATA STANDS FOR THE TOTAL NO: OF DATA POINTS
        NDATA=144
        WRITE(1,*)'              PHASORS OBTAINED FROM SAMPLED-DATA'
        WRITE(2,*)'              PHASORS OBTAINED FROM SAMPLED-DATA'
        WRITE(8,*)'              PHASORS OBTAINED FROM SAMPLED-DATA'
        DO 10 I=0,N
        READ(3,*) TIME(I),TERA(I),TERB(I),TERC(I),BEGA(I),
     1            BEGB(I),BEGC(I),IA(I),IB(I),IC(I)
        READ(3,200)GARB
        READ(3,200)GARB
C THE FILTER OUTPUTS DATA EVERY 10 DEGREES IN A CYCLE .
C FOR THE PHASOR CALCULATION WE NEED TO SAMPLE ONLY ONCE IN 30 DEGREES.
200     FORMAT(1A132)
10      CONTINUE
C SUBROUTINE 'INCLC' DOES THE INITIAL PHASOR CALCULATION FOR THE FIRST
C                                                  TWELVE DATA POINTS
        CALL INCLC(TERA,TERAR,TERAIM)
        CALL INCLC(TERB,TERBR,TERBIM)
        CALL INCLC(TERC,TERCR,TERCIM)
        CALL INCLC(BEGA,BEGAR,BEGAIM)
        CALL INCLC(BEGB,BEGBR,BEGBIM)
        CALL INCLC(BEGC,BEGCR,BEGCIM)
        CALL INCLC(IA,IAR,IAIM)
        CALL INCLC(IB,IBR,IBIM)
        CALL INCLC(IC,ICR,ICIM)
        OLD=0
        WRITE(1,190)
        WRITE(2,191)
        WRITE(8,192)
190     FORMAT(2X,'TIME',12X,'TER1A',15X,'TER1B',15X,'TER1C')
191     FORMAT(2X,'TIME',12X,'BEGA',15X,'BEGB',15X,'BEGC')
192     FORMAT(2X,'TIME',14X,'I1A',16X,'I1B',16X,'I1C')
        WRITE(1,290)
        WRITE(2,290)
        WRITE(8,290)
290     FORMAT(1X,'(SECS)',8X,3('MAGN',4X,'ANGLE',7X))
        DO 30 I=1,NDATA
         NTIME=TIME(OLD)
C SUBROUTINE 'RTOP' CONVERTS FROM RECTANGULAR TO POLAR FORM
         CALL RTOP(TERAR,TERAIM,TERAM,TERAP)
         CALL RTOP(TERBR,TERBIM,TERBM,TERBP)
         CALL RTOP(TERCR,TERCIM,TERCM,TERCP)
```

```
      CALL RTOP(BEGAR,BEGAIM,BEGAM,BEGAP)
      CALL RTOP(BEGBR,BEGBIM,BEGBM,BEGBP)
      CALL RTOP(BEGCR,BEGCIM,BEGCM,BEGCP)
      CALL RTOP(IAR,IAIM,IAM,IAP)
      CALL RTOP(IBR,IBIM,IBM,IBP)
      CALL RTOP(ICR,ICIM,ICM,ICP)
C SUBROUTINE 'VPUCON' CONVERTS THE VOLTAGE PHASORS INTO PER-UNIT
      CALL VPUCON(TERAM,TERBM,TERCM,BEGAM,BEGBM,BEGCM)
C SUBROUTINE 'IPUCON' CONVERTS THE CURRENT PHASORS INTO PER-UNIT
      CALL IPUCON(IAM,IBM,ICM)
      IF (NTIME.LE.0.03) GO TO 100
      WRITE(1,120)NTIME,TERAM,TERAP,TERBM,TERBP,TERCM,TERCP
      WRITE(2,120)NTIME,BEGAM,BEGAP,BEGBM,BEGBP,BEGCM,BEGCP
      WRITE(8,120)NTIME,IAM,IAP,IBM,IBP,ICM,ICP
120   FORMAT(F8.5,3(1X,F10.2,1X,F8.2))
C THE NEXT DATA POINT IS NOW READ IN.THE PHASOR IS NOW CALCULATED
                                                     RECURSIVELY
C
100   READ(3,*,END=600)NTIME,TERAN,TERBN,TERCN,BEGAN,
     1                           BEGBN,BEGCN,IAN,IBN,ICN
      READ(3,200,END=600)GARB
      READ(3,200,END=600)GARB
      DTERA=TERAN-TERA(OLD)
      DTERB=TERBN-TERB(OLD)
      DTERC=TERCN-TERC(OLD)
      DBEGA=BEGAN-BEGA(OLD)
      DBEGB=BEGBN-BEGB(OLD)
      DBEGC=BEGCN-BEGC(OLD)
      DIA=IAN-IA(OLD)
      DIB=IBN-IB(OLD)
      DIC=ICN-IC(OLD)
C SUBROUTINE 'NPHASOR' CALCULATES THE NEW PHASOR RECURSIVELY
      CALL NPHASOR(TERAR,TERAIM,DTERA,OLD)
      CALL NPHASOR(TERBR,TERBIM,DTERB,OLD)
      CALL NPHASOR(TERCR,TERCIM,DTERC,OLD)
      CALL NPHASOR(BEGAR,BEGAIM,DBEGA,OLD)
      CALL NPHASOR(BEGBR,BEGBIM,DBEGB,OLD)
      CALL NPHASOR(BEGCR,BEGCIM,DBEGC,OLD)
      CALL NPHASOR(IAR,IAIM,DIA,OLD)
      CALL NPHASOR(IBR,IBIM,DIB,OLD)
      CALL NPHASOR(ICR,ICIM,DIC,OLD)
      TERA(OLD)=TERAN
      TERB(OLD)=TERBN
      TERC(OLD)=TERCN
      BEGA(OLD)=BEGAN
      BEGB(OLD)=BEGBN
      BEGC(OLD)=BEGCN
      IA(OLD)=IAN
      IB(OLD)=IBN
      IC(OLD)=ICN
      TIME(OLD)=NTIME
      OLD=OLD+1
      IF (OLD.EQ.12)OLD=0
 30   CONTINUE
600   STOP
      END
C END MAIN PROGRAM
      SUBROUTINE NPHASOR(ER,EIM,DE,OLD)
       COMMON THETA,N
       REAL ER,EIM,DE
       INTEGER OLD
```

```
      ER=ER+(((DE)*COS(OLD*THETA))/6.)
      EIM=EIM-(((DE)*SIN(OLD*THETA))/6.)
      RETURN
     END
     SUBROUTINE INCLC(E,ER,EIM)
      COMMON THETA,N
      REAL ER,EIM
      REAL E(0:11)
      ER=0.0
      EIM=0.0
      DO 40 K=0,N
       ER=ER+((E(K)*COS(K*THETA))/6.)
       EIM=EIM-((E(K)*SIN(K*THETA))/6.)
40    CONTINUE
      RETURN
     END
     SUBROUTINE RTOP(ER,EIM,EM,EP)
      REAL ER,EIM,EM,EP
      EPSILON=1.0E-4
      AER=ABS(ER)
      AEIM=ABS(EIM)
      IF((AER.LE.EPSILON).AND.(AEIM.LE.EPSILON)) GO TO 50
      EM=SQRT((ER*ER)+(EIM*EIM))
      EP=ATAN2D(EIM,ER)
      RETURN
50    EP=0.0
      EM=0.0
      RETURN
     END
    SUBROUTINE VPUCON(TERAM,TERBM,TERCM,BEGAM,BEGBM,BEGCM)
C   THIS SUBROUTINE IS FOR CONVERSION OF VOLTAGE INTO P.U
C    ASSUMING A BASE OF 765KV(L-L)AND 100MVA
     TERAM=TERAM/624619.0
     TERBM=TERBM/624619.0
     TERCM=TERCM/624619.0
     BEGAM=BEGAM/624619.0
     BEGBM=BEGBM/624619.0
     BEGCM=BEGCM/624619.0
     RETURN
    END
    SUBROUTINE IPUCON(IAM,IBM,ICM)
C   THIS SUBROUTINE IS FOR CONVERSION OF THE CURRENTS INTO P.U
C    ASSUMING A BASE OF 765KV(L-L)AND 100MVA
     REAL IAM,IBM,ICM
     IAM=IAM/(213.4632)
     IBM=IBM/(213.4632)
     ICM=ICM/(213.4632)
     RETURN
    END
```

```
C     BEGINNING OF THE MAIN PROGRAM
      COMMON TIME,KFAULT,ERR1,ERR2,LINE1,LINE2,ANSWER1
      REAL BEGAM,IAM,IAP,IBM,IBP,ICM,ICP,IFAR,IFAIM,IFAM,IFAP
      REAL IFBR,IFBIM,IFBM,IFBP,IFCR,IFCIM,IFCM,IFCP
      REAL Z1M,Z1P,MM,MP,BEGAP,BEGBM,BEGBP,IAR,IAIM
      REAL BEGCM,BEGCP,IBR,IBIM,ICR,ICIM,I0R,I0IM,I0M,I0P
      REAL MI0R,MI0IM,MI0M,MI0P,ZAM,ZAP,ZBM,ZBP,ZIM,ZIP
      REAL DISTA,DISTB,DISTC,DISTAB,DISTBC,DISTCA
      REAL IFABR,IFABIM,IFABM,IFABP,IFBCR,IFBCIM,IFBCM,IFBCP
      REAL IFCAR,IFCAIM,IFCAM,IFCAP
      REAL  MAXCUR,ERR1,ERR2,ERR3,TIME
      INTEGER ANSWER2,STOP,K
      CHARACTER*1 LINE1,LINE2,ANSWER,ANSWER1,MAINT
      CHARACTER*132 FILENAME
      ERR1=0.05
      ERR2=1.0
      MAXCUR=10.0
C*****************************************************************************
C  'MAXCUR' STANDS FOR THE MAXIMUM CURRENT THAT IS ALLOWABLE TO FLOW IN THE LINE
C*****************************************************************************

      STOP=0

C*****************************************************************************
C   THE FOLLOWING QUERRIES ARE EQUIVALENT TO THE SETTING/RESETTING OF FLAGS
C*****************************************************************************
C   IN THE CASE OF AN ONLINE CASE.
C*****************************************************************************
      WRITE(*,*)'IS LINE SHUNT COMPENSATED'
      READ(*,900)ANSWER
      IF(ANSWER.EQ.'Y') THEN
       WRITE(*,*)' ALGORITHM DOES NOT ACT CORRECTLY IN THIS CASE'
       GO TO 301
      ENDIF
      WRITE(*,*)'IS LINE TRANSPOSED? [Y/N]'
      READ(*,900)ANSWER1
      WRITE(*,*)'WRITE THE NUMBER OF RECLOSING RELAY SHOTS'
      READ(*,*)ANSWER2
      WRITE(*,*)' IS MAINTENANCE WORK IN PROGRESS? '
      READ(*,900)MAINT
      WRITE(*,*)
      WRITE(*,*)'WRITE THE NAME OF THE FILES FROM WHICH THE FOLLOWING'
      WRITE(*,*)'                 ARE TO BE READ :'
      WRITE(*,*)'SOURCE SIDE VOLTAGES '
      READ(*,170)FILENAME
      OPEN(1,FILE=FILENAME,STATUS='OLD')
      WRITE(*,*)'LINE SIDE VOLTAGES '
      READ(*,170)FILENAME
      OPEN(2,FILE=FILENAME,STATUS='OLD')
      WRITE(*,*)'LINE CURRENTS'
      READ(*,170)FILENAME
      OPEN(8,FILE=FILENAME,STATUS='OLD')
      OPEN(9,FILE='RECLSROUT',STATUS='OLD')
      IF (ANSWER1.EQ.'Y') THEN
       WRITE(*,*)'     CASE OF TRANSPOSED LINE'
       WRITE(*,*)
       WRITE(9,*)'                 CASE OF TRANSPOSED LINE'
       WRITE(9,*)
      ELSE
       WRITE(*,*)'     CASE OF UNTRANSPOSED LINE'
       WRITE(*,*)
       WRITE(9,*)'                 CASE OF TRANSPOSED LINE'
       WRITE(9,*)
      ENDIF
```

```
C*****************************************************************************
C           Start of the ' Distance Relay ' Algorithm
C*****************************************************************************
C  This does the work of the 'Protective relay', that is the detection of the
C*****************************************************************************
C  fault and its distance from the C.B . Z0 and Z1 are the zero and positive-
C*****************************************************************************
C  sequence impedances per mile of the Transmission line and  M = [(Z0-Z1)/Z1].
C*****************************************************************************
      Z1M=1.01224E-4
      Z1P=87.856
      MM=1.8785
      MP=-24.24

C*****************************************************************************
C  TERXM STANDS FOR THE SOURCE SIDE VOLTAGE, BEGXM STANDS FOR LINE SIDE VOLTAGE
C*****************************************************************************
C  AND IXM STANDS FOR CURRENT ON THE LINE (X = A, B OR C). ALL OF THESE ARE THE
C*****************************************************************************
C  OUTPUTS OF THE PHASOR PROGRAM WHICH IN TURN ACTS ON THE DIGITALLY FILTERED
C*****************************************************************************
C  OUTPUT OF EMTP.
C*****************************************************************************
200    READ(2,100,ERR=200,END=301)TIME,BEGAM,BEGAP,
     1  BEGBM,BEGBP,BEGCM,BEGCP
210    READ(8,100,ERR=210,END=301)TIME,IAM,IAP,IBM,IBP,
     1  ICM,ICP
211    READ(1,100,ERR=211,END=301)TIME,TERAM,TERAP,
     1  TERBM,TERBP,TERCM,TERCP
      IF ((TERAM.NE.BEGAM).OR.(TERBM.NE.BEGBM).OR.
     1         (TERCM.NE.BEGCM)) THEN
       IF (MAINT.EQ.'Y') THEN
        GO TO 410
       ELSE
        WRITE(*,*)' HAS LINE BEEN REMOVED FOR MAINTENANCE?'
        READ(*,900)MAINT
        IF(MAINT.EQ.'Y')THEN
         WRITE(*,*)TIME,' C.B HAS OPENED FOR MAINTENANCE'
         GO TO 440
        ELSE
         WRITE(*,*)TIME,'      ERROR: LINE HAS OPENED, '
        WRITE(*,*)'WITHOUT THE COMMAND OF THE PROTECTIVE RELAY'
        GO TO 300
        ENDIF
       ENDIF
      ENDIF

C*****************************************************************************
C  Subroutine PTOR converts polar quantities into rectangular quantities and
C*****************************************************************************
C  RTOP, the other way around.
C*****************************************************************************
      CALL PTOR(IAM,IAP,IAR,IAIM)
      CALL PTOR(IBM,IBP,IBR,IBIM)
      CALL PTOR(ICM,ICP,ICR,ICIM)
      I0R=(IAR+IBR+ICR)/3.0
      I0IM=(IAIM+IBIM+ICIM)/3.0
      CALL RTOP(I0R,I0IM,I0M,I0P)
      MI0M=MM*I0M
      MI0P=MP+I0P
      CALL PTOR(MI0M,MI0P,MI0R,MI0IM)
      CALL PTOR(BEGAM,BEGAP,BEGAR,BEGAIM)
      CALL PTOR(BEGBM,BEGBP,BEGBR,BEGBIM)
```

```
      CALL PTOR(BEGCM,BEGCP,BEGCR,BEGCIM)
      DEABR=BEGAR-BEGBR
      DEABIM=BEGAIM-BEGBIM
      DEBCR=BEGBR-BEGCR
      DEBCIM=BEGBIM-BEGCIM
      DECAR=BEGCR-BEGAR
      DECAIM=BEGCIM-BEGAIM
      CALL RTOP(DEABR,DEABIM,DEABM,DEABP)
      CALL RTOP(DEBCR,DEBCIM,DEBCM,DEBCP)
      CALL RTOP(DECAR,DECAIM,DECAM,DECAP)
      DIABR=IAR-IBR
      DIABIM=IAIM-IBIM
      DIBCR=IBR-ICR
      DIBCIM=IBIM-ICIM
      DICAR=ICR-IAR
      DICAIM=ICIM-IAIM
      IFAR=IAR+MI0R
      IFAIM=IAIM+MI0IM
      IFBR=IBR+MI0R
      IFBIM=IBIM+MI0IM
      IFCR=ICR+MI0R
      IFCIM=ICIM+MI0IM
      IFABR=DIABR+MI0R
      IFABIM=DIABIM+MI0IM
      IFBCR=DIBCR+MI0R
      IFBCIM=DIBCIM+MI0IM
      IFCAR=DICAR+MI0R
      IFCAIM=DICAIM+MI0IM
      CALL RTOP(IFAR,IFAIM,IFAM,IFAP)
      CALL RTOP(IFBR,IFBIM,IFBM,IFBP)
      CALL RTOP(IFCR,IFCIM,IFCM,IFCP)
      CALL RTOP(IFABR,IFABIM,IFABM,IFABP)
      CALL RTOP(IFBCR,IFBCIM,IFBCM,IFBCP)
      CALL RTOP(IFCAR,IFCAIM,IFCAM,IFCAP)
      ZAM=BEGAM/IFAM
      ZBM=BEGBM/IFBM
      ZCM=BEGCM/IFCM
      ZABM=DEABM/IFABM
      ZBCM=DEBCM/IFBCM
      ZCAM=DECAM/IFCAM
      DISTA=ZAM/Z1M
      DISTB=ZBM/Z1M
      DISTC=ZCM/Z1M
      DISTAB=ZABM/Z1M
      DISTBC=ZBCM/Z1M
      DISTCA=ZCAM/Z1M
      IF(ANSWER1.EQ.'Y')THEN
       CALL PWMOD(DISTA,DISTB,DISTC,DISTAB,DISTBC,DISTCA,IAM,IBM,ICM)
C****************************************************************************
C  Subroutine PWMOD is used for the case when a Piece-wise model of a transposed
C****************************************************************************
C  line is desired to be used.
C****************************************************************************

      ELSE
       CALL CONTMOD(DISTA,DISTB,DISTC,DISTAB,DISTBC,DISTCA,IAM,IBM,ICM)
C****************************************************************************
C  Subroutine CONTMOD is used both in the case wherea continuouos model of a
C****************************************************************************
C  transposed line is desired and where the line is untransposed.
C****************************************************************************

      ENDIF
      IF(KFAULT.EQ.70) GO TO 200
```

```
C*************************************************************
C  KFAULT IS A VARIABLE OBTAINED FROM 'CONTMOD' AND 'PWMOD' WHICH ASSIGN EACH
C****************************************************************************
C  SPECIFIC TYPE OF FAULT A NUMBER
C****************************************************************************

C****************************************************************************
C  THE FOLLOWING LOGIC IS FOR FINDING WHETHER THE C.B HAS OPENED OR NOT.
C****************************************************************************
410    READ(2,100,ERR=410,END=301)TIME,BEGAM,BEGAP,
      1 BEGBM,BEGBP,BEGCM,BEGCP
420     READ(8,100,ERR=420,END=301)TIME,IAM,IAP,
      1  IBM,IBP,ICM,ICP
421    READ(1,100,ERR=421,END=301)TIME,TERAM,TERAP,
      1 TERBM,TERBP,TERCM,TERCP
       IF ((TERAM.NE.BEGAM).AND.(TERBM.NE.BEGBM).AND.
      1             (TERCM.NE.BEGCM)) GO  TO 430
       GO TO 410

C****************************************************************************
C  AT THIS TIME, WITH THE DETECTION OF THE 'TRIPPING'  OF THE C.B THE COMMAND IS
C****************************************************************************
C  GIVEN TO ENERGISE THE 'CLOSING COIL' OF THE C.B. USING THE LOGIC ILLUSTRATED
C****************************************************************************
C  IN THE REPORT.
C****************************************************************************
430    IF(MAINT.EQ.'Y') WRITE(*,165)TIME
440    WRITE(9,*)'                      RECLOSER DECISION FOR C.B. NEAR GEN1'
       WRITE(*,*)
       WRITE(9,*)'                        FLAG=1 :CONTINUE WITH RECLOSING '
       WRITE(9,*)'                        FLAG=0 :ABORT RECLOSING '
       WRITE(*,*)
       WRITE(9,125)
       WRITE(*,*)

C****************************************************************************
C     'ANSWER2' GIVES THE NO. OF RELAY SHOTS DESIRED. EACH RECLOSING SHOT IS
C****************************************************************************
C     TIME DELAYED WITH RESPECT TO THE PREVIOUS ONE BY ONE CYCLE, JUST AS
C****************************************************************************
C     THE FIRST SHOT IS.
C****************************************************************************
230    CYCLE = 0.0
       DO 700 K=1,ANSWER2

C****************************************************************************
C  THE FOLLOWING LOGIC IS FOR FINDING WHETHER THE C.B HAS RECLOSED.
C****************************************************************************
240    READ(2,100,ERR=240,END=300)TIME,BEGAM,BEGAP,
      1 BEGBM,BEGBP,BEGCM,BEGCP
250    READ(8,100,ERR=250,END=300)TIME,IAM,IAP,
      1  IBM,IBP,ICM,ICP
260    READ(1,100,ERR=260,END=300)TIME,TERAM,TERAP,
      1 TERBM,TERBP,TERCM,TERCP
       IF ((TERAM.EQ.BEGAM).OR.(TERBM.EQ.BEGBM).OR.(TERCM.EQ.BEGCM))
      1                                                  GO TO 220
       GO TO 240
220    IF(MAINT.EQ.'Y')THEN
        WRITE(*,*)'IS MAINTENANCE STILL GOING ON ?'
        READ(*,900)MAINT
        IF(MAINT.EQ.'Y') THEN
         WRITE(*,*)TIME,'ERROR: LINE(S) HAS(HAVE) RECLOSED WITHOUT'
         WRITE(*,*)'      COMPLETION OF MAINTENANCE'
         GO TO 300
```

```
      ELSE
       WRITE(*,*)TIME,' C.B. RECLOSING AFTER COMPLETION'
       WRITE(*,*) '     OF MAINTENANCE'
       GO TO 550
      ENDIF
     ENDIF

C*****************************************************************************
C    THE PHASOR PROGRAM UTILISES A 12 SAMPLE WINDOW TO CALCULATE THE PHASORS.
C*****************************************************************************
C    HENCE, TO ENSURE ACCURACY AFTER THE LINE HAS RECLOSED, WE WAIT FOR TWELVE
C*****************************************************************************
C    SAMPLES (ONE CYCLE) TO START OUR 'POST-RECLOSE' ANALYSIS
C*****************************************************************************
      DO 400 J=1,12
      READ(2,100,ERR=400,END=300)TIME,BEGAM,BEGAP,BEGBM,
     1  BEGBP,BEGCM,BEGCP
      READ(8,100,ERR=400,END=300)TIME,IAM,IAP,IBM,
     1 IBP,ICM,ICP
      READ(1,100,ERR=400,END=300)TIME,TERAM,TERAP,
     1 TERBM,TERBP,TERCM,TERCP
      CYCLE=CYCLE + 0.083333
400   CONTINUE

C*****************************************************************************
C    AFTER HAVING WAITED FOR SUFFICIENT TIME TO ASSURE ACCURACY OF THE PHASOR-
C*****************************************************************************
C    MEASUREMENT, THAT IS, 12 SAMPLES OR 1 CYCLE IN THIS RESEARCH, THE ACTUAL
C*****************************************************************************
C    RECLOSING ALGORITHM STARTS.
C*****************************************************************************
C 3456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
      IF((KFAULT.EQ.11).OR.(KFAULT.EQ.21).OR.(KFAULT.EQ.40))THEN
       IF(IAM.GE.MAXCUR) GO TO 500
       LINE1='B'
       LINE2='C'
       CALL RECLOSER(BEGBM,BEGBP,BEGCM,BEGCP,IBM,ICM,STOP,K,CYCLE)
      ELSE
       IF((KFAULT.EQ.1).OR.(KFAULT.EQ.22).OR.(KFAULT.EQ.50))THEN
        IF(IBM.GE.MAXCUR) GO TO 500
        LINE1='A'
        LINE2='C'
        CALL RECLOSER(BEGAM,BEGAP,BEGCM,BEGCP,IAM,ICM,STOP,K,CYCLE)
       ELSE
        IF((KFAULT.EQ.2).OR.(KFAULT.EQ.12).OR.(KFAULT.EQ.30))THEN
         IF(ICM.GE.MAXCUR) GO TO 500
         LINE1='B'
         LINE2='A'
         CALL RECLOSER(BEGBM,BEGBP,BEGAM,BEGAP,IBM,IAM,STOP,K,CYCLE)
        ELSE
         IF(KFAULT.EQ.0)THEN
          IF((IBM.GE.MAXCUR).OR.(ICM.GE.MAXCUR)) GO TO 500
          IF(IBM.NE.0.0) THEN
           LINE1='A'
           LINE2='C'
           CALL RECLOSER(BEGAM,BEGAP,BEGCM,BEGCP,IAM,ICM,STOP,K,CYCLE)
          ELSE
           LINE1='B'
           LINE2='A'
           CALL RECLOSER(BEGBM,BEGBP,BEGAM,BEGAP,IBM,IAM,STOP,K,CYCLE)
          ENDIF
         ELSE
          IF(KFAULT.EQ.10)THEN
           IF((IAM.GE.MAXCUR).OR.(ICM.GE.MAXCUR)) GO TO 500
```

```
          IF(IAM.NE.0.0) THEN
           LINE1='B'
           LINE2='C'
           CALL RECLOSER(BEGBM,BEGBP,BEGCM,BEGCP,IBM,ICM,STOP,K,CYCLE)
          ELSE
           LINE1='B'
           LINE2='A'
           CALL RECLOSER(BEGBM,BEGBP,BEGAM,BEGAP,IBM,IAM,STOP,K,CYCLE)
          ENDIF
         ELSE
          IF(KFAULT.EQ.20)THEN
           IF((IBM.GE.MAXCUR).OR.(IAM.GE.MAXCUR)) GO TO 500
           IF(IBM.NE.0.0) THEN
            LINE1='A'
            LINE2='C'
            CALL RECLOSER(BEGAM,BEGAP,BEGCM,BEGCP,IAM,ICM,STOP,K,CYCLE)
           ELSE
            LINE1='B'
            LINE2='C'
            CALL RECLOSER(BEGBM,BEGBP,BEGCM,BEGCP,IBM,ICM,STOP,K,CYCLE)
           ENDIF
          ENDIF
         ENDIF
        ENDIF
       ENDIF
      ENDIF
      ENDIF
      IF (STOP.EQ.1)GO TO 550
C***************************************************************************
C     'STOP' DIRECTS THE RECLOSING RELAY TO RETURN TO NORMAL ACTION IN CASE THE
C***************************************************************************
C     SYSTEM RETURNS TO NORMAL.
C***************************************************************************

100   FORMAT(F8.5,3(1X,F10.2,1X,F8.2))
125   FORMAT(2X,10HTIME(SECS),1X,9HTRIAL NO.,13X,8HDECISION,20X,4HFLAG)
165   FORMAT(F8.5,3X,27HCIRCUIT BREAKER OPENING FOR,
     1                24H MAINTENANCE ON THE LINE)
170   FORMAT(A132)
185   FORMAT(F8.5,3X,32HNO FAULT ON THE LINE: DON'T TRIP)
190   FORMAT(F8.5,
     1 53H FAULT ON RECLOSED LINE : RECLOSING SHOULD BE ABORTED)
800   FORMAT(36HTOTAL NUMBER OF CYCLES FOR RECLOSING,
     1 20H ALGORITHM TO WORK =,F6.2)
900   FORMAT(A1)
700   CONTINUE
300   WRITE(*,*)
      WRITE(*,800)CYCLE
      WRITE(*,*)
      WRITE(*,*)' OUTPUT IS GIVEN IN FILE NAMED RECLSROUT'
      WRITE(*,*)
      GO TO 301

C***************************************************************************
C     THE FOLLOWING ALGORITHM WAITS FOR THE C.B TO RECLOSE COMPLETELY ON ALL
C***************************************************************************
C     POLES IN THE CASE WHEN THE SYSTEM HAS RETURNED TO NORMAL AFTER WHICH IT
C***************************************************************************
C     IT RETURNS CONTROL TO THE 'PROTECTIVE RELAY'.
C***************************************************************************
550   READ(2,100,ERR=550,END=301)TIME,BEGAM,BEGAP,
     1 BEGBM,BEGBP,BEGCM,BEGCP
560   READ(8,100,ERR=560,END=301)TIME,IAM,IAP,
     1  IBM,IBP,ICM,ICP
```

```
570   READ(1,100,ERR=570,END=301)TIME,TERAM,TERAP,
     1 TERBM,TERBP,TERCM,TERCP
      IF ((TERAM.EQ.BEGAM).AND.(TERBM.EQ.BEGBM).AND.(TERCM.EQ.BEGCM))
     1                                                GO TO 200
      GO TO 550
500   WRITE(9,190)TIME
301   STOP
      END
      SUBROUTINE PWMOD(DISTA,DISTB,DISTC,DISTAB,DISTBC,DISTCA,IAM,
     1                                                     IBM,ICM)
      COMMON TIME,KFAULT,ERR1,ERR2,LINE1,LINE2,ANSWER1
      REAL IAM,IBM,ICM

C***************************************************************************
C     SUBROUTINE 'PWMOD' DETECTS A FAULT IN THE CASE OF A TRANSPOSED LINE. THE
C***************************************************************************
C     LINE IS ASSUMED TO BE 113.4 MILES IN LENGTH.
C***************************************************************************
      IF((DISTAB.LE.113.4).AND.(ABS(IAM-IBM).LE.ERR2))THEN

       WRITE(*,*)TIME,' L-L FAULT ON AB, RECLOSE ON C'
       KFAULT=30
      ELSE
       IF((DISTBC.LE.113.4).AND.(ABS(IBM-ICM).LE.ERR2)) THEN
        WRITE(*,*)TIME,' L-L FAULT  ON BC, RECLOSE ON A'
        KFAULT=40
       ELSE
        IF((DISTCA.LE.113.4).AND.(ABS(ICM-IAM).LE.ERR2)) THEN
         WRITE(*,*)TIME,' L-L FAULT ON CA, RECLOSE ON B'
         KFAULT=50
        ELSE
         IF(DISTA.LE.37.8) THEN
          WRITE(*,*)TIME,' L-G FAULT IN ZONE1 ON A:'
          WRITE(*,*)'                         RECLOSE ON PHASE C'
          KFAULT=2
         ELSE
          IF(DISTA.LE.75.6) THEN
           WRITE(*,*)TIME,' L-G FAULT IN ZONE2 ON A: '
           WRITE(*,*)'                        RECLOSE ON B OR C'
           KFAULT=0
          ELSE
           IF(DISTA.LE.113.4) THEN
            WRITE(*,*)TIME,' L-G FAULT IN ZONE3 ON A: '
            WRITE(*,*)'                       RECLOSE ON PHASE B'
            KFAULT=1
           ELSE
            IF(DISTB.LE.37.8) THEN
             WRITE(*,*)TIME,' L-G FAULT IN ZONE1 ON B: '
             WRITE(*,*)'                      RECLOSE ON A OR C'
             KFAULT=10
            ELSE
             IF(DISTB.LE.75.6) THEN
              WRITE(*,*)TIME,' L-G FAULT IN ZONE2 ON B:'
              WRITE(*,*)'                 RECLOSE ON PHASE  C'
              KFAULT=12
             ELSE
              IF(DISTB.LE.113.4) THEN
               WRITE(*,*)TIME,' L-G FAULT IN ZONE3 ON B:'
               WRITE(*,*)'                RECLOSE ON PHASE A'
               KFAULT=11
              ELSE
               IF(DISTC.LE.37.8) THEN
                WRITE(*,*)TIME,' C-G FAULT IN ZONE1:'
                WRITE(*,*)'                RECLOSE ON PHASE A'
```

```
            KFAULT=21
           ELSE
            IF(DISTC.LE.75.6) THEN
             WRITE(*,*)TIME,' C-G FAULT IN ZONE2:'
             WRITE(*,*)'                RECLOSE ON PHASE B'
             KFAULT=22
            ELSE
             IF(DISTC.LE.113.4) THEN
              WRITE(*,*)TIME,' C-G FAULT IN ZONE3:'
              WRITE(*,*)'                RECLOSE ON A OR B'
              KFAULT=20
             ELSE
              WRITE(*,*)TIME,' NO FAULT DETECTED ON THE SYSTEM'
              KFAULT=70
             ENDIF
            ENDIF
           ENDIF
          ENDIF
         ENDIF
        ENDIF
       ENDIF
      ENDIF
     ENDIF
    ENDIF
   ENDIF
  ENDIF
  RETURN
  END
  SUBROUTINE CONTMOD(DISTA,DISTB,DISTC,DISTAB,DISTBC,DISTCA,IAM,
 1                                                  IBM,ICM)
  COMMON TIME,KFAULT,ERR1,ERR2,LINE1,LINE2,ANSWER1
  REAL IAM,IBM,ICM

C*****************************************************************************
C     SUBROUTINE 'CONTMOD' DETECTS A FAULT IN THE CASE OF AN UNTRANSPOSED LINE.
C*****************************************************************************
C     THE LINE IS ASSUMED TO BE 56.8 MILES IN LENGTH.
C*****************************************************************************
      IF((DISTAB.LE.56.8).AND.(ABS(IAM-IBM).LE.ERR2))THEN
       WRITE(*,*)TIME,' L-L FAULT ON AB, RECLOSE ON C'
       KFAULT=30
      ELSE
       IF((DISTBC.LE.56.8).AND.(ABS(IBM-ICM).LE.ERR2)) THEN
        WRITE(*,*)TIME,' L-L FAULT  ON BC, RECLOSE ON A'
        KFAULT=40
       ELSE
        IF((DISTCA.LE.56.8).AND.(ABS(ICM-IAM).LE.ERR2)) THEN
         WRITE(*,*)TIME,' L-L FAULT ON CA, RECLOSE ON B'
         KFAULT=50
        ELSE
         IF(DISTA.LE.56.8) THEN
          WRITE(*,*)TIME,' L-G FAULT ON A: RECLOSE ON C'
          KFAULT=2
         ELSE
          IF(DISTB.LE.56.8) THEN
           WRITE(*,*)TIME,' L-G FAULT ON B: RECLOSE ON  A OR C'
           KFAULT=10
          ELSE
           IF(DISTC.LE.56.8) THEN
            WRITE(*,*)TIME,' L-G FAULT ON C: RECLOSE ON A'
            KFAULT=21
           ELSE
            WRITE(*,*)TIME,' NO FAULT DETECTED ON SYSTEM'
            KFAULT=70
```

```
          ENDIF
         ENDIF
        ENDIF
       ENDIF
      ENDIF
     ENDIF
     RETURN
     END
C*****************************************************************************
C    SUBROUTINE 'RECLOSER' IS USED FOR THE POST-RECLOSING PERIOD.
C*****************************************************************************
      SUBROUTINE RECLOSER(V1M,V1P,V2M,V2P,I1M,I2M,STOP,K,CYCLE)
      COMMON TIME,KFAULT,ERR1,ERR2,LINE1,LINE2,ANSWER1
      REAL I1M,I2M
      CHARACTER*1 LINE1,LINE2,ANSWER1
      INTEGER STOP,K
      IF((ABS(V1M-1.0).LE.ERR1).AND.(ABS(V2M-1.0).LE.ERR1).AND.
     1                                  (ABS(V1P-V2P).GT.ERR2))THEN
       WRITE(9,185)TIME,K
       STOP=1
      ELSE
       IF(V1M.LE.ERR1)THEN
        WRITE(9,160)TIME,K,LINE1
       ELSE
        IF(V2M.LE.ERR1)THEN
         WRITE(9,160)TIME,K,LINE2
        ELSE
         IF(ANSWER1.NE.'Y')THEN
          IF((ABS(V1M-V2M).LE.ERR1))THEN
           WRITE(9,120)TIME,K,LINE1,LINE2
           IF(LINE1.EQ.'B')THEN
            IF(LINE2.EQ.'C')THEN
             KFAULT=40
            ELSE
             KFAULT=30
            ENDIF
           ELSE
            KFAULT=50
           ENDIF
          ELSE
           WRITE(9,170)TIME,K
           STOP=1
          ENDIF
         ELSE
          IF(ABS(V1M-V2M).LE.ERR1) THEN
           RTIME=TIME
           GO TO 500
          ELSE
           WRITE(9,121)TIME,K
          ENDIF
         ENDIF
        ENDIF
       ENDIF
      ENDIF
      RETURN
500   READ(1,600,ERR=500,END=350)RTIME,V1AM,V1AP,
     1                         V1BM,V1BP,V1CM,V1CP
510   READ(2,600,ERR=510,END=350)RTIME,V2AM,V2AP,
     1                         V2BM,V2BP,V2CM,V2CP
520   READ(8,600,ERR=520,END=350)TIME,CAM,CAP,
     1                         CBM,CBP,CCM,CCP
      CYCLE=CYCLE + 0.083333
      IF((LINE1.EQ.'A').AND.(LINE2.EQ.'C'))THEN
```

```
      IF((V1AM.EQ.V2AM).OR.(V1CM.EQ.V2CM))THEN
       IF((ABS(V2AM-V2CM).LE.ERR1).AND.
     1                  (ABS(V2AP-V2CP).LE.ERR2))THEN
        WRITE(9,120)RTIME,K,LINE1,LINE2
        KFAULT=50
        GO TO 360
       ELSE
        WRITE(9,170)RTIME,K
        STOP=1
        GO TO 360
       ENDIF
      ELSE
       WRITE(9,175)RTIME,K
       GO TO 500
      ENDIF
     ELSE
      IF((LINE1.EQ.'B').AND.(LINE2.EQ.'C'))THEN
       IF((V1BM.EQ.V2BM).OR.(V1CM.EQ.V2CM))THEN
        IF((ABS(V2BM-V2CM).LE.ERR1))THEN
         WRITE(9,120)RTIME,K,LINE1,LINE2
         KFAULT=40
         GO TO 360
        ELSE
         WRITE(9,170)RTIME,K
         STOP=1
         GO TO 360
        ENDIF
       ELSE
        WRITE(9,175)RTIME,K
        GO TO 500
       ENDIF
      ELSE
       IF((LINE1.EQ.'B').AND.(LINE2.EQ.'A'))THEN
        IF((V1BM.EQ.V2BM).OR.(V1AM.EQ.V2AM))THEN
         IF((ABS(V2BM-V2AM).LE.ERR1).AND.
     1                  (ABS(V2BP-V2AP).LE.ERR2))THEN
          WRITE(9,120)RTIME,K,LINE1,LINE2
          KFAULT=30
          GO TO 360
         ELSE
          WRITE(9,170)RTIME,K
          GO TO 360
         ENDIF
        ELSE
         WRITE(9,175)RTIME,K
         GO TO 500
        ENDIF
       ENDIF
      ENDIF
     ENDIF
350   WRITE(9,195)RTIME,K
600   FORMAT(F8.5,3(1X,F10.2,1X,F8.2))
120   FORMAT(2X,1F8.5,11X,I1,6X,13HL-L FAULT ON ,A1,5H AND ,A1,
     1  18H                 0)
121   FORMAT(2X,F8.5,10X,I1,2X,
     1 46H UNABLE TO DETECT ANY FAULT ON TRANSPOSED LINE)
160   FORMAT(2X,1F8.5,10X,I1,2X,13HL-G FAULT ON ,A1,
     1 24H                      0)
170   FORMAT(2X,1F8.5,10X,I1,6X,
     1            38HNO FAULT DETECTED ON THE LINE         1)
175   FORMAT(2X,1F8.5,10X,I1,2X
     1  51H WAITING FOR A 2ND POLE TO RECLOSE TO MAKE DECISION)
185   FORMAT(2X,F8.5,10X,I1,2X,
     1  48H OTHER SIDE HAS RECLOSED, RECLOSING CAN CONTINUE)
```

```
195   FORMAT(2X,F8.5,10X,I1,2X,26H END OF FILE REACHED WHILE,
     1  34H WAITING FOR A 2ND POLE TO RECLOSE)
360   RETURN
      END
      SUBROUTINE RTOP(ER,EIM,EM,EP)
      REAL ER,EIM,EM,EP
      EPSILON=1.0E-4
      AER=ABS(ER)
      AEIM=ABS(EIM)
      IF((AER.LE.EPSILON).AND.(AEIM.LE.EPSILON)) GO TO 50
      EM=SQRT((ER*ER)+(EIM*EIM))
      IF (AER.LE.EPSILON) GO TO 60
      IF (ER.GT.0.0) THEN
       EP= ATAN(EIM/ER)*(180.0/3.1416)
      ELSE
       IF (EIM.GE.0.0) THEN
        EP= ATAN(EIM/ER)*(180.0/3.1416) + 90
       ELSE
        EP= ATAN(EIM/ER)*(180.0/3.1416) + 180
       ENDIF
      ENDIF
      RETURN
50    EP=0.0
      EM=0.0
      RETURN
60    IF (EIM.GE.0.0)THEN
       EP=90.0
      ELSE
       EP=270.0
      ENDIF
      RETURN
      END
      SUBROUTINE PTOR(EM,EP,ER,EIM)
      REAL ER,EIM,EM,EP
      EP=EP*3.14159/180.0
      ER=EM*COS(EP)
      EIM=EM*SIN(EP)
      RETURN
      END
```

```
BEGIN NEW DATA CASE
C_____________________________________________________________________________
C        CASE OF LINE TO GROUND FAULT ON B(NO REACTOR)
C                          EMTP LISTING
C_____________________________________________________________________________
C FIRST MISC. DATA CARD
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
C TSTEP   TMAX      XOPT   COPT
  6.6E-5   0.2E0   0.6E2   0.6E2
C INTEGER MISC. DATA CARD
C   IOUT      IPLOT   IDOUBL  KSSOUT   MAXOUT        ICAT
       7        1       1       0        1              0
C BRANCHES
C REACTANCE BEHIND GEN. TERMINALS
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
$VINTAGE,1
 1GENA  TERA                      0.5906680E1     0.5906680E2
 2GENB  TERB                      0.4028303E1     0.2715743E1
                                  0.5906680E1     0.5906680E2
 3GENC  TERC                      0.4028303E1     0.2715743E1
                                  0.4028303E1     0.2715743E1
                                  0.5906680E1     0.5906680E2
$VINTAGE,0
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
C TRANSMISSION LINES(UNTRANSPOSED LINE,TRAVELLING WAVE MODEL,56.8 MILES)
C                  (LINE FROM BEGA TO MIDA)
$VINTAGE,1
-1BEGA  MIDA                   0.51631E00  0.15891E01  0.53023E01   -0.284E02 0 03

-2BEGB  MIDB                  0.22005E-01  0.59198E00  0.73602E01   -0.284E02 0 03

-3BEGC  MIDC                  0.18181E-01  0.48133E00  0.86535E01   -0.284E02 0 03

C 345678901234567890123456789012345678901234567890123456789012345678901234567890
$VINTAGE,0
  0.59439000 -0.70711000 -0.41336000
 -0.23500E-2 -0.11219E-7 -0.20707E-2
  0.54164000  0.11564E-5  0.81133000
  0.46125E-2  0.23008E-6-0.118868E-2
  0.59439000  0.70711000 -0.41336000
 -0.23350E-2 -0.11220E-7 -0.20705E-2
C                  (LINE FROM MIDA TO ENDA)
$VINTAGE,1
-1MIDA  ENDA                   0.51631E00  0.15891E01  0.53023E01   -0.284E02 0 03

-2MIDB  ENDB                  0.22005E-01  0.59198E00  0.73602E01   -0.284E02 0 03

-3MIDC  ENDC                  0.18181E-01  0.48133E00  0.86535E01   -0.284E02 0 03
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
$VINTAGE,0
  0.59439000 -0.70711000 -0.41336000
 -0.23500E-2 -0.11219E-7 -0.20707E-2
  0.54164000  0.11564E-5  0.81133000
  0.46125E-2  0.23008E-6-0.118868E-2
  0.59439000  0.70711000 -0.41336000
 -0.23350E-2 -0.11220E-7 -0.20705E-2
BLANK CARD ENDING BRANCHES
C SWITCHES (C.B'S ONLY AT SENDING END)
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
  TERA  BEGAM -2.5       0.05
  TERB  BEGBM -2.5       0.05
  TERC  BEGCM -2.5       0.05
  MIDB        -2.5       2.0
```

```
  TERA  BEGAM 0.094710    2.0
C MEASURING SWITCHES IN PHASES B AND C
  BEGAM BEGA  -2.0        2.0
  BEGBM BEGB  -2.0        2.0                                                 1
  BEGCM BEGC  -2.0        2.0                                                 1
BLANK CARD ENDING SWITCHES                                                    1
C SOURCES (L-L:765KV)
C 345678901234567890123456789012345678901234567890123456789012345678901234567890
14GENA    6.246190E5 60.0      0.0
14GENB    6.246190E5 60.0     -120.0                        -1.0
14GENC    6.246190E5 60.0     -240.0                        -1.0
BLANK CARD ENDING SOURCES                                   -1.0
  TERA  TERB  TERC  BEGA  BEGB  BEGC
BLANK CARD ENDING NODE VOLTAGE OUTPUT REQUESTS
BLANK CARD ENDING PLOT OUTPUT REQUESTS
BLANK CARD ENDING CASE
```

APPENDIX-II

PLOTS OF SIMULATIONS

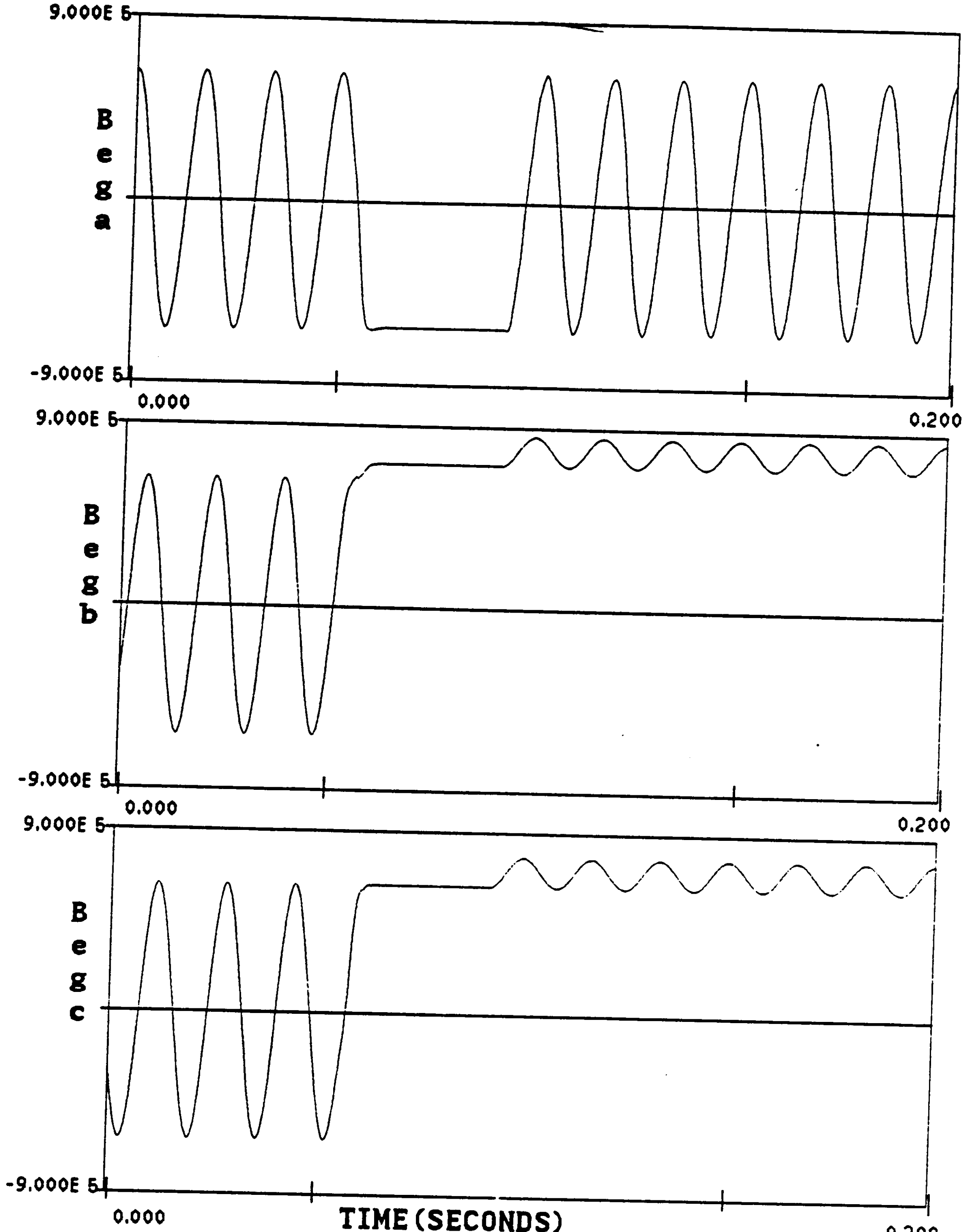


Figure 1: Line side voltages (normal case)
transposed lines, no reactor

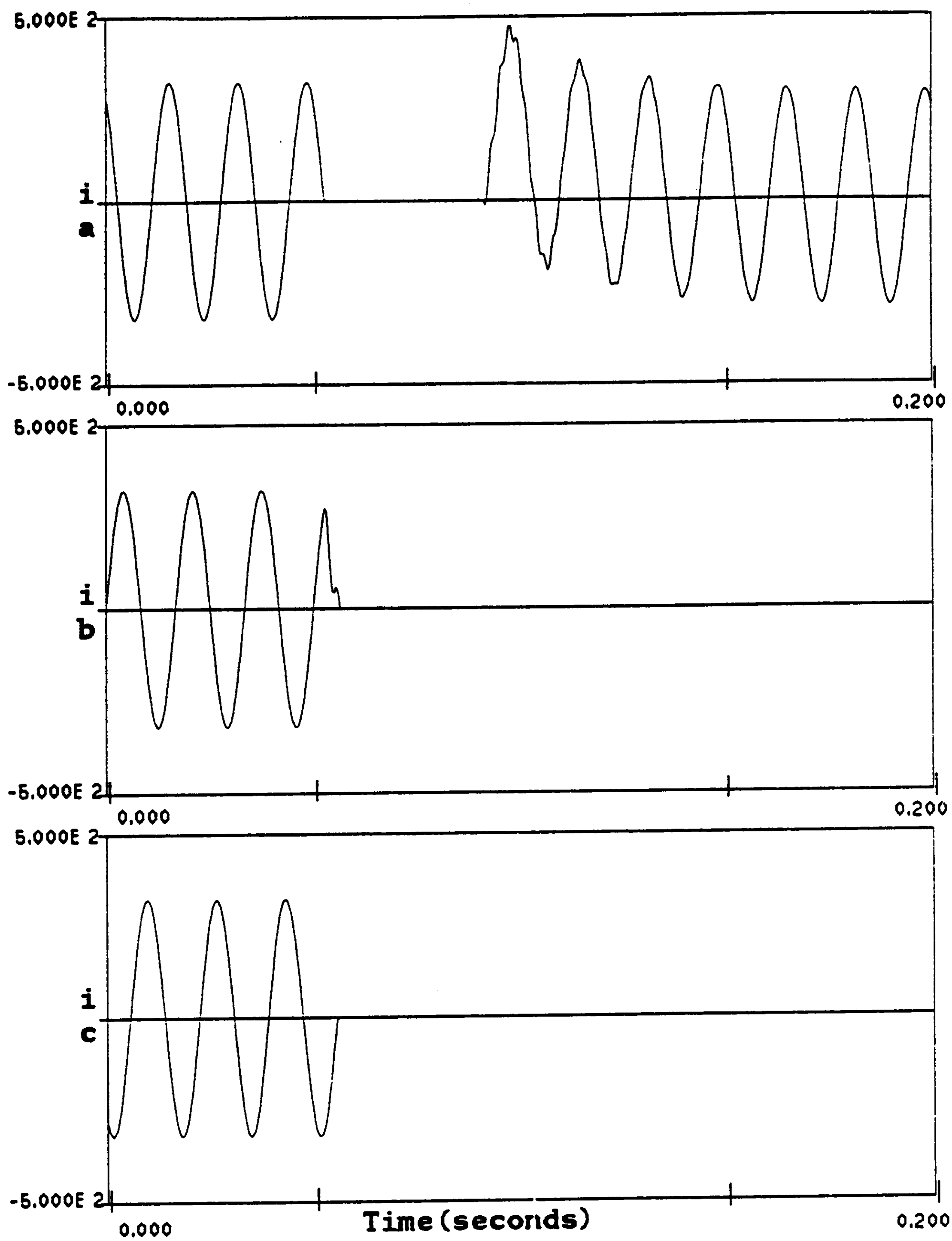


Figure 2: Line currents (normal case)
transposed lines, no reactor

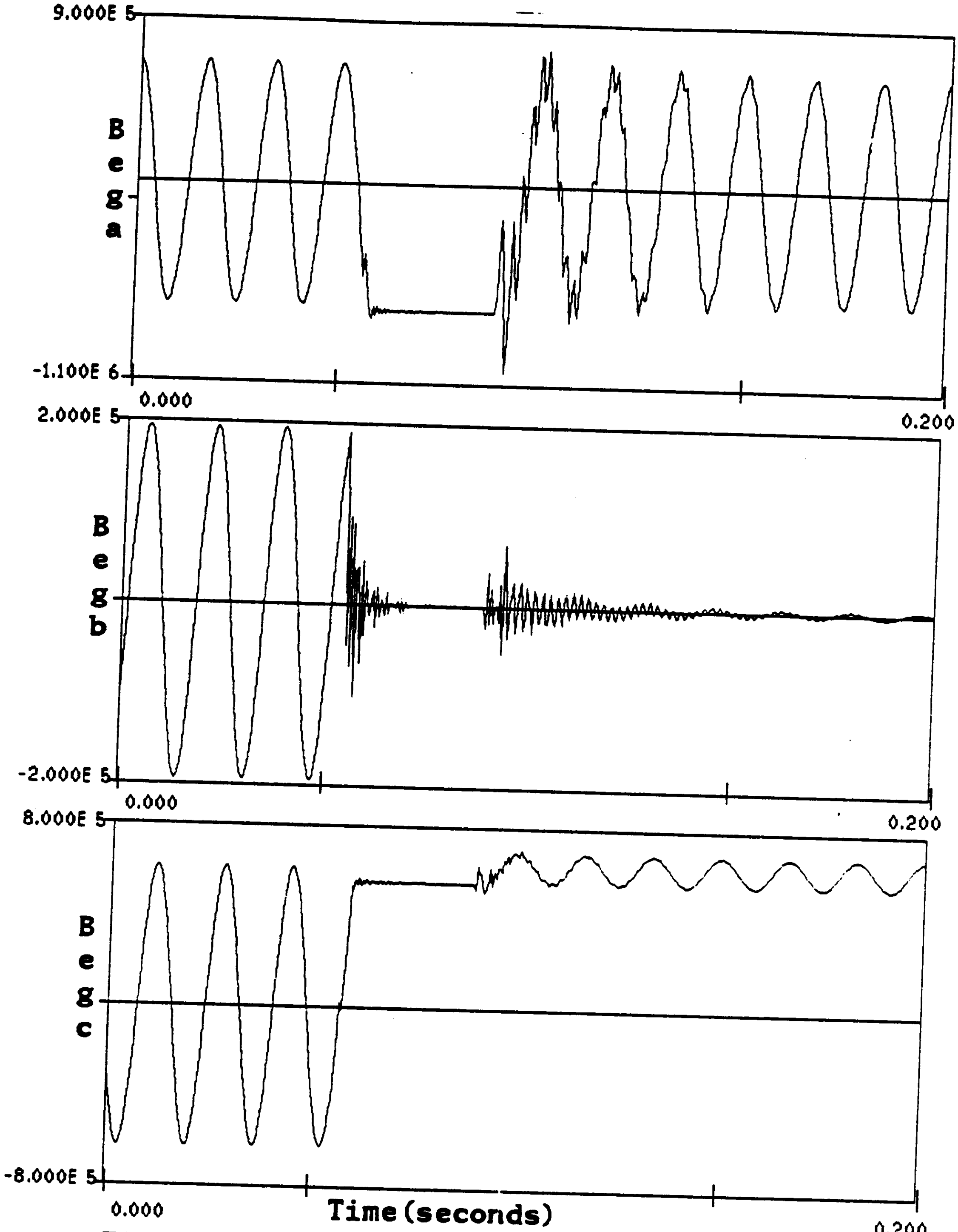


Figure 3: Line side voltages (1-g case)
transposed lines, no reactor

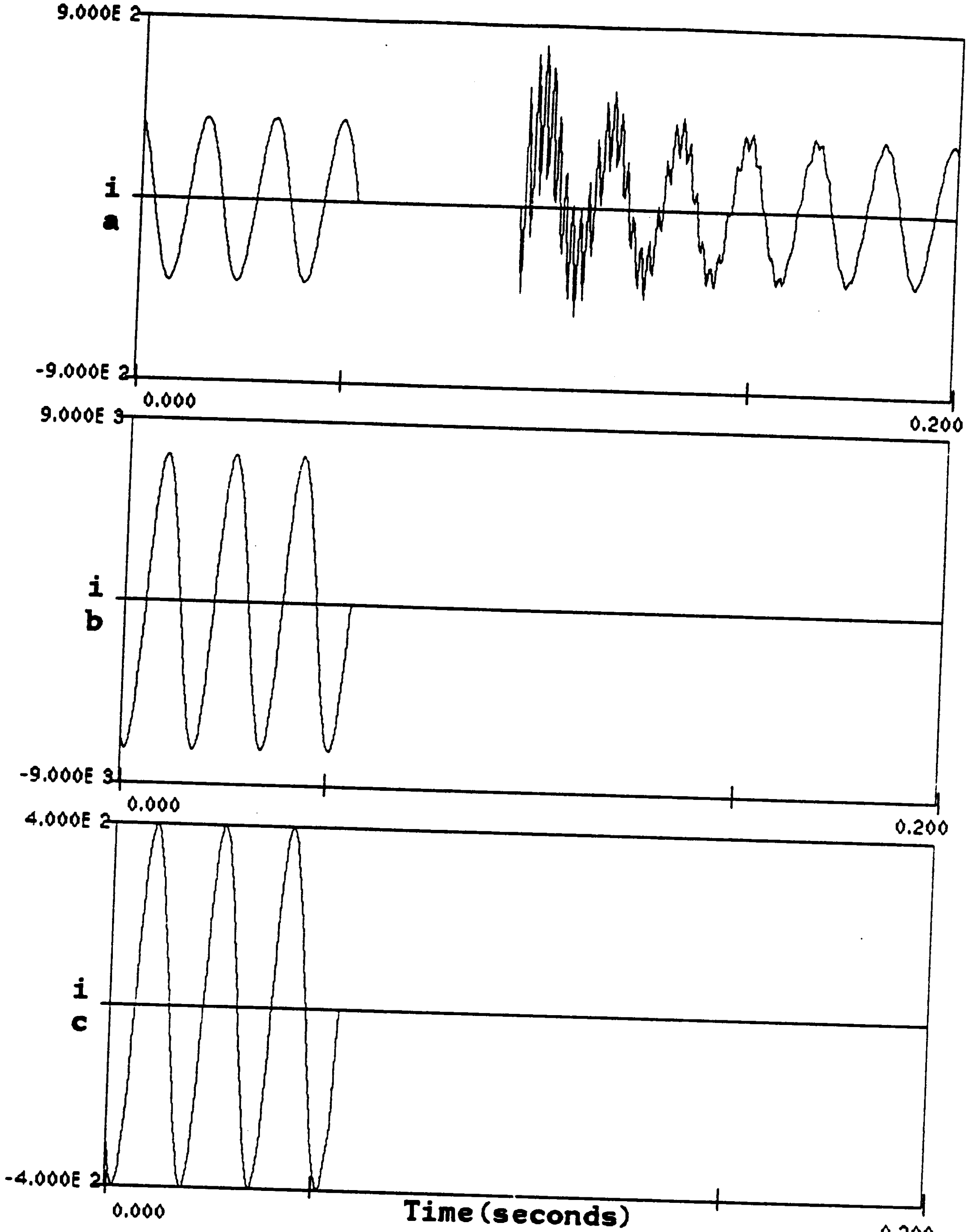


Figure 4: Line currents (1-g case) transposed lines, no reactor

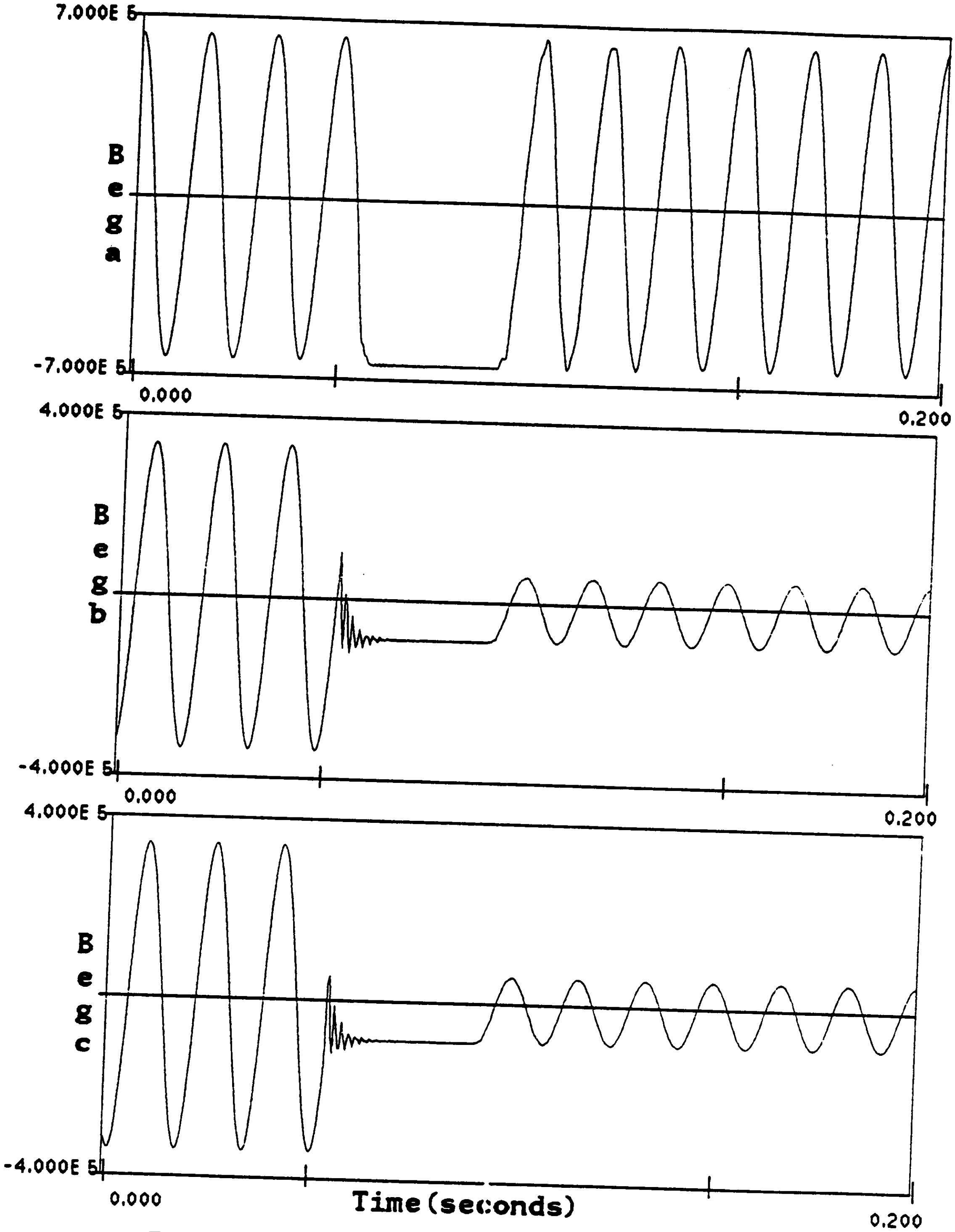


Figure 5: Line side voltages (1-1 case) transposed line, no reactor

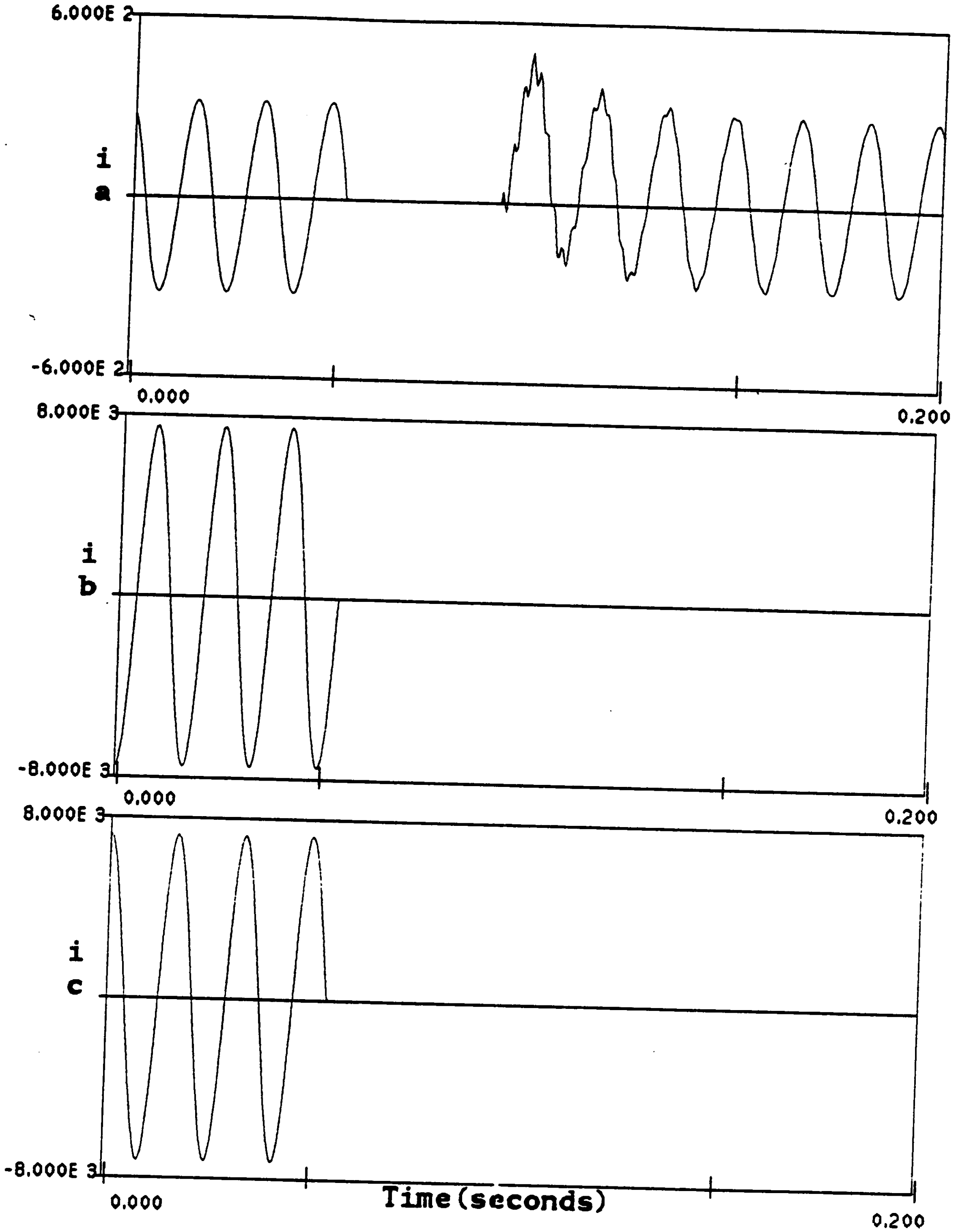


Figure 6: Line currents (1-1 case)
.. transposed, no reactor

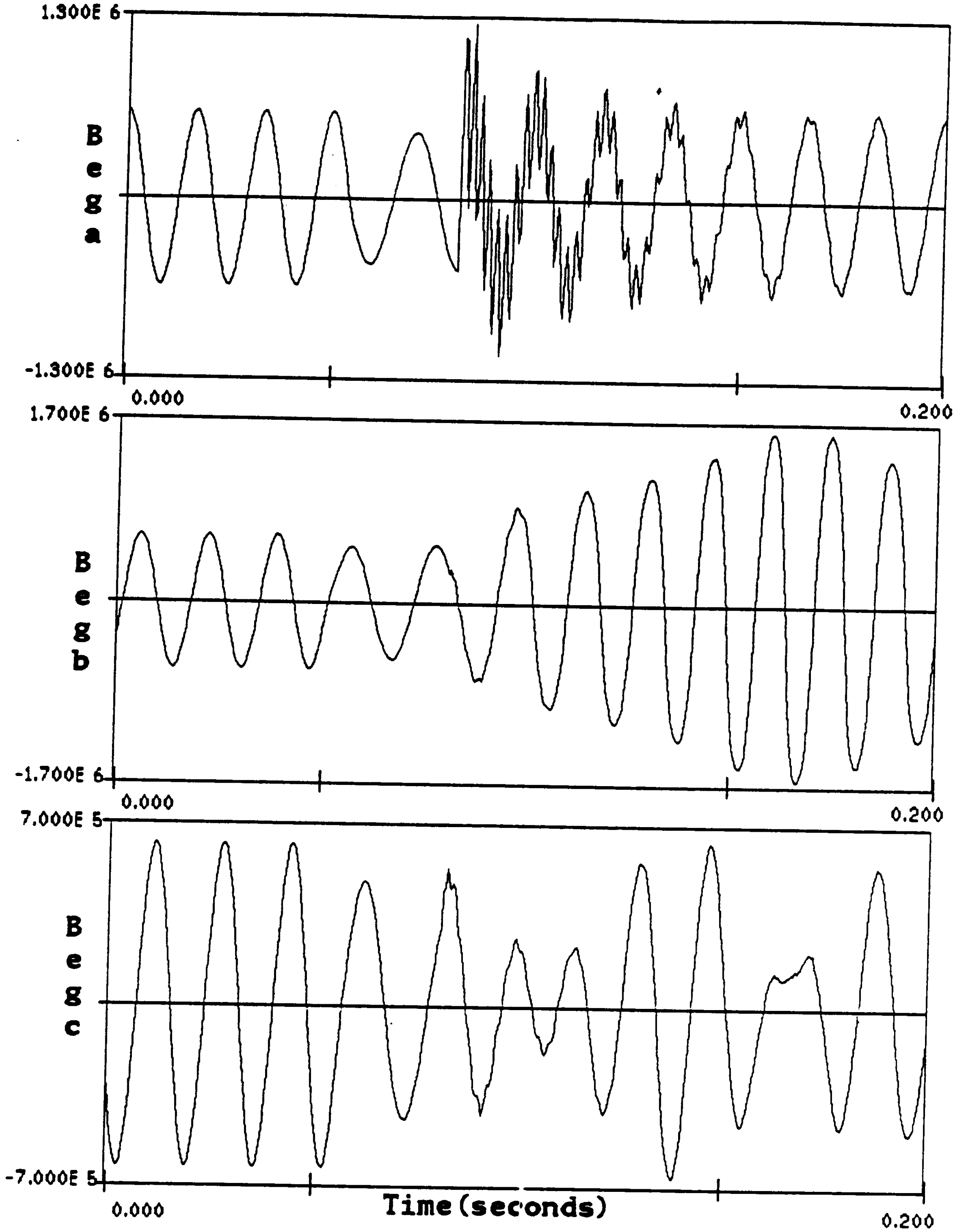


Figure 7: Line side voltages (normal case) transposed line, with reactor

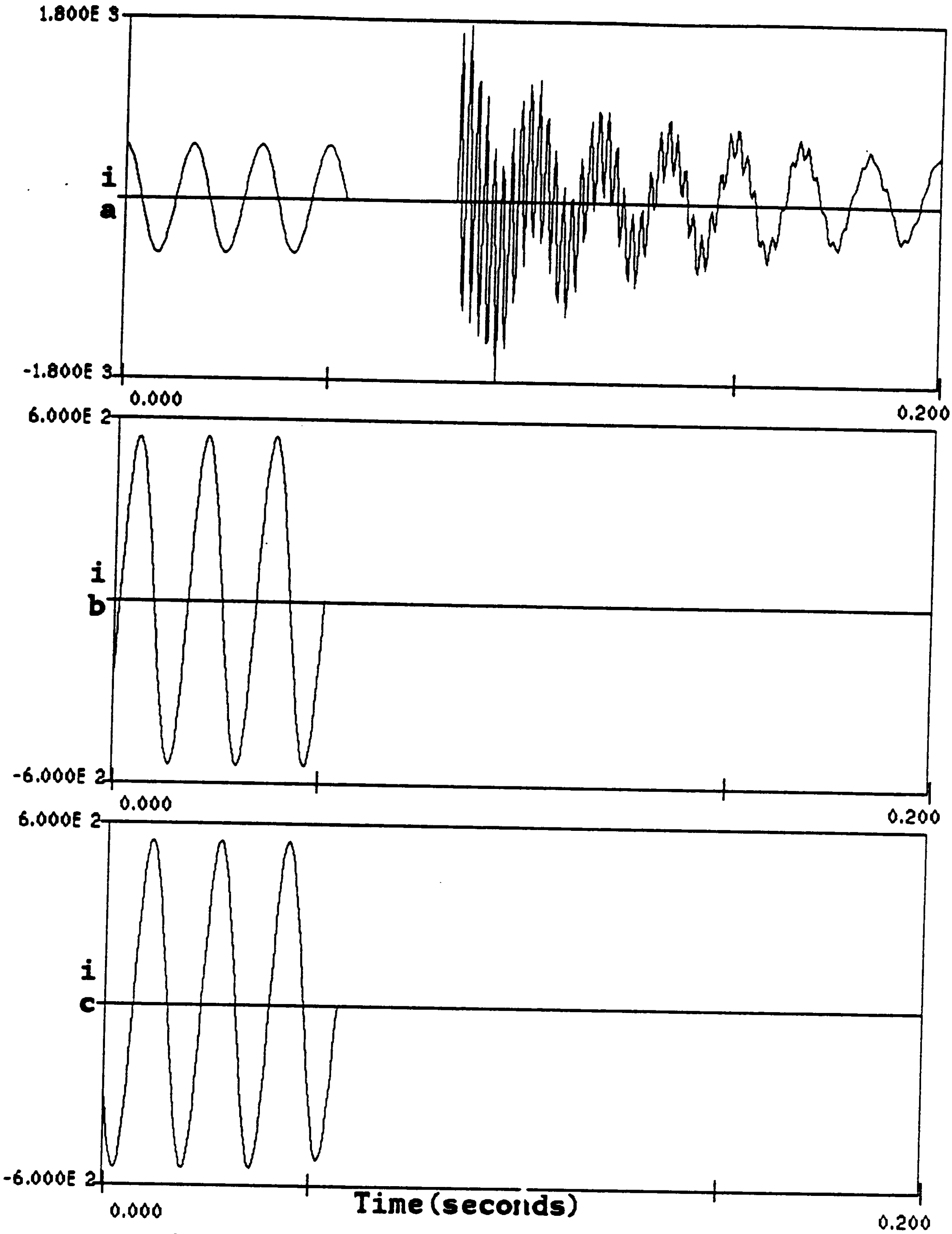


Figure 8: Line currents (normal case) transposed case, with reactor

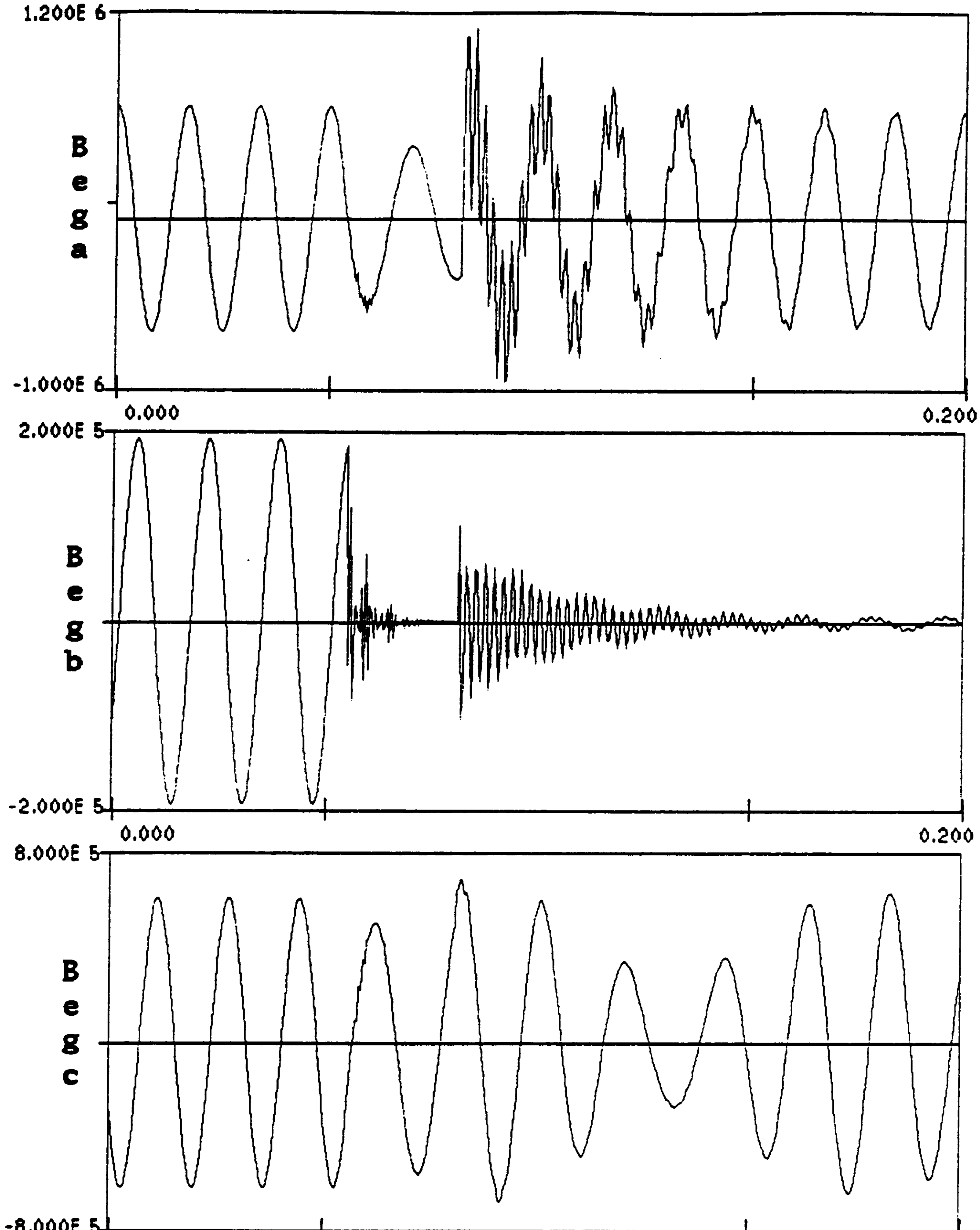


**Figure 9: Line side voltages (1-g fault)
transposed line, with reactor**

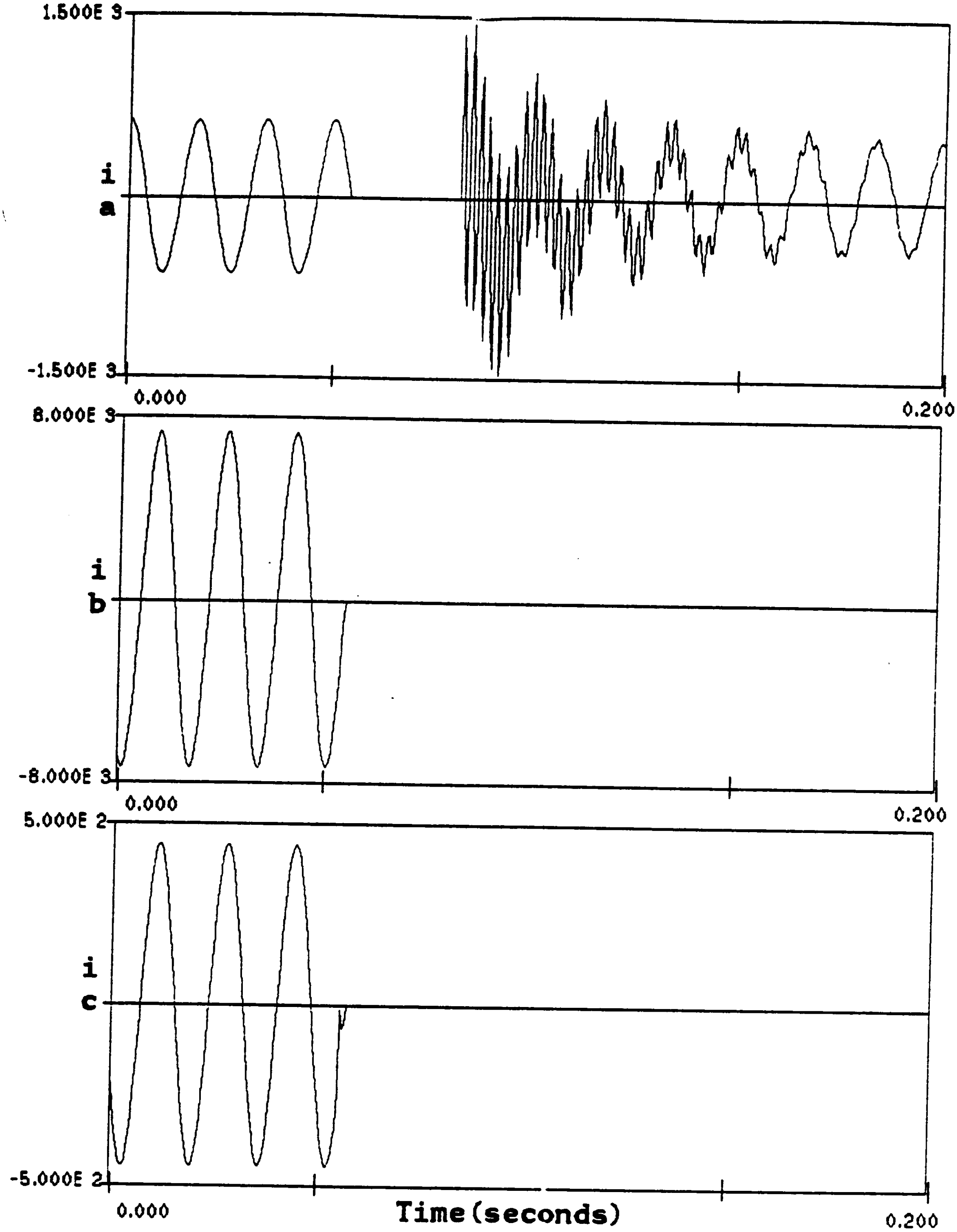


Figure 10: Line currents (l-g fault)
transposed lines, with reactor

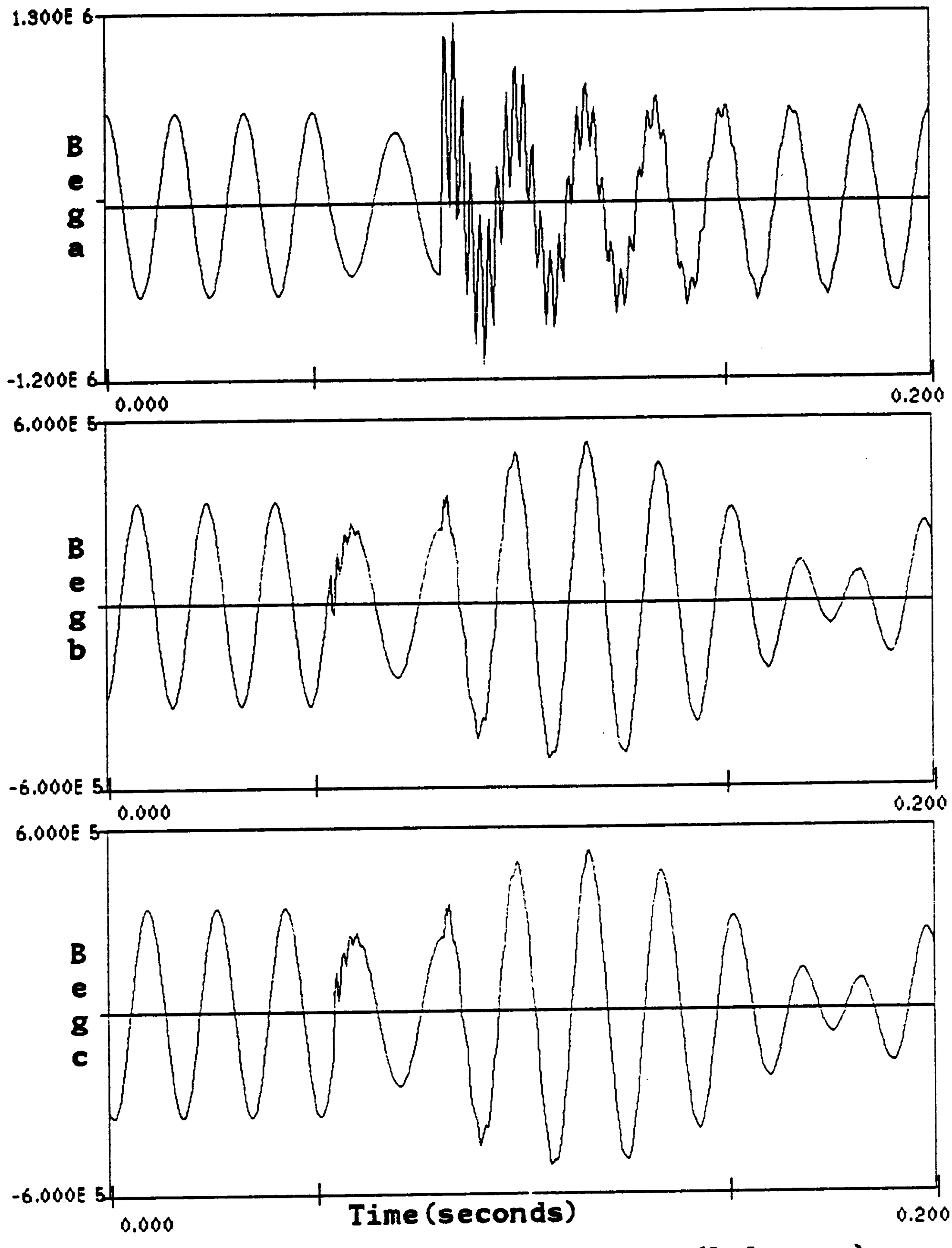


Figure 11: Line side voltages (1-1 case) transposed lines, with reactor

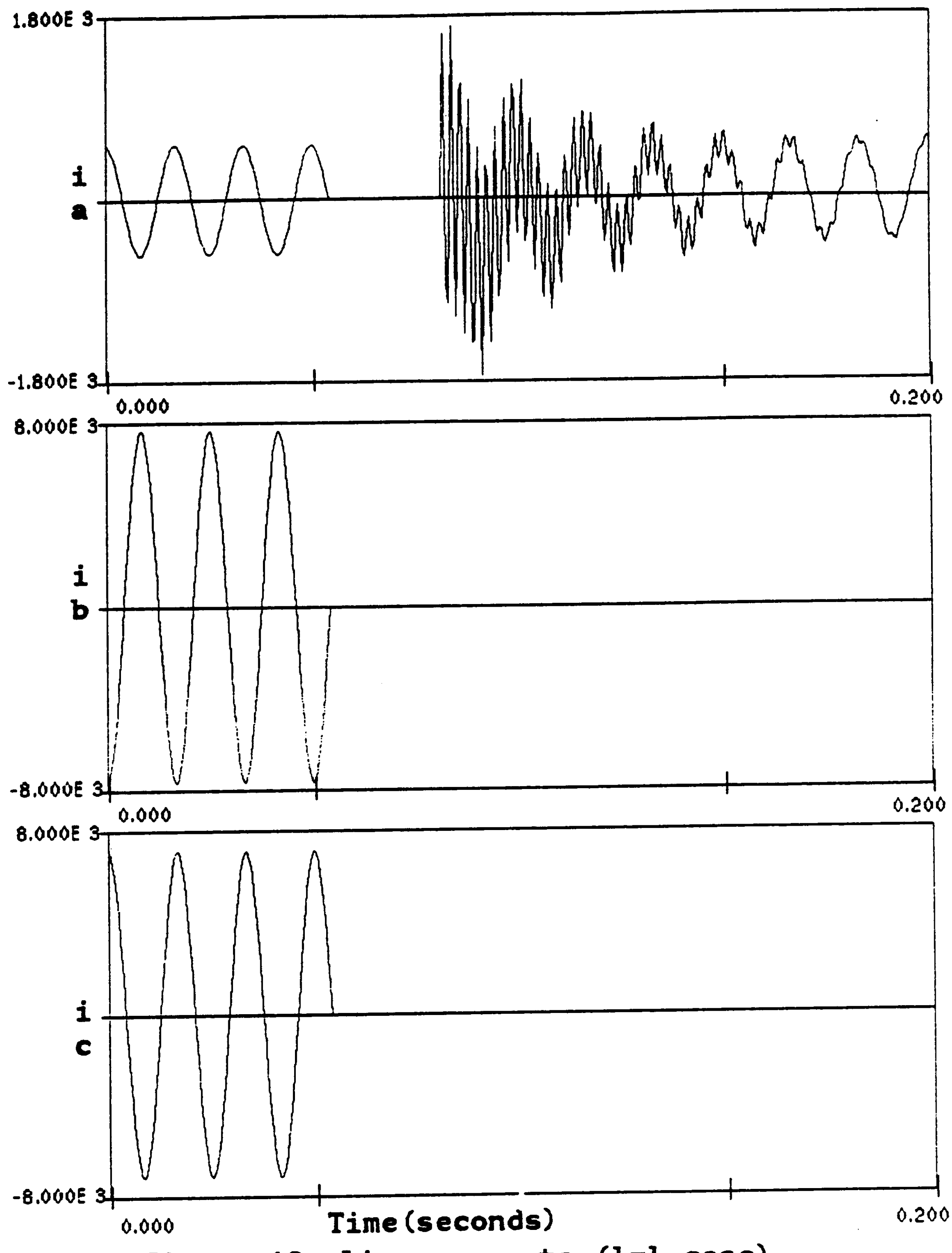


Figure 12: Line currents (1-l case)
transposed lines, with reactor

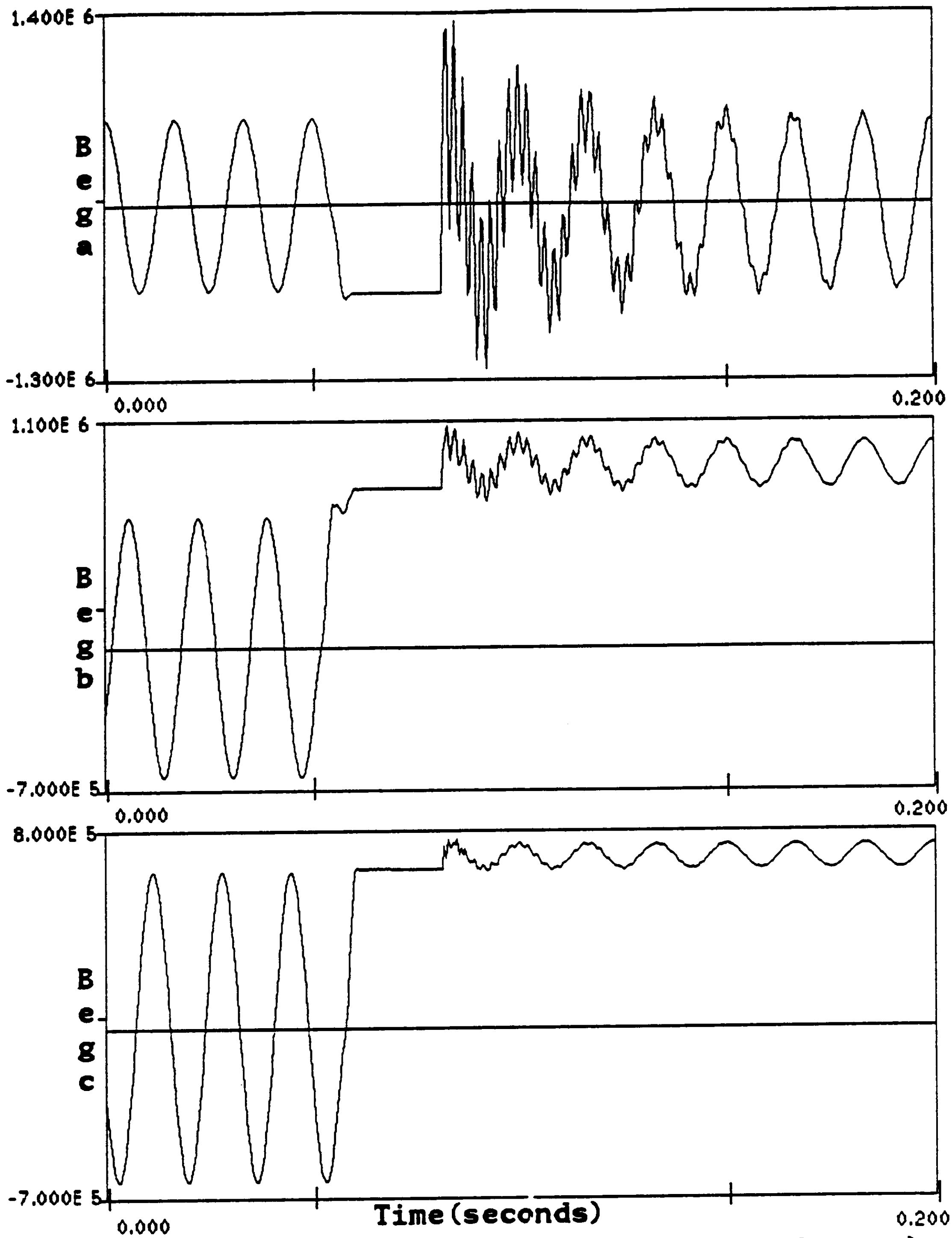


Figure 13: Line side voltages (normal case) untransposed lines, no reactor

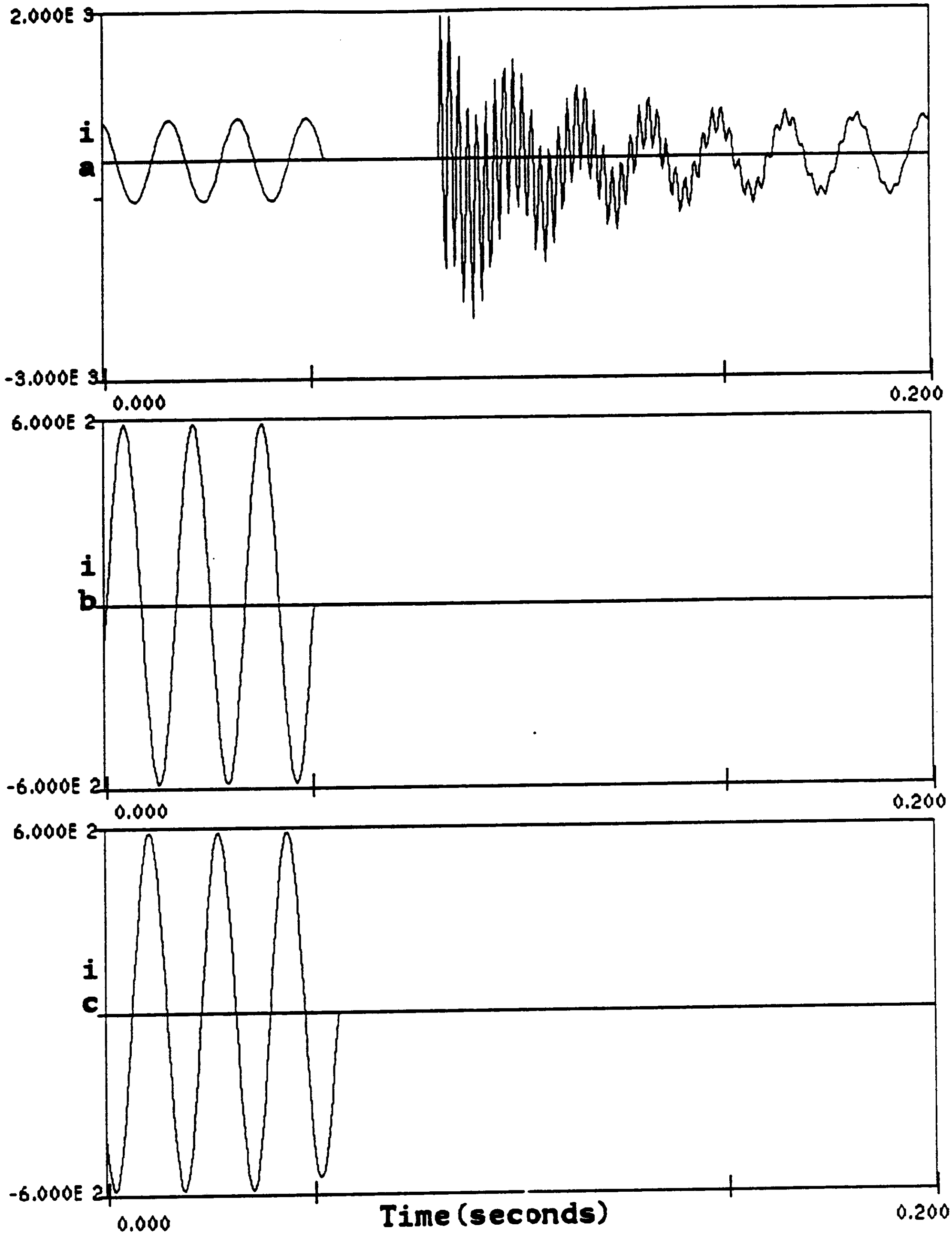


Figure 14: Line currents(normal case) untransposed lines, no reactor

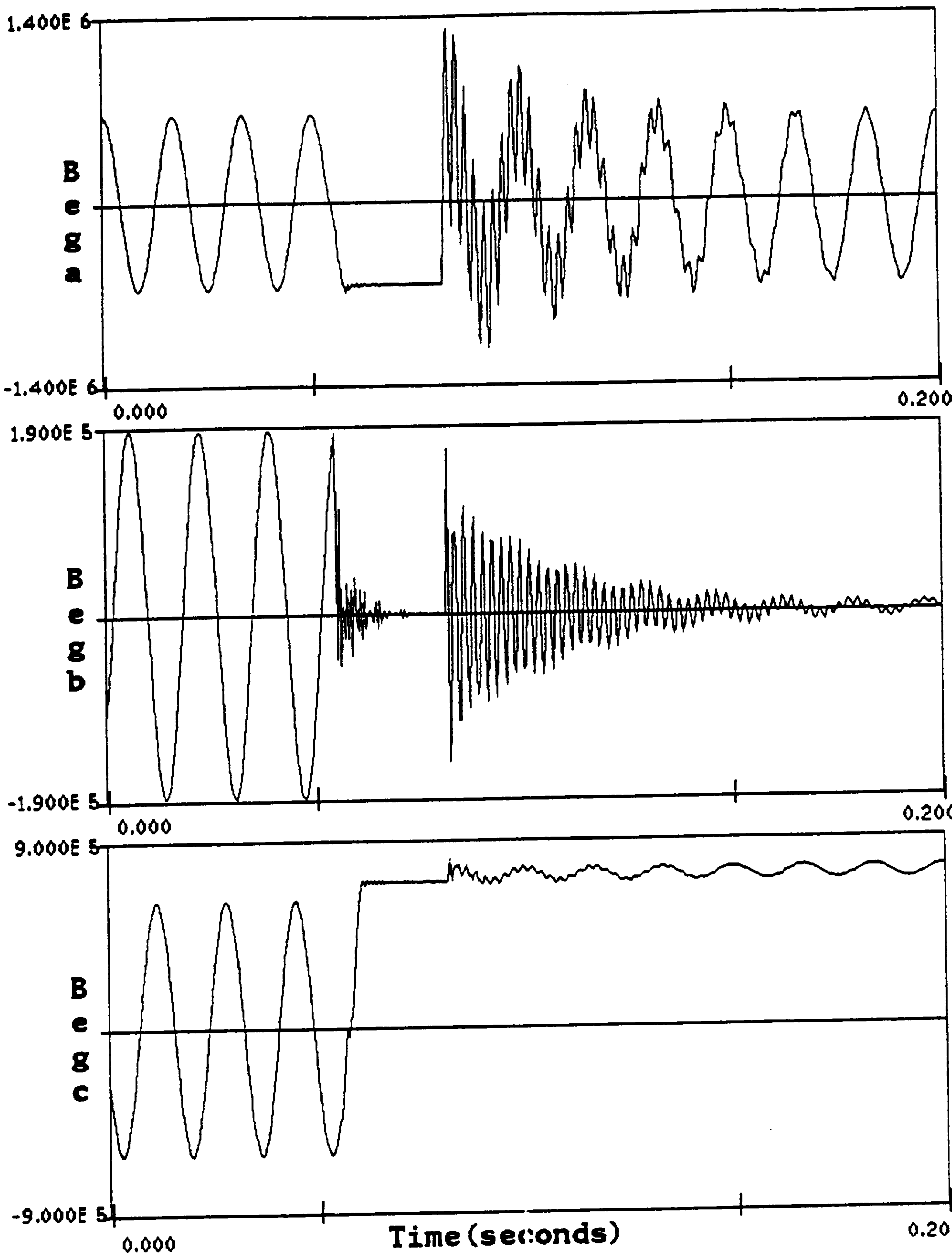


Figure 15: Line side voltages (l-g fault) untransposed line, no reactor

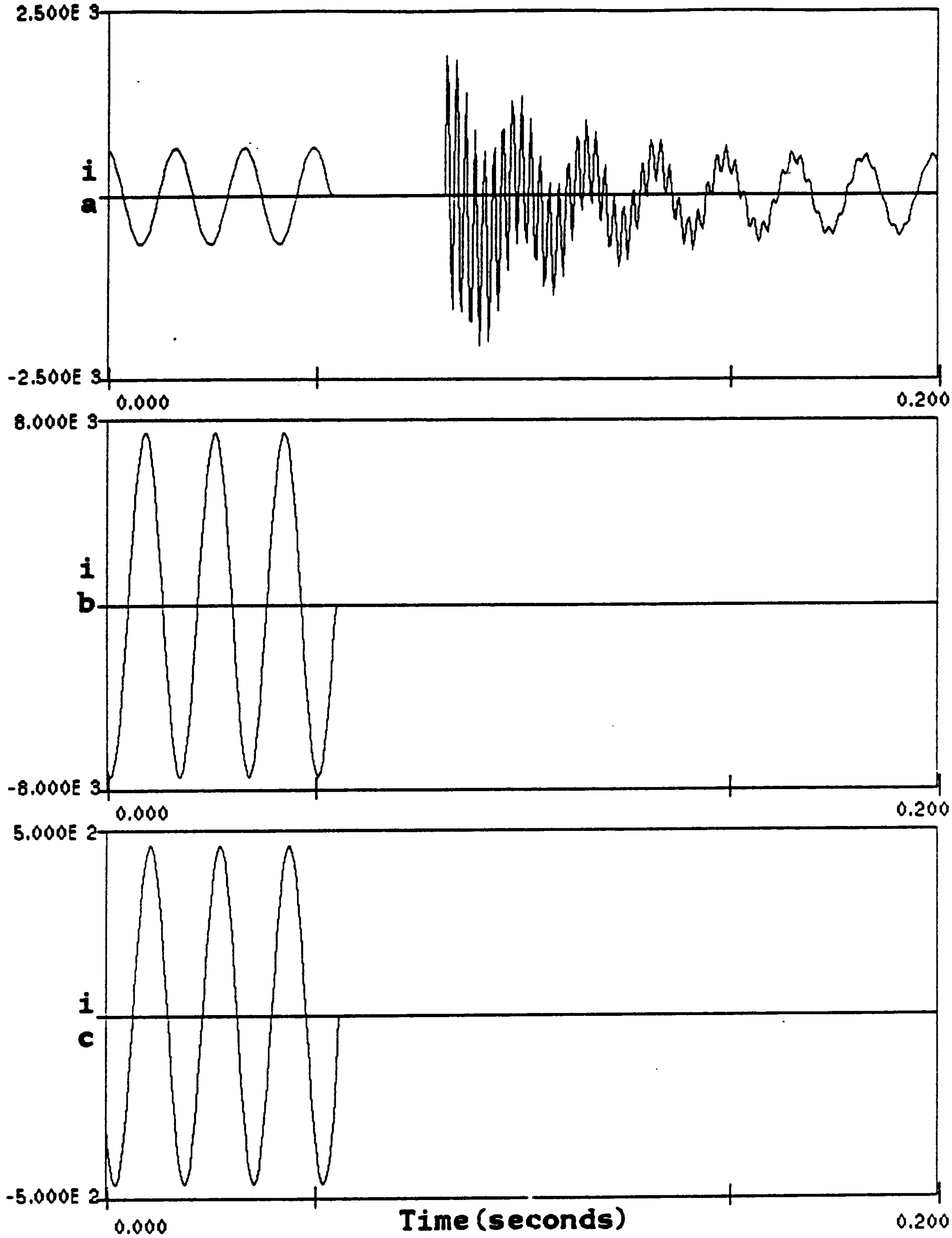


Figure 16: Line currents (1-g fault) untransposed lines, no reactor

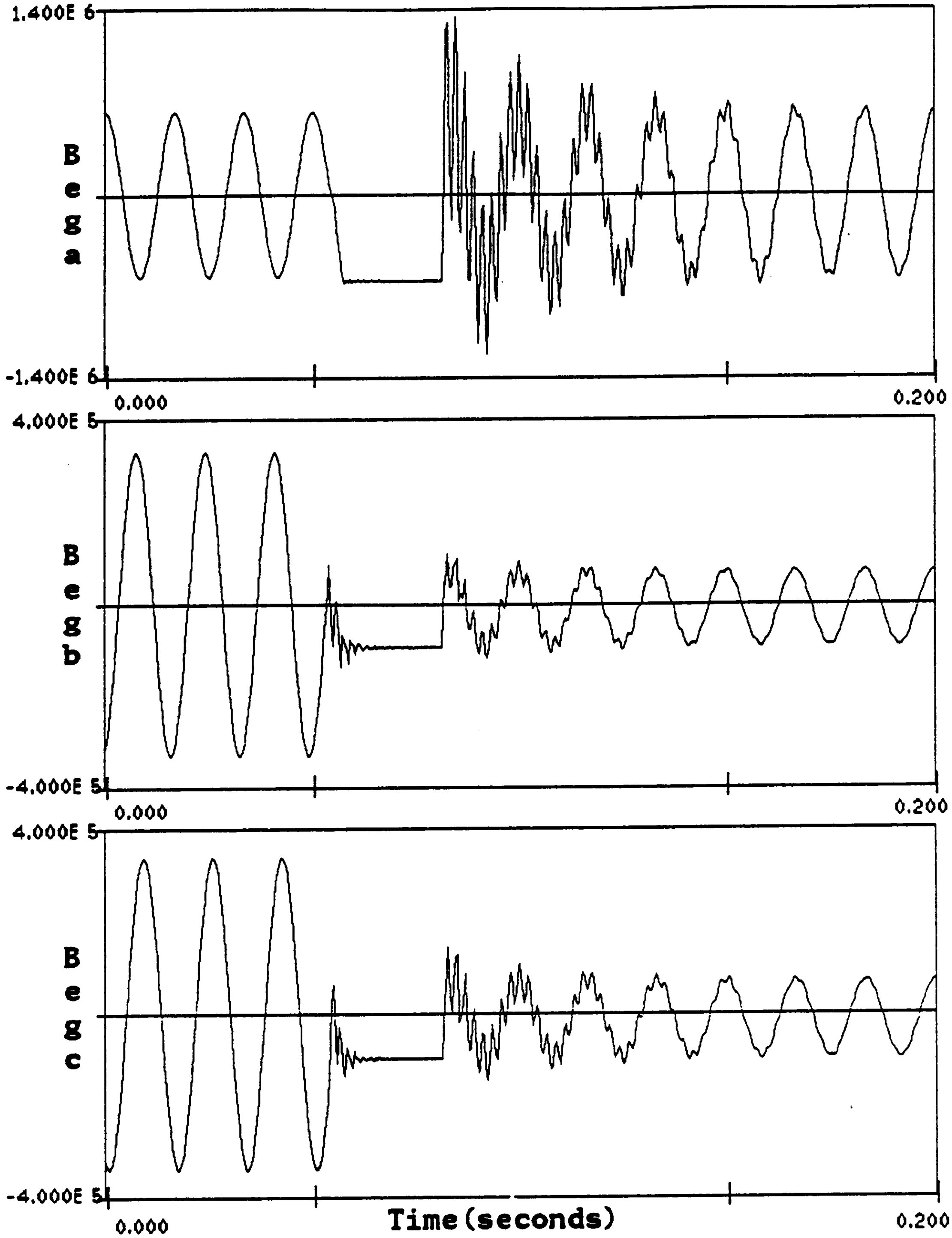


Figure 17: Line side voltages (l-l fault)
untransposed line, no reator

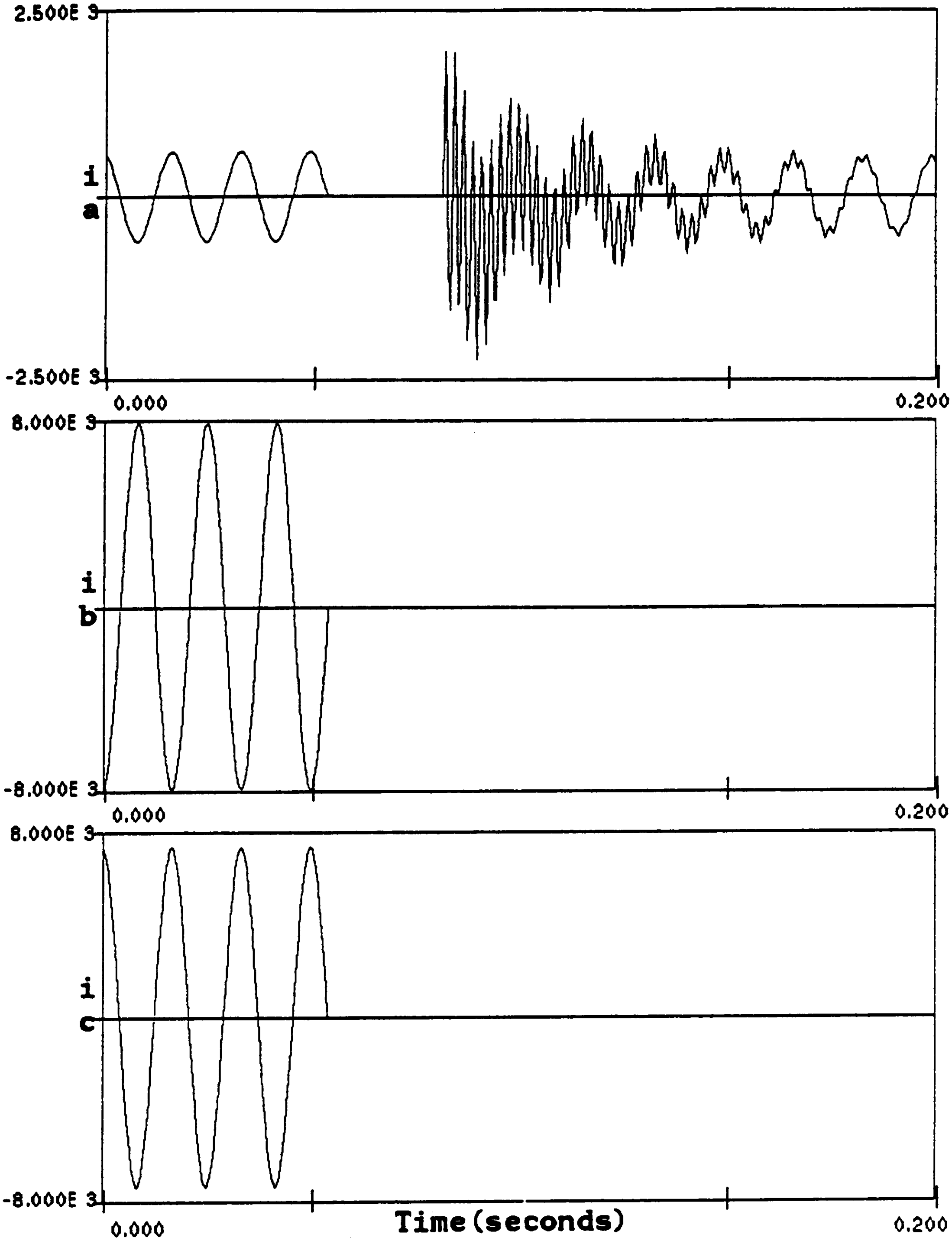


Figure 18: Line currents (l-l fault) untransposed lines, no reactor

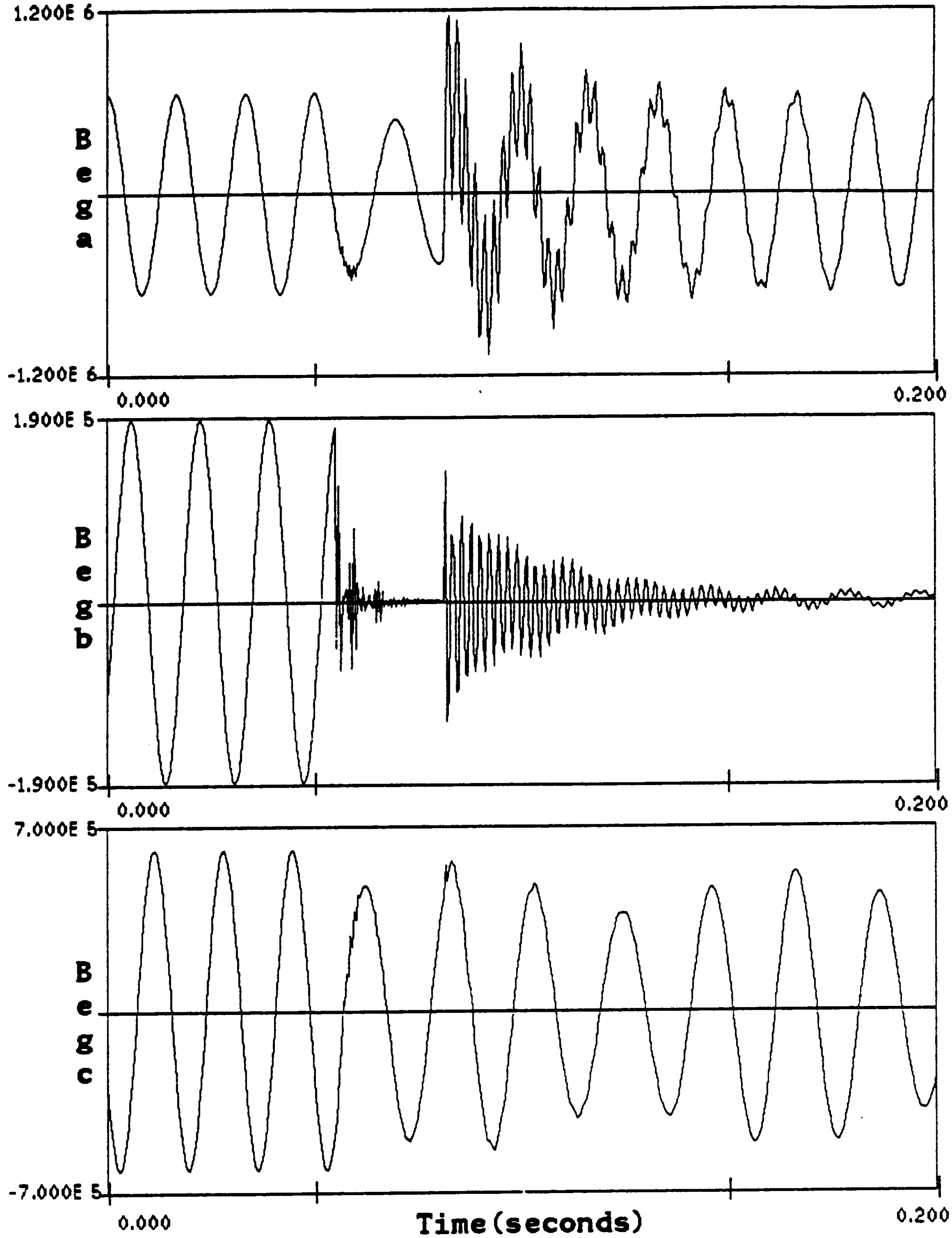


Figure 19: Line side voltages (1-g fault) untranspoxed line, with reactor

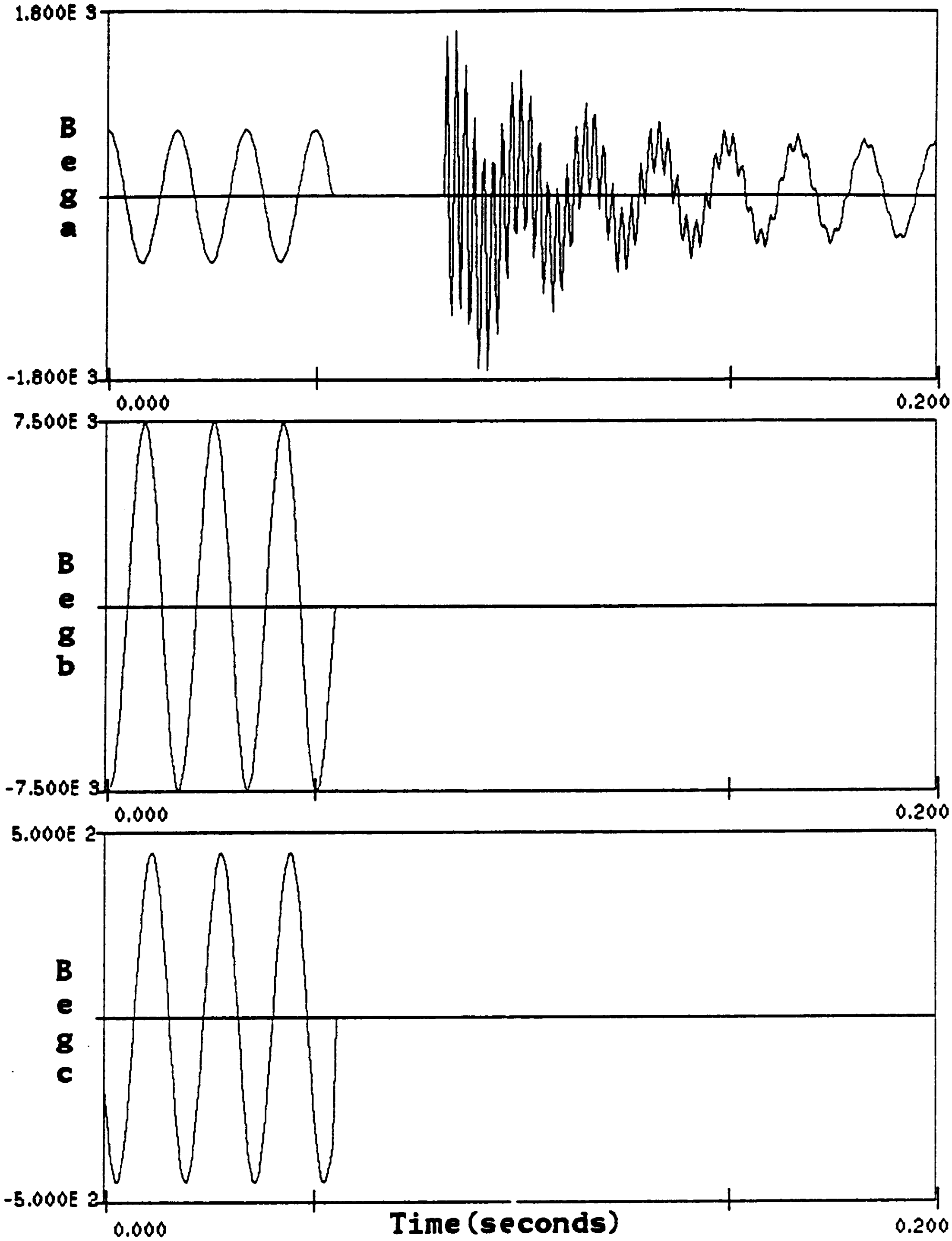


Figure 20: Line currents (1-g fault) untransposed lines, with reactor

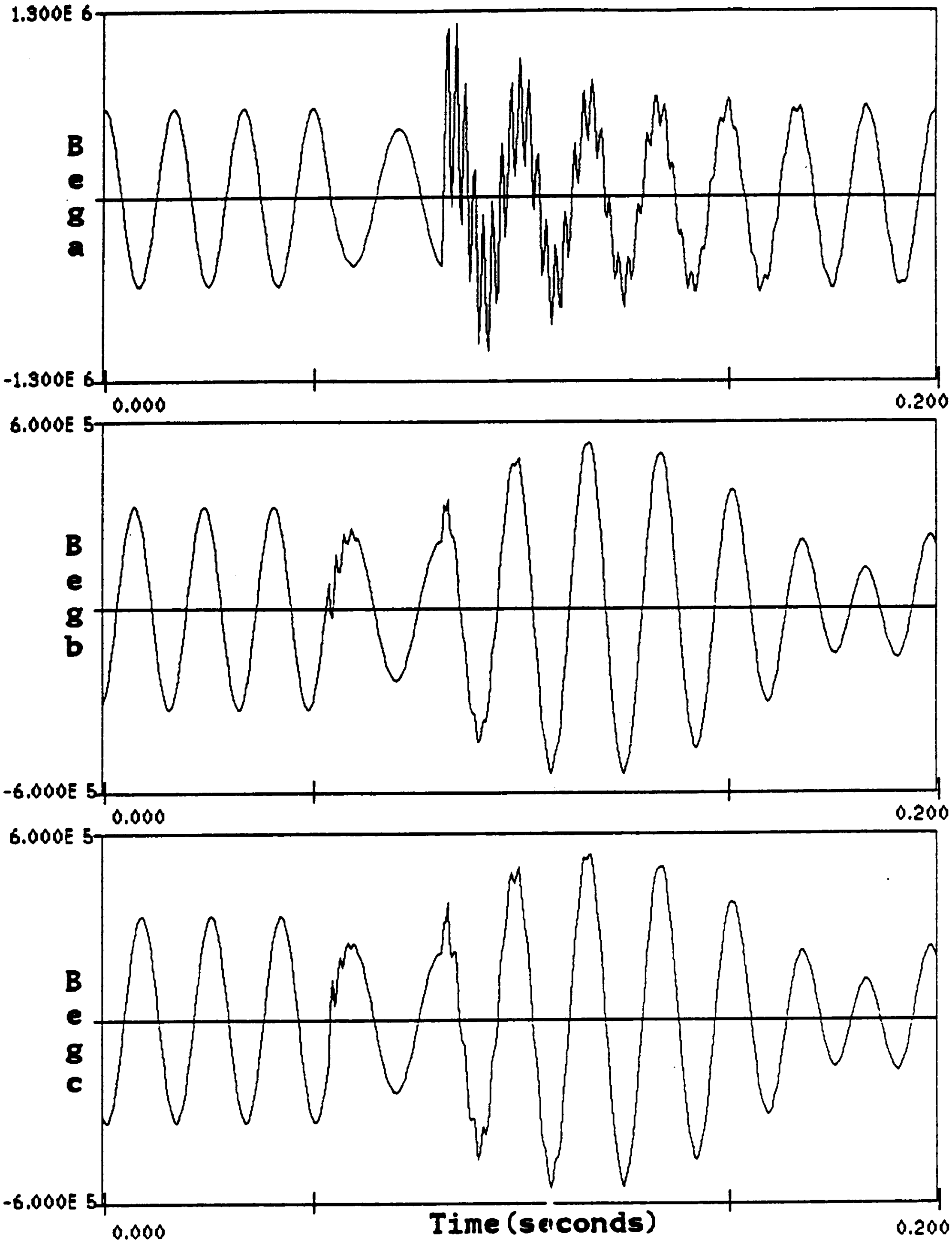


**Figure 21: Line side voltages (1-1 fault)
untransposed line, with reactor**

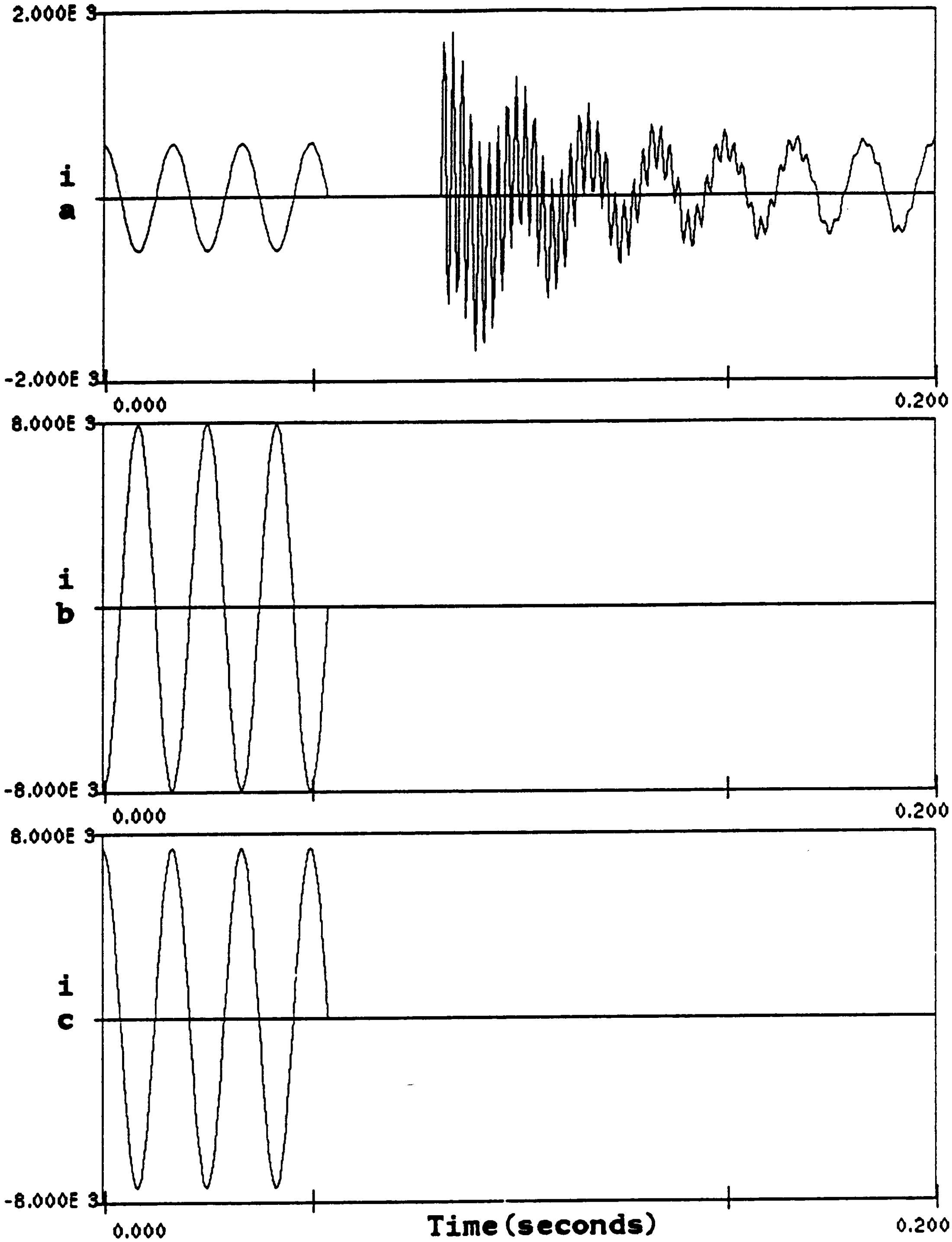


Figure 22: Line currents (l-l fault)
untransposed line, no reactor

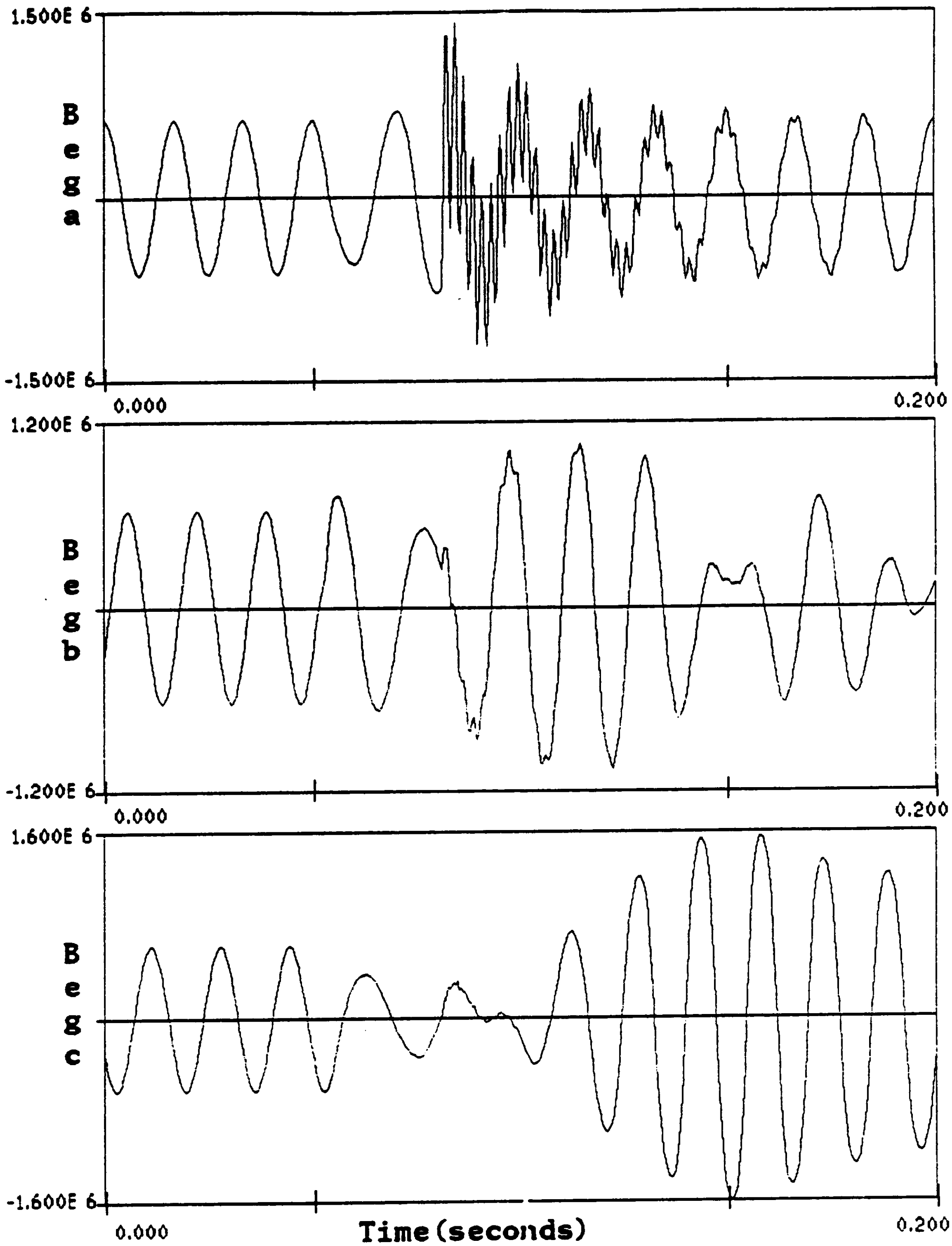


Figure 23 Line side voltages (normal case) untransposed lines, with reactor

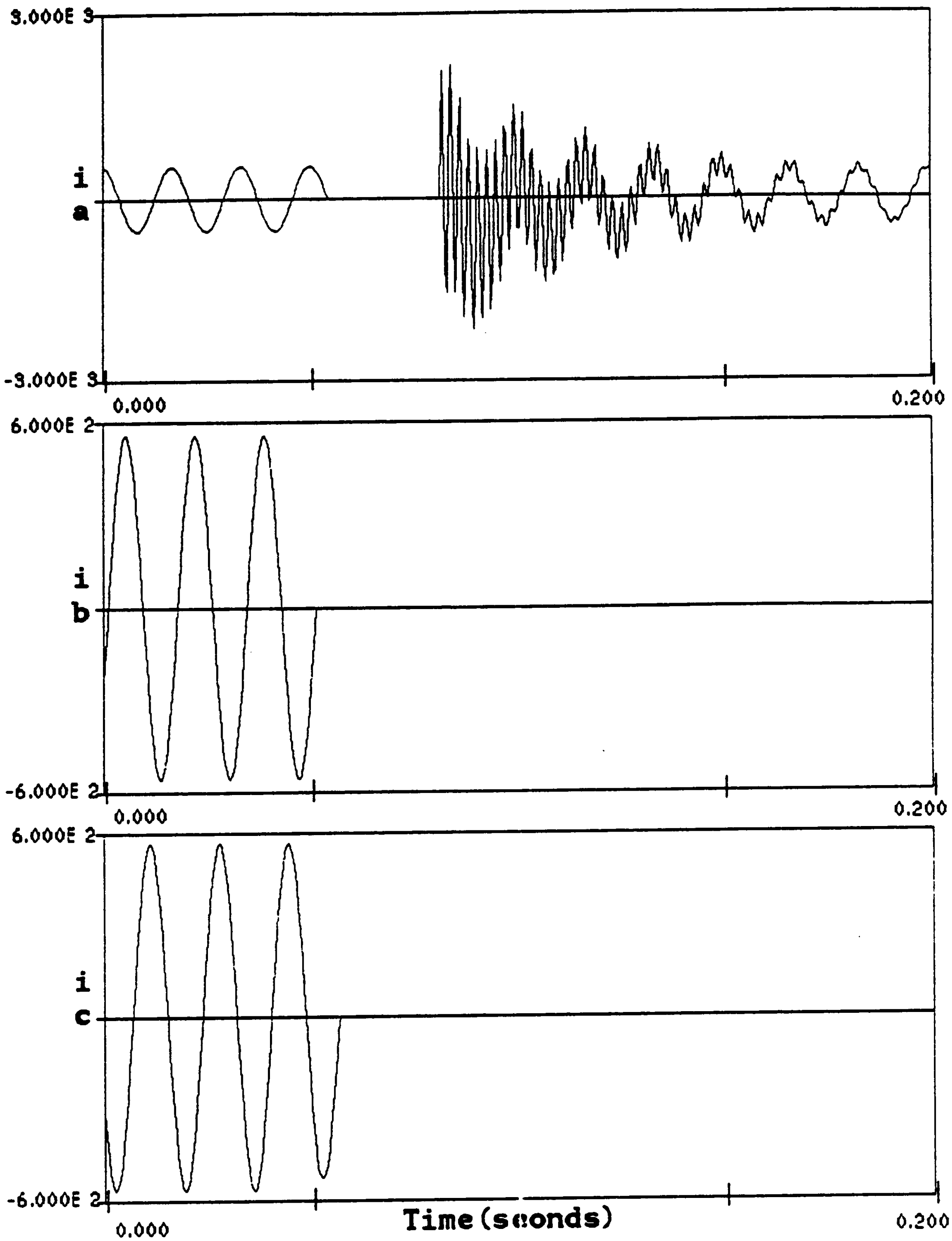


Figure 24 Line currents (normal case) untransposed lines, with reactor

The vita has been removed from the scanned document