Development and Validation of a Detailed Software Model for a Numerical Line Protection Relay

by

Todd E. Buchholzer

A Thesis submitted to the Faculty of Graduate Studies

in partial fulfilment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba, Canada

0-612-79937-9

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES
*****

DEVELOPMENT AND VALIDATION OF A DETAILED SOFTWARE MODEL FOR A NUMERICAL LINE PROTECTION RELAY

BY

TODD. E. BUCHHOLZER

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University

of Manitoba in partial fulfillment of the requirements of the degree

of

Master of Science

ABSTRACT

This thesis reports on the development and validation of a detailed software model for a numerical transmission line protection relay for use in the time domain simulation program PSCAD®/EMTDC™. The transmission line protection relay selected for modelling is the NxtPhase APT LPRO™ model 2100 firmware version 2.6b. Detailed software models for commercial line protection relays provide a convenient alternative to injection testing methods but are difficult to produce due to the proprietary nature of their design details.

A review of transmission line protection systems is provided. The functional characteristics of numerical overcurrent and distance line protection relay elements are explained as well as the generalized functional blocks within a typical numerical protection relay. Phasor based relay models and detailed software models are discussed and their applications explained. Some power system transients of interest driving the development of detailed relay software models are identified. The issues surrounding the development of detailed software models for commercial relays are also explored.

A performance objective is established to have the relay model perform with 100% accuracy for element operating decisions and within +/- 1 protection processing interval (+/- 2.1 ms) for element pickup and reset timing when compared to the physical relay for the same input signals. The model is developed with a detailed analysis provided for each functional block of the model. A model validation test plan is proposed to validate the response of 5 model protection elements to 25 different transmission line faults simulated in PSCAD/EMTDC. Five of these faults include varying degrees of current transformer saturation. The simulated currents and voltages applied to the model relay are recorded to a file in IEEE COMTRADE format and injected into the physical relay using a relay test set. The response of the model elements and relay elements are compared and the overall accuracy of the relay model assessed.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. Glenn Swift for agreeing to be both my Thesis advisor and key contact at NxtPhase in this endeavour. His assistance and encouragement along the way helped to keep me both motivated and focused. Thanks for making this an enjoyable project as well as an incredible learning experience!

I also extend my appreciation to the members of the examination committee Dr. Ani Gole and Mr. Dennis Woodford for their involvement and effort in the examination process.

I thank Dr. Peter McLaren for his guidance and help as my graduate program advisor.

Numerous people at NxtPhase assisted with the development and validation testing of the LPRO model. I admire the corporate culture at NxtPhase for understanding the benefits of developing this model. In particular, I would like to thank Dr. Zhiying Zhang, and Dean Weiten for their help with developing the model, and Dean Ouellette for his assistance with validation testing.

My fiancée Jennifer Pickell deserves special thanks for being understanding and supportive through this process. I most certainly would not have succeeded without Jen's assistance along the way to make the rest of my life run so smoothly.

Finally, thanks to my parents Patricia and Ernie for their unwavering support and encouragement through my many years of education.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Purpose for this Research

The purpose of this research is to develop and validate a detailed software model of a commercially available numerical transmission line protection relay for use in the time domain simulation program PSCAD®/EMTDC™ [1]. The transmission line protection relay selected for modelling is the NxtPhase APT LPRO™ [2] model 2100 firmware version 2.6b. The goal is to demonstrate that an accurate software model of a commercial numerical relay can be developed by an independent model developer using a combination of public domain information and certain proprietary details from the relay manufacturer. The author selected this particular relay because of NxtPhase's strong commitment to support the proposed model development including the disclosure of design details normally not released into the public domain.

1.2 The Need for Detailed Software Models

Modern numerical transmission line protection relays are functionally complex devices using sophisticated signal processing techniques and complicated protection logic. These protective relays offer a large number of protection elements and logic functions that operate on highly processed current and potential signals from the power system. Protection engineers may need to deal with numerous makes and models of commercially available numerical relays from various manufacturers each of which may use very different operating principles. Understanding every detail of the complex signal processing and protection algorithms that these relays use is a daunting task on its own.

[1] PSCAD is a registered trademark of the Manitoba HVDC Research Centre and EMTDC is a trademark of Manitoba Hydro. PSCAD (Power System Computer Aided Design) is a graphical interface used to build power system models using "drag and drop" elements from a component library. EMTDC (Electromagnetic Transients DC) is a general purpose time domain simulation program used to solve the systems created in PSCAD. The PSCAD/EMTDC integrated software tool is marketed by the Manitoba HVDC Research Centre

[2] LPRO is a trademark of NxtPhase Inc, a supplier of the APT line of numerical protection relays.

Trying to accurately calculate or even estimate the response of these relays to transient current and voltage signals is almost impossible. The sophisticated technology behind these relay platforms presents a terrific challenge to the protection engineer when assessing a relay for a particular application, determining relay settings, or diagnosing a relay mal-operation.

In order to maintain stability in heavily loaded power systems, modern relays are often required to identify power system faults as fast as one power system cycle (1/60th of a second) or less, and determine if a circuit breaker should or should not be tripped. Following the occurrence of a fault, a relay's current and potential signals are very likely to contain significant non fundamental signal components due to effects such as exponential offset, fault noise, or instrument transformer induced transients. Ironically, most power system relays including numerical relays are set by protection engineers based only on steady state fundamental conditions determined from phasor based fault studies. Numerical relays which make operating decisions within the first one or two cycles are doing so with current and potential signals rich in transient components.

While these relays are designed with the intention of filtering out undesirable transients and signal components, it is not always predictable what signals a relay may be subjected to, or how the relay's various signal processing and protection algorithms will respond. Some typical concerns surround the transient reach of overcurrent or distance elements, how different polarizing choices may effect operation, the transient security of directional elements, or the response of complex protection logic using numerous element inputs.

Investigating a relay's response to a particular system disturbance or fault can be done by applying the disturbance to the real power system including the protection system under investigation. This method of applying "staged faults" is extremely undesirable due to its impact on the real power system not to mention the cost and complexity involved when numerous events require investigation. More commonly, testing of the relay or protection system is accomplished by simulating the power system signals with time domain simulation programs such as EMTP/ATP [3] or PSCAD/EMTDC and saving the relay current and potential signals to a file. Later the current and potential signals are played back (injected) to the relay in real time using a specialized injection test set. Alternatively, the power system signals may be simulated and injected into the relay directly in real time using a power system real time simulator. Protection engineers use both the real time file playback and the real time simulator "injection testing" methods today to assess relay performance to real world signals. Both real time file playback and real time simulator testing of relays can be an expensive, laborious, and time consuming process involving specialised test equipment to carry out the tests. For these reasons both methods can often only be justified in the most extreme cases where relay operating margins are critical or the application is unique.

Often protection engineers would like to study the suitability of a relay for particular applications where there are unique application considerations but cannot justify the additional expense and time required for real time file playback or real time simulator testing. For these cases a detailed relay software model is very valuable. Such a model ideally runs within a time domain simulation program such as EMTP or EMTDC. The simulation program can accurately determine the relevant power system signals and at the same time simulate the relays response to these same signals. Alternatively, real system fault data from a digital fault record file can be played back into a software model to assess relay performance to the disturbance, or to analyze a mal-operation of a relay. Models also offer the additional advantage of being able to look inside at internal relay quantities to assess margins of operation. A detailed relay software model therefore affords the protection engineer the convenience to do away with complicated injection based tests and accurately evaluate the performance of a relay to any number of system disturbance events using only a personal computer.

Relay software models have existed in various forms for some time and are used by manufacturers for relay development, by utility protection engineers in application

[3] EMTP (Electromagnetic Transients Program) and ATP (Alternative Transients Program) are general purpose time domain simulation programs.

studies, and by academics for research and educational tools [1], [2], [3]. The level of detail in modelling can vary greatly from a simple phasor based equation set, to a complex time domain simulation subroutine. The level of modelling detail is ultimately justified by the model user's particular requirements.

There are presently many commercially available software packages providing phasor based models for commercially available protective relays. Detailed transient software models running in time domain simulation programs have been developed successfully by academics for relays built in house where they have full access to all the design details [2], [7]. Few such detailed transient models exist however for commercial relays. Relay manufacturers are generally not anxious to release the proprietary details of their relays, in particular the software coding or "firmware", which are required to build detailed transient models. Transient relay models presently existing in the public domain are most often developed based only on generic protection algorithms [1], [3]. Such transient models are valuable for educational purposes, but are of little value to practicing protection engineers. The limited availability of information on commercial relay design details constrains model developers from building accurate and therefore reliable software models for relays applied in power systems.

1.3 Scope of Work

A review of transmission line faults and transmission line protection systems is provided. Basic line protection relay elements are explained as well as the generalized functional blocks within a typical modern numerical protection relay.

Phasor based relay models and detailed software models are discussed and their applications explained. Some of the power system transients of interest driving the development of detailed relay software models are identified. The issues surrounding the development of detailed relay software models are also explored.

A model strategy is proposed which separates the LPRO software model into several functional blocks. A desirable performance objective for the model is proposed. The LPRO model is developed with a detailed analysis provided for each functional block of the model. In order to ensure a reasonable amount of work for this Thesis, the LPRO model will include only instantaneous overcurrent elements and 2 zones of phase and ground distance elements along with all their associated supervisory elements.

A model validation test plan is proposed to validate the response of 5 LPRO model protection elements to 25 different transmission line faults simulated in PSCAD/EMTDC. Five of these faults include varying degrees of current transformer saturation. The simulated currents and voltages applied to the model relay are recorded to a file in IEEE COMTRADE format and played back into the physical relay using a relay test set. The behaviour of the 5 model elements and relay elements are directly compared and the overall accuracy of the relay model assessed.

2. NUMERICAL TRANSMISSION LINE PROTECTION RELAYS

2.1 Power Transmission Lines and Their Protection

Transmission lines are used to route electrical power to geographic areas where it is required. Transmission lines can vary in length from a fraction of a kilometre to hundreds of kilometres and operate at voltages up to 1000kV or more. Most transmission lines are of the "overhead" type constructed on towers or poles with 3 bare phase conductors attached to the tower by means of a connecting "insulator". Often more then one transmission line may share the same tower.

Overhead transmission lines by nature of their exposed design are vulnerable to faults. A fault is any event where the insulation of any or all of the high voltage conductors fails. This typically results in a large magnitude current flow, often through an arc, from the faulted conductor to either another of the high voltage conductors or to ground. Each high voltage transmission line conductor is electrically insulated from others in the vicinity by the air surrounding it, and insulated from the grounded tower by its attaching insulator. The events which lead to a failure of the air or attachment insulators are numerous including such things as overvoltage from lightning strikes, inadvertent contact with grounded objects such as trees, deteriorated insulators, or tower structural failures. Once a fault has occurred on a transmission line, the only remedy is to de-energize the faulty line to extinguish the fault.

Devices that detect faults on the transmission line are called line protective relays, but are often commonly referred to as either the " line protection", or more generically the "relay". The role of the line protection relay is to quickly identify a faulty line and initiate the disconnection of the line from the greater power system. Quick disconnection of faulted transmission lines is required to ensure human safety, prevent power system instability, and to prevent damage to system apparatus including the line itself.

The entire transmission line protection system is composed of three main components including the instrument transformers, the transmission line protection relay, and the power circuit breaker as shown in Figure 2.1 [4]. The instrument transformer's role is to reduce the large magnitude voltage or current signals on a transmission line to a level suitable for input into a protective relay. The line's "primary" currents and voltages are scaled down to low level or "secondary" currents and potentials by the current transformer (CT) and potential transformer (PT) respectively. The transmission line protection relay uses the secondary current and potential quantities to decide if the transmission line is faulted. If the relay decides the line is faulted, it initiates a "trip" to the transmission line's power circuit breaker(s). The power circuit breaker must then operate to disconnect the faulty line from the greater power system. The relay is made to distinguish between normal operating conditions and faulty conditions by means of its applied relay settings. The protection engineer determines relay settings based upon analytical system studies and the relevant setting philosophies for the application.

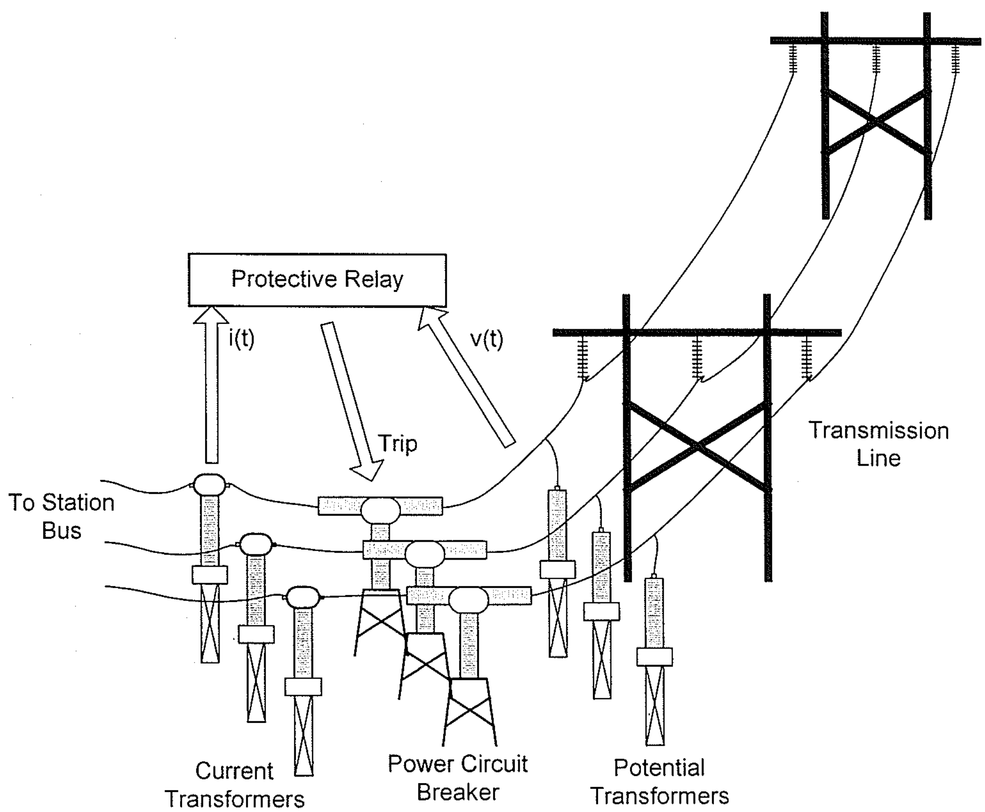


Figure 2.1: A typical transmission line protection system.

Transmission line protection can be broken down into non-pilot schemes (single-ended) or pilot schemes (double ended). Non-pilot scheme relays use only the line end current and potential signals to make tripping decisions based on current magnitudes and/or measured impedance. Pilot schemes use information from both ends of the line transmitted back and forth over a communication channel. Many numerical transmission line protection relays include both pilot and non-pilot schemes. This discussion will be limited to non-pilot transmission line protection.

2.2 How Numerical Transmission Line Protection Relays Detect Faults

Numerical transmission line relays detect power system faults by monitoring line end current and potential quantities to determine if the line is faulted according to the rules of its protection algorithms and the user applied settings. The numerical relay's hardware provides the means with which to process the incoming current and potential signals and provide the required input/output (I/O) capability. The hardware of numerical relays is discussed in the following section. The software coding or firmware applied to the numerical relay provides the signal processing methods and protection algorithms required to process the incoming current and potential signals and decide whether or not the relay should trip the circuit breaker.

A typical firmware block diagram is illustrated in Figure 2.2. The discrete line current $i[n]$ and potential $v[n]$ signals are obtained form the relay hardware's analogue to digital (A/D) converter. These discrete signals are processed numerically to filter out undesirable signal components and extract signal quantities such as fundamental frequency phasors, root-mean-square (RMS) magnitudes, and symmetrical component phasors. Fundamental frequency phasors are the most common quantity of interest and can be extracted using techniques such as the Discrete Fourier Transform (DFT) or Walsh Functions [5]. Both RMS and symmetrical components are easily extracted using well established procedures [6]. These various processed signal quantities are fed to the protection algorithms to determine if the line is faulted.

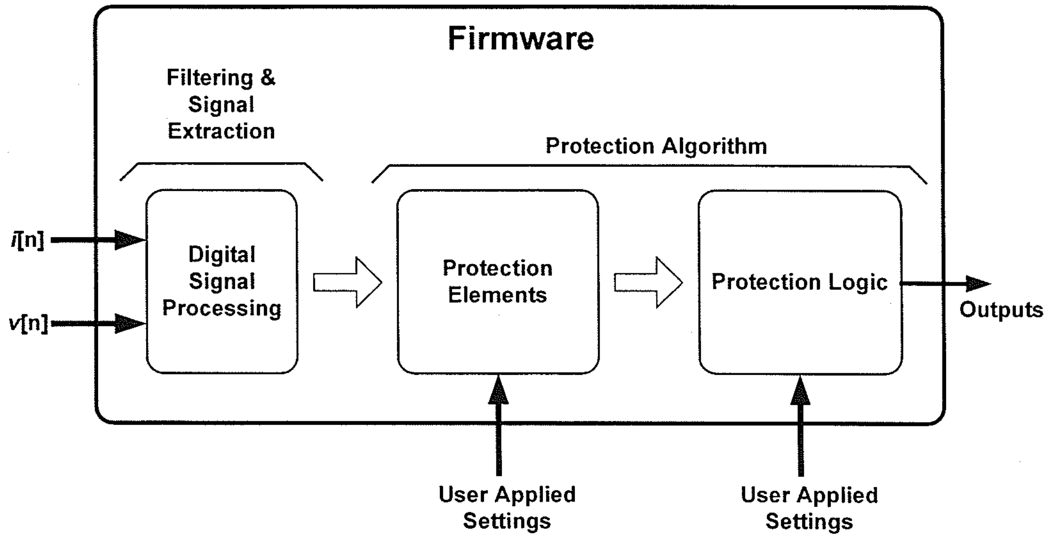


Figure 2.2: A typical numerical relay firmware functional block diagram.

The relay's protection algorithm can be subdivided into two distinct components which include (1) the protection elements, and (2) the protection logic. The first component, the protection elements, are functions which use the signal quantities discussed above to determine if a specific threshold has been exceeded. The threshold level is usually established by a user applied setting. Protection elements are generally two state devices. If the element's operating threshold is not exceeded, the element is said to be in the "reset" state. If the threshold has been exceeded than the element is said to be in the "operated" state. It should be pointed out that the "operation" of an element does not by itself necessarily imply the relay will "trip" and the two terms should not be used interchangeably.

The second major component of the protection algorithm, the protection logic, uses the state of the individual protection elements along with basic logic gates, timers, and user applied settings to decide if the relay should indeed trip. Protection logic is also used to implement auxiliary functions such as loss of potential (LOP) and dead line pickup.

The two most common line protection elements are (1) level detection elements which determine if a measured signal quantity is over or under a certain level, and (2) distance elements which test if the line end apparent impedance phasor falls within a characteristic shape in the resistance(R)-reactance(X) plane (R-X plane) [4]. Of the level detection and distance element types just described, two stand out as being the most commonly available and widely used in transmission line protection. These are the overcurrent level detection element and the mho distance element.

2.2.1 The Overcurrent Level Detection Element

The overcurrent level detection element is used to determine if the magnitude of a current signal exceeds the threshold level established by the user applied setting. Overcurrent elements that operate without intentional delay are referred to as "instantaneous" and enter the "operate" state as soon as the magnitude of the measured current signal exceeds the applied setting. Time overcurrent elements start their operation when the applied setting is exceeded but operate with a time delay that is determined by the particular time-current characteristic. Figure 2.3 illustrates the process. Typical numerical relay overcurrent elements operate on fundamental RMS current magnitude, true RMS current magnitude, or positive, negative, or zero sequence symmetrical component RMS current magnitudes.

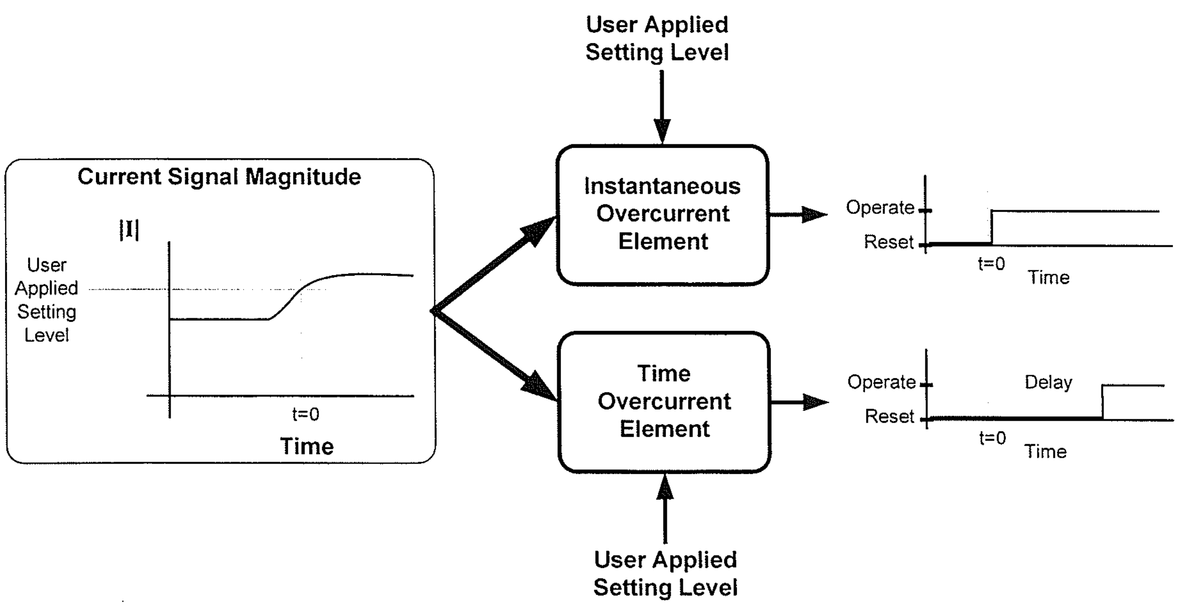


Figure 2.3: The numerical overcurrent level detection element.

2.2.2 The Mho Distance Element

The mho distance element determines if the measured line end apparent impedance phasor falls within a characteristic mho circle shape as shown below in Figure 2.4. The mho element uses the line end current and potential phasors to measure the protected line's impedance from the relay location. Normal loading impedances will appear large and plot far outside the mho circle. A line fault however will typically reduce the impedance measured by the relay to some value less than the impedance of the line itself. The size of the mho circle is determined by a user applied "reach" setting usually based on a multiple of the protected transmission line's positive sequence impedance.

In Figure 2.4, the mho circle size or "reach" of the element is set at 80% of the transmission line impedance $\mathbf{Z_{T\text{-}LINE}}$. Line faults within 80% of the line length from the relay location should cause the mho element to operate providing the fault resistance is not excessive. Figure 2.4 shows impedance phasor $\mathbf{Z_{LOAD}}$ plotted for a typical normal load impedance, and phasor $\mathbf{Z_{FAULT}}$ plotted for a

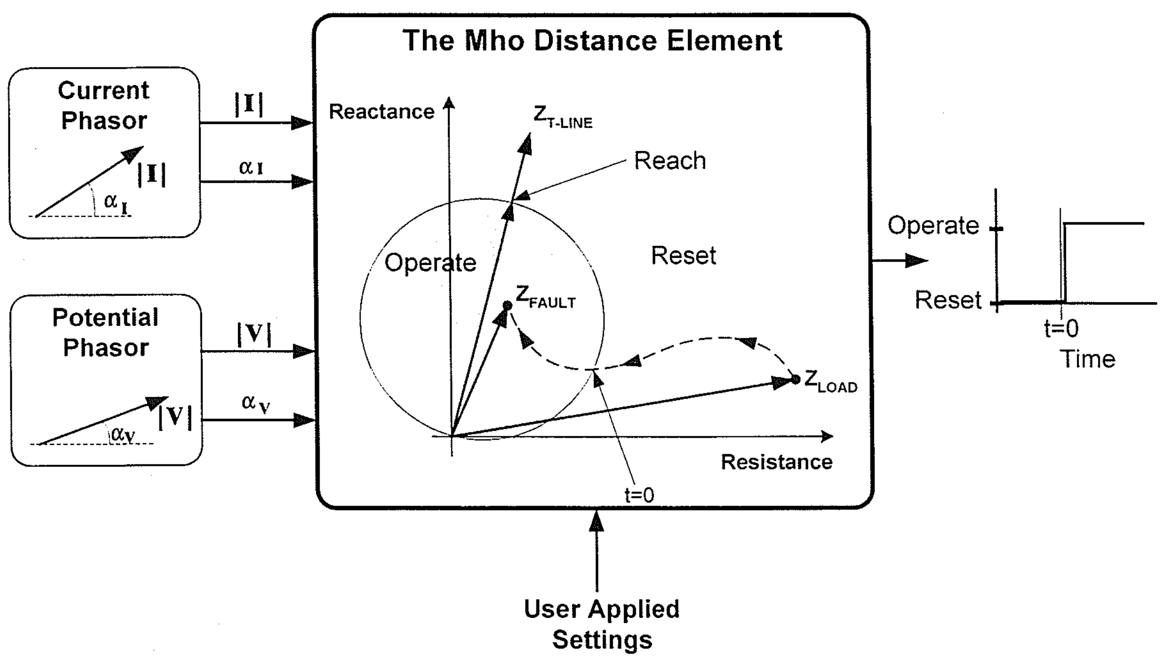


Figure 2.4: A numerical mho distance element

typical fault. When a fault occurs, the impedance calculated by the relay does not instantaneously change to the faulted impedance. The dashed line indicates a possible numerical fault impedance trajectory from the load impedance phasor to the fault impedance phasor. Real time is required for this measurement transition to take place and is the cause of inherent delay from the time of fault incidence until the element operates. Operation of the mho element occurs at t=0 when the measured impedance just enters the mho circle.

Mho elements are generally not realized through an explicit **Z=V/I** apparent impedance phasor calculation. Very commonly, numerical mho distance characteristics are implemented by comparing the phase angle between derived "operating" and "polarizing" phasors. The phase angle comparison based distance element operates if the phase angle difference between the two phasors is within a defined range. By adjusting the limits of the angle range different mho shapes can be realized including the lens and tomato shapes. This approach has been widely used and is well documented [6].

Phase mho relays and ground mho relays are used to detect phase faults and ground faults respectively. The reach setting for both phase and ground mho distance relays is set as a multiple of the protected transmission lines positive sequence impedance. A ground mho relay however requires unique compensation for the line it protects to allow it to correctly measure the positive sequence impedance during ground faults [4], [6].

2.3 The Hardware Technology of Modern Numerical Transmission Line Protection

Most modern numerical protection relays are based upon similar hardware platforms. The biggest differences between various manufacturers' relays is the firmware applied to the relay as discussed in Section 2.2.

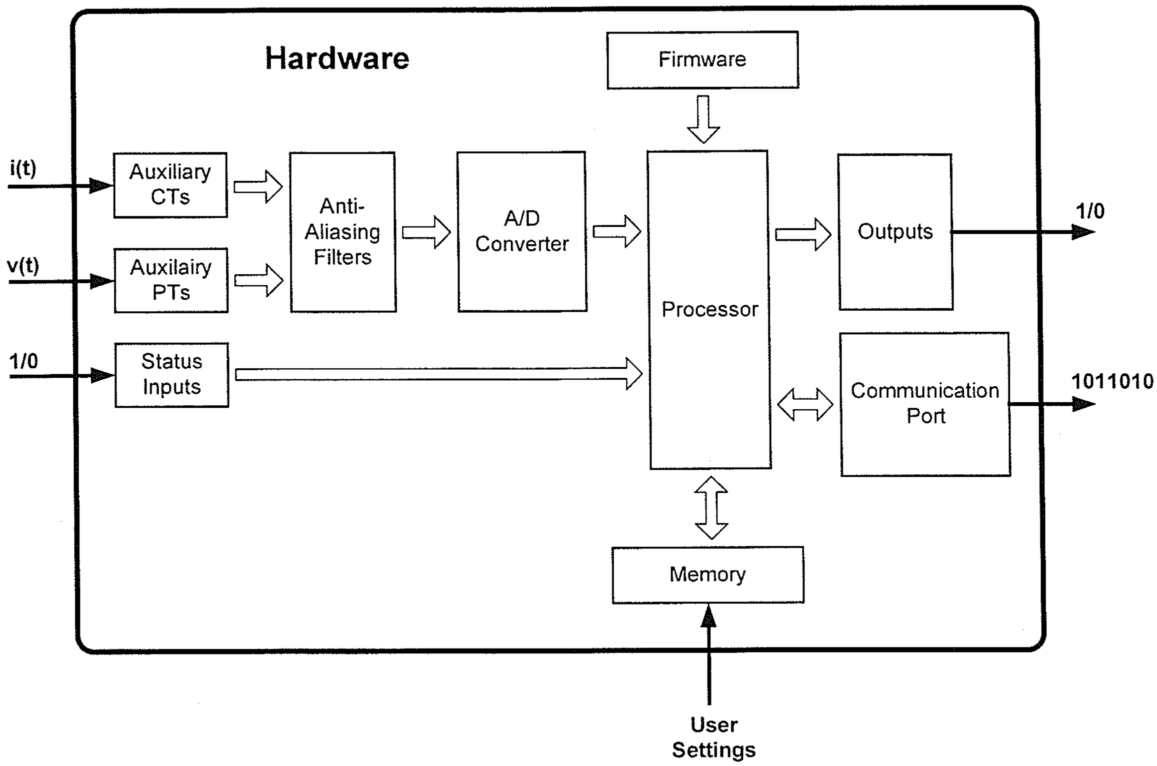


Figure 2.5: A generalized hardware block diagram for a numerical relay.

A generalized functional block diagram for a modern numerical relay is illustrated in Figure 2.5. Analogue currents i(t) and potentials v(t) from the power system CTs and PTs enter the relay and are reduced in magnitude to levels appropriate for electronic equipment by means of the auxiliary input CTs and PTs.

The current and potential signals are next passed through the anti-aliasing filters. The anti-aliasing filter is a low pass filter designed to attenuate frequency components in the input signal which exceed one-half the relay's sampling frequency (*fs*). The filter ensures accurate sampling of the input signal according to the sampling theorem, by band limiting the incoming signals to below one-half the sampling frequency. If the filter was not included, frequency aliasing may occur if the input signal contains frequency components which exceed one-half the sampling frequency. Power systems often have significant levels of harmonics which could corrupt sampled current and potential signals if left uncorrected.

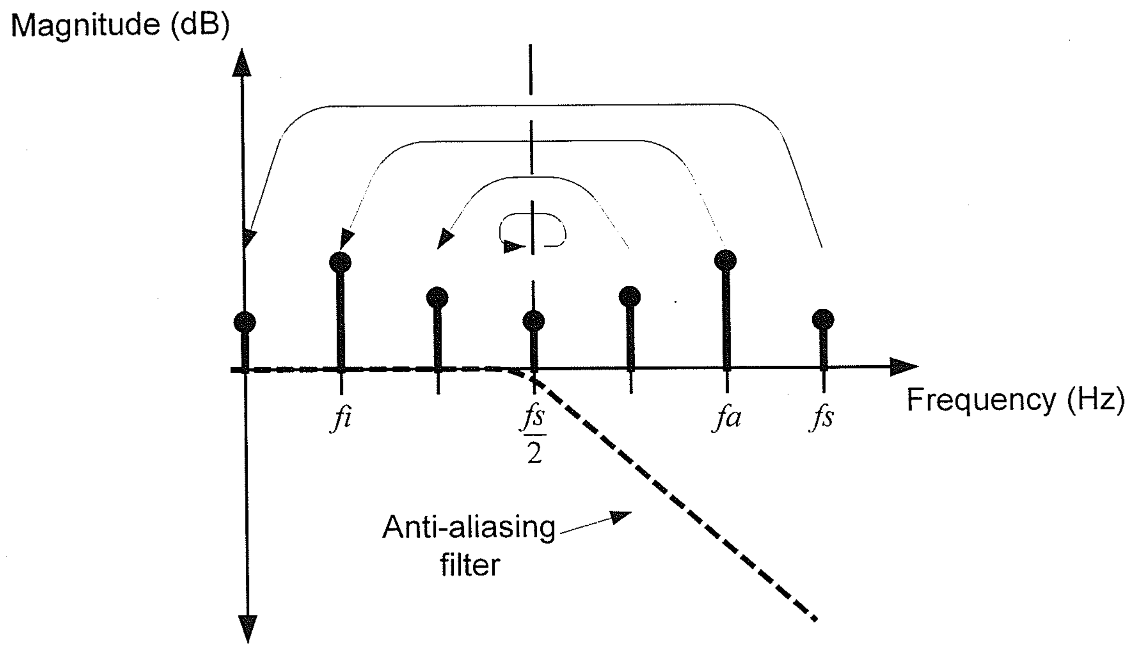


Figure 2.6: Frequency aliasing folds back frequencies exceeding *fs*/2.

Aliasing is the process whereby frequencies which exceed one-half the sampling frequency (*fs*/2) fold back upon frequencies less than *fs*/2 as shown in Figure 2.6. The higher frequency components mimic their associated low frequency components during the sampling process which result in measurement errors. Any signal containing frequency components less than *fs*/2 can be sampled without error. Signals containing frequency components exceeding *fs*/2 will cause sampling errors if the high frequency aliasing component (f_a=*fs*-*fi*) which folds back upon the frequency of interest (*fi*) being sampled is of significant magnitude. The anti-aliasing filter therefore prevents aliasing errors by attenuating all frequencies exceeding *fs*/2.

After the anti-aliasing filters, the analogue current and potential signals are sampled and converted to discrete signals in the A/D converter. These discrete signals are now ready to be analyzed in the relay processor.

Modern numerical relays use either microprocessors or DSPs to perform the required signal processing and protection algorithms. The relay firmware provides the manufacturer's unique instruction sets used by the processor to digitally process the current and potential signals for quantities of interest and to implement the protection algorithms. Numerical relays also require read/write memory to support the signal processing and protection algorithms, and recording functions.

External events can be entered into the relay's logic by connecting external contacts to drive "status inputs". When the processor determines that some action such as a trip must be initiated, it does so through outputs which generally are in the form of either a high speed electro-mechanical relay or a solid state switching device. The most modern numerical relay platforms also offer I/O capability through data communication ports using various communication protocols to connect to programmable logic controllers (PLCs) or other intelligent electronic devices (IEDs).

3. PROTECTIVE RELAY SOFTWARE MODELS

3.1 Protective Relay Software Model Types and Applications

Relay software models have existed in various forms for some time and are used by manufacturers for relay development, by utility protection engineers in application studies, and by academics for research and educational tools. Relay software models can be categorized as either phasor based models or transient models depending on the solution approach [1].

3.1.1 Phasor Based Relay Models

Phasor based relay software models consist of a set of phasor equations into which phasor quantities of current or voltage are substituted to solve for the relay response. The majority of relay models in use today are in fact phasor based models [1]. Often phasor based software models are included with commercially available phasor based fault simulation packages. These software packages allow the user to fully model their power system and simulate various types of faults. The program provides a steady state phasor solution to the applied fault. These programs can also take this phasor solution and run it through a phasor based relay model to determine if the relay operated for the applied fault.

Phasor based models can accurately represent the performance of their physical counterparts for steady state conditions. Phasor based models are very useful for the majority of users who are only concerned with setting relays in standard applications or for verifying relay coordination. With this approach the occasional slow clearing fault or false trip due to the unforeseen effect of some transient event is acceptable compared to the cost and complexity of more detailed transient studies. In fact, most protective relays are set based on steady state phasor studies only.

The major drawback to phasor based models however is their inherent inability to accurately represent how a relay responds to transient events which can not be represented in phasor domain solutions. The response of a relay to a particular power system disturbance could be affected by transients generated from the disturbance itself, the transient response of the instrument transformers, or the transient response of the relays hardware and/or firmware functions. The end result may be that a relay trips very slowly, false trips or does not trip at all.

3.1.2 Transients Affecting Relay Performance

The affects of transients on relay performance are often more of a concern with applications where fault clearing times and/or security against false operation are particularly critical. Transients alter the relay signals which may cause the protective elements in relays to function incorrectly resulting in relays falsely tripping for faults outside their protected zones (over reaching), or not tripping for faults inside their protected zones (under reaching). There are numerous circumstances that may occur where relays are subjected to transient quantities in their current and/or potential signals that may adversely affect their performance [1].

Both CTs and PTs have unique transient and steady state responses dependent upon their particular design and application. CTs and PTs may exhibit either transient or steady state ratio errors due to magnetic core saturation. CT saturation results in distorted secondary current waveforms causing numerical relays to measure less fundamental current than is actually present in the primary of the CT resulting in overcurrent or distance element under reaching. Some high voltage PTs incorporate a capacitive voltage divider in their design and are referred to as a capacitor voltage transformers or CVTs. Poor CVT transient response during faults may cause the fundamental voltage component measured by the numerical relay to appear smaller than it actually is resulting in distance element over reaching.

The presence of non-fundamental signal components should not normally present a problem to most numerical relays in the steady state since they typically extract only the fundamental component for operating purposes. Transient events such as energizing transformer banks or capacitor banks however can expose relays to transient inrush signals composed of both fundamental and non-fundamental components. It is not always clear what levels of particular frequency components exist within these types of signals making it difficult for protection engineers to appropriately apply the relay. Transient events such as these may require study to ensure relay algorithms function as intended – especially in those cases where non-fundamental signal quantities are used for restraining or operating signals.

Most system faults yield undesirable relay signal components such as "fault noise" and exponential signal offset (most often called DC offset). Transients such as these are directly targeted for removal by appropriate filters included in the relay design. The overall response and accuracy of a relay design for these conditions may be of interest in some applications.

Analyzing relay performance becomes even more complicated for evolving system faults, especially for transmission line relays employing single pole tripping and reclosing schemes. Evaluating and configuring complicated relay elements and logic under all fault scenarios is not easily done other than by injection testing methods or simulation models.

Other transmission line protection scenarios that may require transient studies include transmission lines with series capacitors, power swing studies (out of step protection), and applications electrically close to generating stations where source impedances may vary with time following a system disturbance or fault.

3.1.3 Transient Software Relay Models

Transient software relay models can be subdivided into two classes [1]. The first is the "generic" relay model that is designed based upon the general operating principles of the relay. These models for example may include software code for an element based only on a typical description such as a "positive sequence voltage polarized mho element". The generic model code implements this element using one of many generally accepted textbook techniques. Generic models of commercial relays however may be limited in their overall accuracy since the element itself and the signals it operates on may be realized differently in the physical relay, but still be suitable classified by the manufacturer's generic description. Generic relay models by virtue of their lack of detail are simple to develop but may lack accuracy with respect to operational margins and timing. Generic relay models are excellent for educational purposes but probably are not sufficiently accurate for most real world application studies [1], [3].

The second class of transient relay models is the "detailed" model. Detailed software models try to exactly incorporate the physical relay's hardware characteristics, signal processing techniques, and protection algorithm coding. Each functional block within the detailed model is designed as much as possible based upon the exact implementation within the physical relay. Detailed models offer the advantage of excellent accuracy with respect to operational margins and timing. Since detailed models attempt to produce the physical relay's internal signals exactly, detailed models offer the additional advantage of "looking inside" the relay at these signals to gain further insight into margins and timing. Numerical relays in particular lend themselves well to detailed modeling since it is possible to exactly reproduce the firmware coding of the physical relay in a modeling environment. Detailed software models have been developed by academics for relays developed in house where the details of the relay's hardware and firmware are fully known [2], [7], [8]. Detailed software models for commercially available relays however are rare and therefore not widely used today. The issues surrounding this problem are explored in the following section.

3.2 Commercial Relays and Detailed Software Modelling

Detailed software models for commercially available numerical relays are not widely available. The main reason for this is due to the reluctance of manufacturers to release any proprietary details of their products [1]. Detailed models for commercial relays can come into existence by one of two different methods. The first method has the relay manufacturer themselves develop and validate the detailed model and distribute it to their customers. Alternately, the users of the relays themselves or some third party can develop the model if the relay manufacturer is willing to release key details of the relay design.

Commercial relay manufacturers, because of their complete access to relay design details, are best suited for the job of creating accurate detailed models. For numerical relays, manufacturers could conceivably package their exact firmware coding along with the hardware model to provide a very accurate software model. In many cases, manufacturers create detailed models of their products in the development stages to test various algorithms. Regardless, few if any models ever come directly from the manufacturer.

For protection engineers and other users of the physical relays, it is a very laborious task to develop and validate a detailed software model. It would be hard to justify the time and labour expense in developing such a model versus transient testing the physical relay with injection testing methods. A great deal of information and effort is required to develop and validate an accurate model. This information can be obtained from various sources including published papers, patents, relay manuals, and the manufacturer themselves [1]. Regardless, it is hard to conceive that a manufacturer would ever hand over the design schematics and firmware coding for their product to a customer or third party for the purposes of model development. As a result, models developed by customers or third parties will always be subject to accuracy limitations.

The degree of accuracy required from the model by a user may depend entirely on their particular study requirements or personal comfort level. In some critical cases even the most accurate of software models may not satisfy the protection engineer, and therefore warrant injection testing methods. In model development, there is no way of guaranteeing a specific degree of model accuracy perhaps other than creating an exact component level hardware model which also incorporates the exact firmware scripting. Such a model should, in theory be close to exact, however would be laborious to build, and most likely would be exceedingly slow in any simulation environment. In fact research has indicated that even when the physical relay hardware and firmware are completely known, uncontrollable factors such as unsynchronized relay/model sampling can lead to inaccuracies in the model performance [2]. Regardless, it does remain possible to develop models that are sufficiently accurate for many users.

A necessary step in the development of an accurate software model is validation testing. Validation testing attempts to verify that the response of the software model is sufficiently accurate compared to the response of the physical relay for the same signal inputs. The most common model validation techniques involve injecting the physical relay with the very same simulated signals applied to the relay model [2], [7]. Academic research indicates it is desirable to validate internal signals, calculated quantities, and protection element responses in the relay model against the same from the physical relay [2], [7], [8]. Retrieving internal signals and calculated quantities from most commercial platforms may be difficult however as they are not designed for this purpose. The commercial relay model designer must therefore use whatever signals are available to establish a measure of confidence in the model. Developing an acceptably accurate model may involve an iterative process of development and validation testing.

4. DEVELOPMENT OF THE NXTPHASE LPRO SOFTWARE MODEL FOR PSCAD/EMTDC

4.1 General Description of the LPRO Relay

The NxtPhase APT LPRO model 2100 protection relay pictured in Figure 4.1 is a multifunction transmission line protection device which is based around phase and ground distance protection. The LPRO also includes numerous non-directional, directional, and negative sequence overcurrent elements, over and under voltage elements, and an out of step element. The relay includes the logic required to implement standard communication aided (pilot) tripping schemes such as Permissive Over-Reaching Transfer Trip (POTT), weak in-feed logic, breaker fail protection, dead line pickup, loss of potential (LOP), and synchronism check. The platform also provides transient recording capability to capture disturbance events with a recording clarity up to the 25th harmonic.

Figure 4.1: The NxtPhase APT LPRO Model 2100 Relay

The relay is implemented on a hardware platform consisting of 7 printed circuit board modules. The "AC Analogue Input Board" includes current and potential input transformers and anti-aliasing filter circuits. The "Main Processor Board (MPB)" utilizes a floating point DSP processor to perform all protection and recording signal processing and protection algorithm processing. The "486 CPU Board" uses a 486 class central processing unit (CPU) to provide data management utilities and communication processing. The "External Input and Comm Board" provides both contact status inputs and communication ports to connect to other external intelligent electronic devices (personal computers, remote terminal units). The "Relay Output and DC Analogue Input Board" provides high speed contact outputs for initiating external circuits. The "Front Panel Board" includes LED status indicators, an alphanumeric display and keypad to form an integrated human machine interface (HMI). The "Power Supply Board" converts connected DC power supplies to the appropriate internal board level voltages.

The LPRO relay's primary signal sampling rate is 96 samples per power system cycle (s/c) for a sampling frequency of 5760 Hz. This sampling rate is used for recording all input analogue signals. The sample rate is reduced to 8 s/c (480Hz) for all protection elements and protection logic processing. The relay protection elements all operate on numerical phasor based techniques. The mho distance elements are realized using numerical phasor based techniques to implement the classical phase angle comparison method for mho distance elements.

4.2 Model Strategy

4.2.1 Objectives

The accuracy of a detailed software model for a commercial relay depends significantly on the model designer's access to proprietary hardware and firmware details about the relay being modelled. The author in this case did receive details of specific hardware or firmware elements upon request however, the author did not receive complete access to design documents such as schematic diagrams and firmware coding. Consequently it was possible to develop a much more detailed model then would normally be possible by using only public domain information, but still not an exact model.

A desirable and reasonable objective is to have the LPRO software model perform with 100% accuracy for element operating decisions. This implies that the relay and model elements both operate or both do not operate for a given disturbance. Further, when an element does operate, the pick up and reset times of the model element must be within +/- 1 protection processing interval (+/- 2.1ms) compared to the same physical relay element for the same input signals. This level of accuracy should certainly be sufficient for the bulk of both utility and academic user study requirements. It is reasonable at the level of detailed modelling carried out to expect results where trip decision timing is significantly better.

4.2.2 Model Structure and Code Flow

The LPRO relay model is broken down into separate hardware and firmware model blocks. Each block attempts to accurately model its associated part of the physical relay hardware or firmware coding. For the most part, the hardware model blocks including the analogue signal conditioning and A/D converter are prepared based on specific circuit details released by NxtPhase. The hardware blocks are modelled with sufficient detail that the intermediate analogue signals and A/D output should match those of the physical relay. The firmware model blocks including digital signal processing, protection elements and logic are developed with the LPRO User's Manual and additional proprietary information provided by NxtPhase. This model is developed for LPRO firmware version 2.6b.

This model is developed to run within the PSCAD/EMTDC time domain simulation environment. The PSCAD/EMTDC simulation tool is composed of a graphical "CAD" based user interface (PSCAD), and a time domain simulation engine (EMTDC). The PSCAD tool provides the user with a drag and drop user interface to build power system models using standard libraries or user built models of power system components such as generators, transmission lines, control functions, and fault functions. The PSCAD interface also serves to present the time domain simulation solution to the user through various plotting features. The EMTDC simulation engine solves the case model code built in PSCAD and supplies the time domain solution back to PSCAD for the user to see. User models such as this LPRO model are included in the EMTDC case code and are solved as part of the total case solution.

The LPRO model code is written using FORTRAN 77 and is included in Appendix I. A model functional diagram is illustrated in Figure 4.2. The PSCAD/EMTDC simulation case calculates the primary system currents and potentials required at the relay location (Ia, Ib, Ic, Va, Vb, Vc). The model is developed to accept connection to either primary or instrument transformer (secondary) level system currents and voltages. Primary signals are scaled down in the LPRO model by means of an integrated external ideal ratio CT and/or PT. Users who implement actual external CT and/or PT models can elect to use secondary level inputs to the model by setting the external CT/PT ratios to 1. The relay signal quantities are then passed through auxiliary input CTs and PTs. The current and potential signals pass through the anti-aliasing filter and gain path adjustment block. Up until this point, the discrete time solution signals (Ia, Ib, Ic, Va, Vb, Vc) are at a sample rate determined by the user's selection for a PSCAD/EMTDC solution time step. The interpolating sampler and A/D converter block first samples at 96 samples/cycle and then converts those sample values to an integer value. Following this the sample decimation filter block reduces the 96 sample/cycle rate down to 8 samples/cycle and controls processing of the following blocks at that rate. All the following signal processing, protection elements, and trip and recording outputs are then processed at 8 samples/cycle. The model code then returns to 96 sample/cycle processing to update the trip and recording outputs. Outputs from the LPRO model include one dedicated trip contact output, and 9 user selectable recording outputs which will allow the user to plot in PSCAD any combination of selectable internal LPRO quantities. A detailed model code flowchart is also included in Figure 4.3.

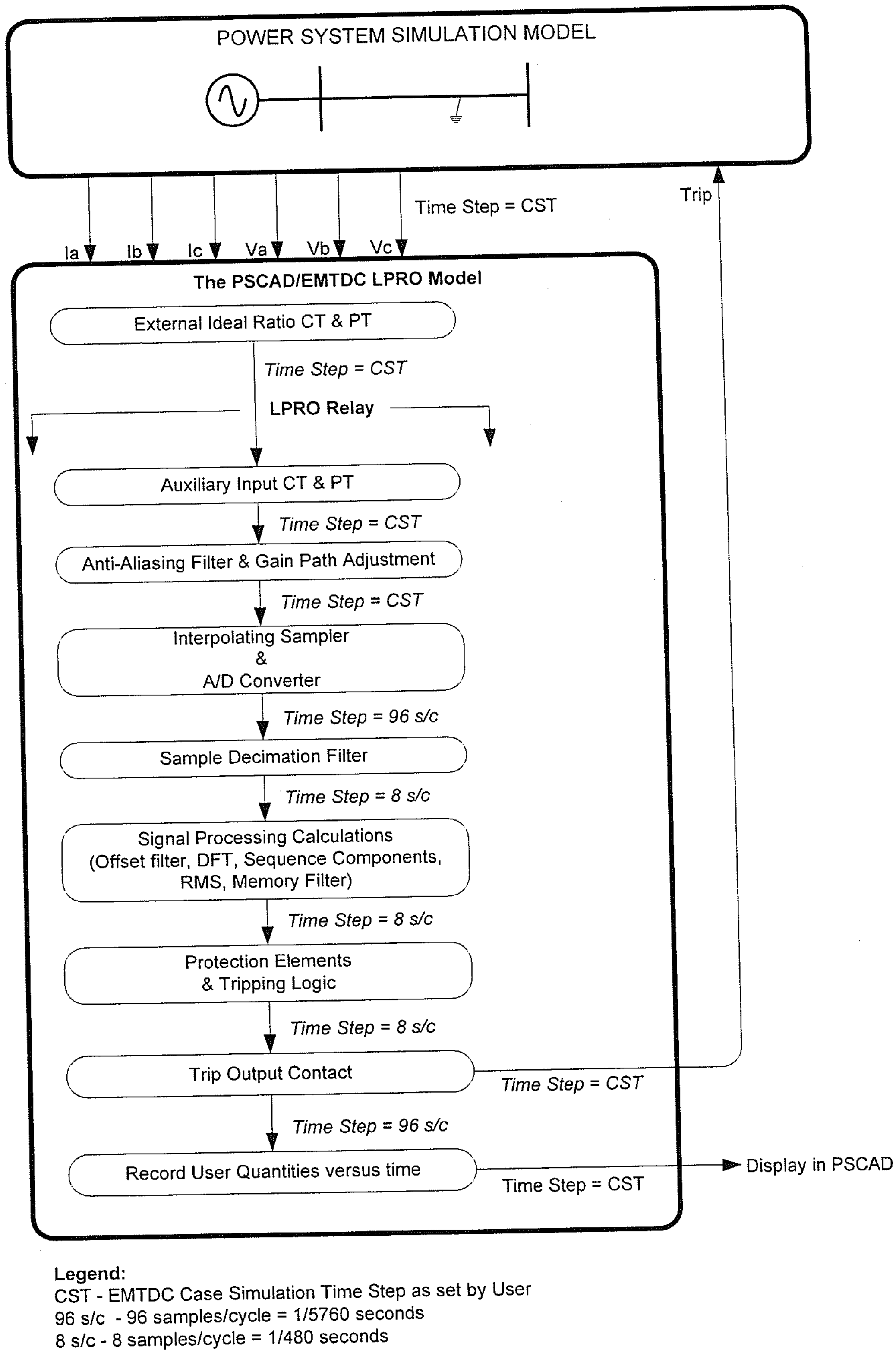


Figure 4.2: LPRO model functional block diagram.

The LPRO model includes most, but not all protection elements and logic within the relay for firmware version 2.6b. To provide a reasonable amount of work for this thesis, only those elements and functions which were seen as the most significant for transient studies were included [1]. The criteria in this case includes only those elements and functions which may operate within the first two or three cycles following the inception of a power system disturbance. Based on this criteria, the model includes all instantaneous overcurrent elements, Zone 1 and Zone 2 phase and ground distance, loss of potential logic (LOP), dead line pickup logic, and any support elements required to implement the previous such as the directional element. Elements and functions that are not included are communication aided (pilot) tripping schemes (POTT, PUTT, Blocking), Zone 3 and 4 distance elements, time overcurrent elements, voltage elements, out of step, and sync check.

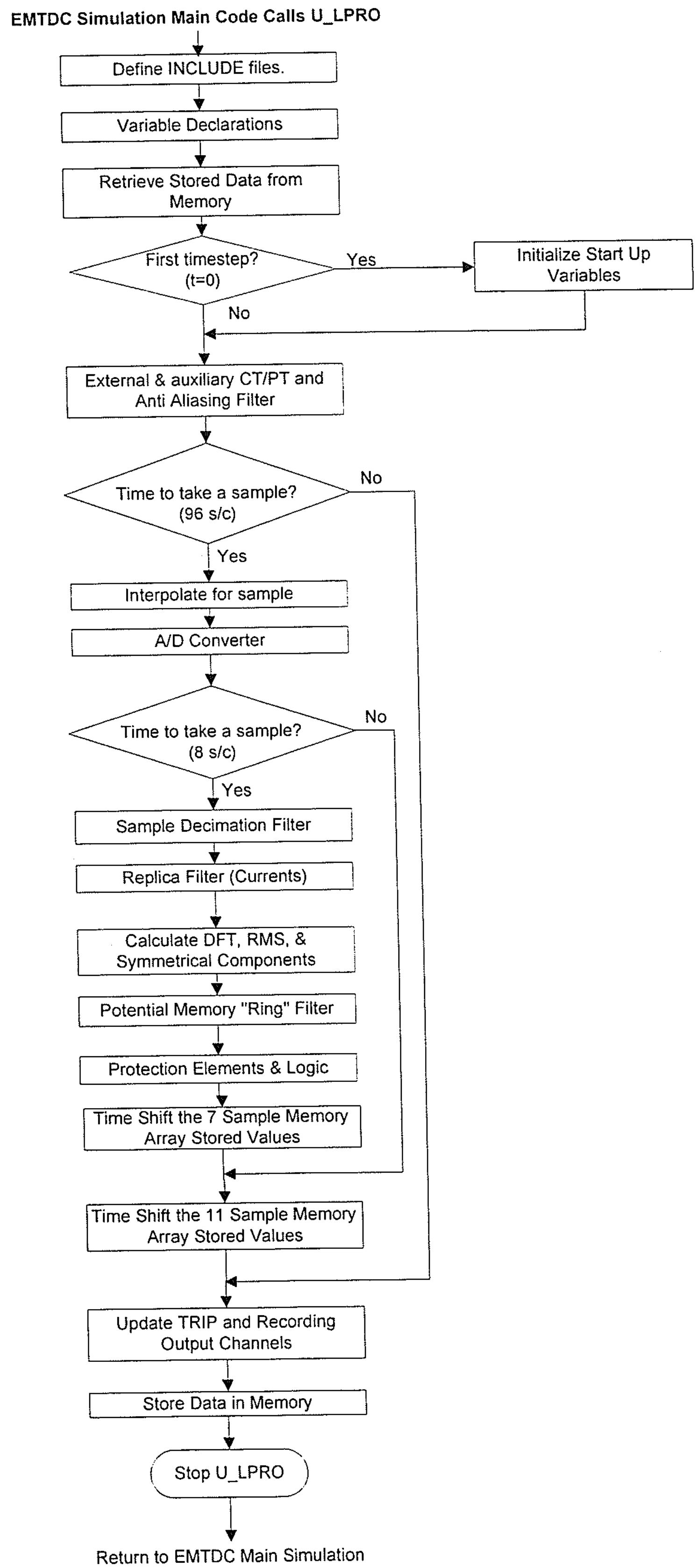


Figure 4.3: LPRO model code flowchart.

4.3 Model Block Development

4.3.1 Auxiliary Input Transformers

External current and potential signals from high voltage instrument transformers are reduced to levels appropriate for the LPRO's signal processing hardware by means of the auxiliary input transformers. The LPRO can accommodate up to 4 three phase current inputs and up to 2 three phase potential inputs. Only the "Main AC Line Currents" input and the "Main AC Volts" potential inputs are required for this model. The remaining inputs are used for recording purposes or for protective elements not included in this firmware version. The current input is officially rated conservatively by NxtPhase at 5A nominal current (In), and 100A without distortion [9]. The potential input is rated at 69V nominal voltage (Vn), and 138V continuous[9].

NxtPhase provided further comments on the physical details of this block. The auxiliary input CT secondary winding is terminated with a resistor to convert the current signal to a voltage quantity at a ratio of 1V/21.51A. NxtPhase advised that the CT and resistor burden resulted in linear operation of the current transformer for current inputs of up to 200 A RMS. The auxiliary input PT has a turns ratio of 48.8:1. NxtPhase advised that the PT operated linearly for potential inputs of up to 138 V RMS (phase to neutral).

It will be assumed that the ratio and phase error for the auxiliary input transformers is negligible when operating linearly. Based on this assumption, the input CTs and PTs are modelled with sufficient accuracy as ideal ratio transformers. In fact, the LPRO's auxiliary CTs impose a slight phase angle shift to the current signals. The current signal phase angles are later numerically corrected in firmware. Since the software model developed here implements an ideal auxiliary input CT model, the model need not incorporate the firmware phase angle correction.

4.3.2 The Anti-Aliasing Filter

Section 2.3 identified the requirement for anti-aliasing filters on numerical relays. The following circuit diagram for the analogue anti-aliasing filter was supplied by NxtPhase.

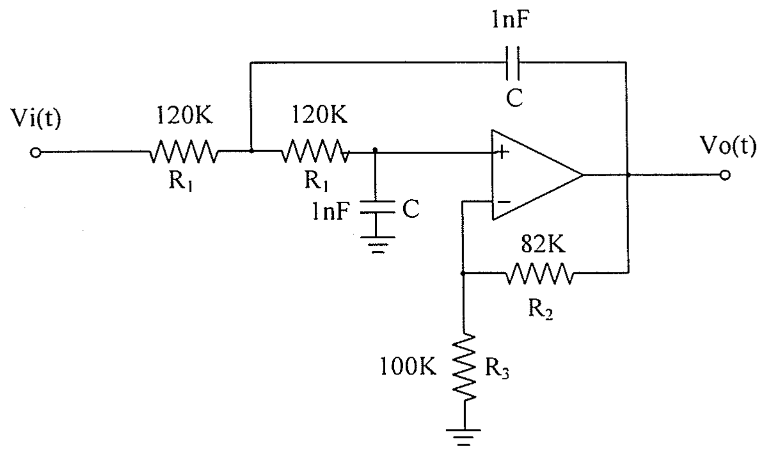


Figure 4.4: The LPRO anti-aliasing filter circuit.

The circuit in Figure 4.1 was analyzed to determine the following closed loop transfer function $A(s)$ [10].

$$A(s) = \frac{K}{s^2(CR_1) + s(3-K)(CR_1) + 1} \tag{4.1}$$

where

$$K = 1 + \frac{R_2}{R_3} = 1.82$$

The constant K represents the closed loop DC gain of the filter circuit. Comparing the transfer function to the standard form for a second order low pass filter we observe a cutoff frequency (ω_o) given by

$$\omega_O = \frac{1}{CR_1} \tag{4.2}$$

Section 2.3 indicated that anti-aliasing filters must have a cutoff frequency less than one-half the sampling frequency. The LPRO uses a sampling rate of 96 samples/cycle or 5760 Hz which requires an anti-aliasing filter with a cut off frequency of 2880 Hz or less. The component values given for the LPRO filter circuit yield a cutoff frequency of 1326 Hz.

The Bode plot in Figure 4.5 illustrates the magnitude and phase response of the anti-aliasing filter. The plots illustrate that the DC gain of the filter (K=1.82 = 5.2dB) extends out flatly to approximately 1000Hz and then drops off to 0dB just before 2000Hz. The filter roll off occurs at a rate of 40dB per octave as expected for a second order filter. The magnitude roll off after 1000Hz implies some attenuation in the magnitude of harmonics above approximately the 18th or 1080Hz. The phase angle response indicates that the phase angle delay at 60Hz is negligible.

The PSCAD/EMTDC Control Systems Modelling Functions (CSMF) library contains a standard block for implementing 2nd order transfer functions of the type shown in equation 4.1 above [11]. The standard block is of the form

$$\frac{G}{\dfrac{s^2}{\omega_o^{\ 2}} + s\dfrac{2z}{\omega_o} + 1} \qquad (4.3)$$

The parameters of the standard block will be set as follows to emulate the LPRO anti-aliasing filter

$G = 1.82 \qquad \omega_o = 2\,\pi\,1326$ rad/s $\qquad z = 0.59$

This standard filter block is used as a subroutine within the code of the LPRO model.

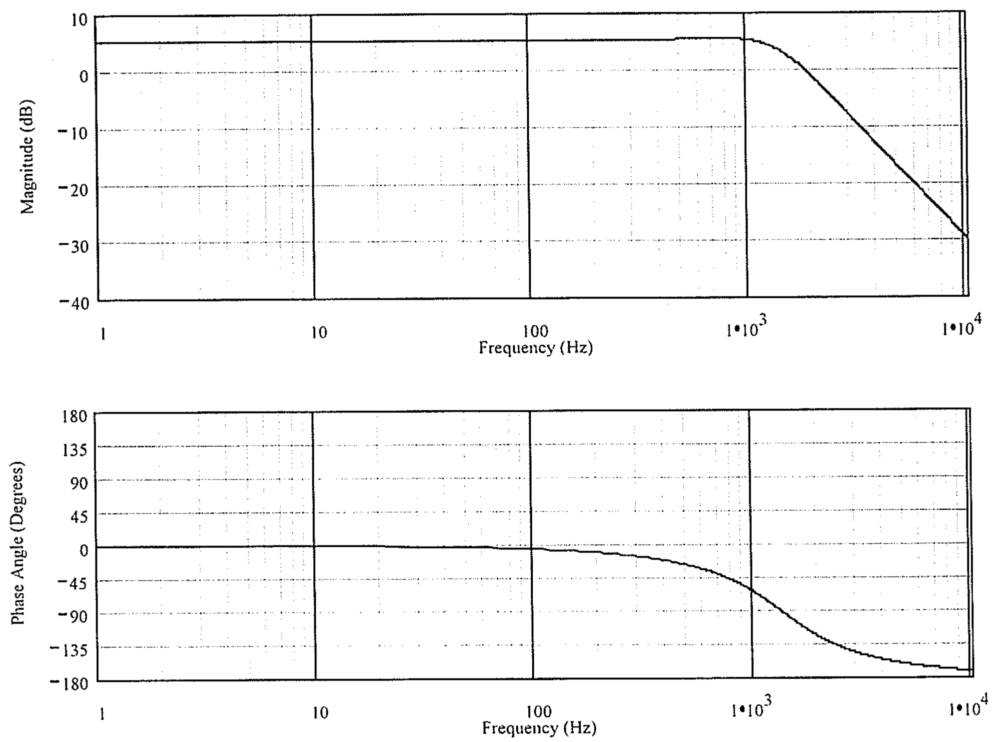


Figure 4.5: Anti-aliasing filter Bode plot

The LPRO's AC Analogue Input Board also implements other stages of signal gain adjustments both preceding and following the anti-aliasing filter. These adjustments alter the levels and reference of the signal but will be modelled as ideal gain adjustments assuming that they do not significantly affect the signal information. Consequently these stages of gain and reference adjustment are simply lumped together into a "Gain Path Adjustment" placed in the anti-aliasing filter model with net gains of 0.3144 for the PT signal and 0.1731 for the CT signal. The input PT and CT signals are therefore applied to the analogue to digital converter with an overall net gain of 1V/85.25V (1V at relay input = 11.73mV at A/D input) and 1A/68.26V (1A at relay input =14.65mV at A/D input) respectively. The reference level for the signals is shifted from 0V to 2.5V(required for the monopolar A/D converter). Finally, the PT and CT signals are limited on their reference shifted negative half cycle to 0.1 volts to address an actual limitation within the input amplifier circuits.

4.3.3 Analogue to Digital Converter

The LPRO relay converts the PT and CT analogue signals using analogue to digital (A/D) converters. Modeling of the A/D converters is achieved based on the specific characteristics of the converter [7]. In this case, NxtPhase provided both the manufacturer, part number, and design configuration for the LPRO relay A/D converters. The following converter characteristics are relevant to this particular software model.

1. The word length for the converter is12 bits plus a sign bit. The converter represents negative numbers using 2's complement. The count range is therefore –4096 to +4095.
2. The analogue voltage range that the input signal may span is +0.1 to +5.0 volts. The reference voltage level for the signal input is +2.5 volts.
3. The converter will round up on conversions that do not yield a whole number.
4. The sampling rate to the DSP processor is at a rate of 5760Hz (96 samples per power system cycle).

The converter is modelled to achieve the 4 characteristics above. The converter model includes an interpolating sampler to perform the sampling function. The sampler interpolates the EMTDC solution where relay samples are required between simulation time steps. The advantage of the interpolating sampler is that the simulation user is afforded flexibility in selecting a simulation time step rather than having to pick an exact multiple of the LPRO's sampling rate.

The A/D converter then quantizes the sampled simulation signal. The quantization process determines the digital integer representation for the sample magnitude. Samples in the range of 2.5 to 5.0 volts will yield an integer representation from 0 to 4095. Samples from 0.1 to 2.5 volts will yield an integer representation from –3931 to 0. Recall that the negative signal headroom is limited by the analogue signal amplifier which directly limits the negative converter range count. The converter model rounds up to the next largest integer for samples that do not directly yield whole numbers. Clearly the physical converter does not have infinite precision and will tolerate some amount of remainder in the conversion before the count is rounded up. It was not entirely clear from the component data sheet how this could be quantified or modelled except by intuition. To realize this, the model is crafted so as not to round up unless the conversion count remainder exceeds 0.1. For example, a converter measurement of 2876.3 would round up to 2877 however count 2876.08 would truncate to 2876. The final stage of the A/D model limits the output integer representation between the converter's count range of –4095 to 4096. Based on the analogue gain path established in 4.3.2 and the converter count range, the resulting converter resolution is 611 microvolts (applied at the converter's signal input) per converter output count which matches the exact resolution of the relay.

Included in the A/D converter model is control logic to accomplish sampling of the simulation signals at the rate of 5760Hz (96 samples/60 Hz cycle). As in the LPRO, the signal sampling is not adjusted for changing system frequency. The sampling control logic simply keeps track of what point in time the next sample is to be taken based on the fixed sample period of 1/5760 seconds.

4.3.4 Sample Decimation

The LPRO relay samples at the rate of 96 samples per cycle in order to achieve good resolution of signal transients for the purposes of oscillographic recording. This sample rate is however too fast to allow for the complex processing associated with the protection algorithms. Consequently the signal sample rate is reduced, or decimated, down to a lower rate of 8 samples per cycle or 480Hz. Simply selecting every 12^{th} sample in the sample stream would certainly reduce the rate as required however this would permit frequency aliasing since the input analogue anti-aliasing filter has a cut off frequency of 1326 Hz. As explained in

Section 2.3, preventing frequency aliasing at the new sampling frequency of 480Hz requires that the decimation filter have a cutoff frequency (-3dB) of 240 Hz or less. Failure to include this filtering function would allow 7^{th} harmonic components (420 Hz) to fold back upon the 60 Hz fundamental and corrupt the sampling process.

Sample decimation is performed by the LPRO relay in software using the DSP. The sample decimation process includes a low pass digital decimation filter function which is calculated at a rate of 96 samples/60 Hz cycle. The low pass decimation filter used in the LPRO is a nonrecursive, discrete-time finite impulse response (FIR) filter commonly referred to as a "moving average" or "smoothing" function. The filter is represented by the following discrete time equation with input samples $x[n]$ and output samples $y[n]$ both at a sample rate of 5760Hz.

$$y[n] = \frac{1}{12}\sum_{k=0}^{11} x[n-k] \qquad (4.4)$$

This decimation filter averages the present sample and the previous 11 consecutive samples to output a discrete signal at 5760 Hz representing a 12 sample moving average window.

Simulation of this filter function for a 60Hz signal indicated that the filter slightly attenuated the signal. Filter functions such as this may cause magnitude or phase angle changes to their output signals which are undesirable and numerically corrected in the relay firmware. Failure to include these magnitude and/or phase corrections will result in a model which functions incorrectly. These details may not be directly communicated by the relay manufacturer and therefore it is prudent to review the filter frequency response to ensure any required corrections are identified and included in the model.

The frequency response for the linear time invariant (LTI) constant coefficient difference equation of 4.4 can be determined easily by inspection [12]. Filter equation 4.4 is a constant coefficient difference equation with inputs $x[n]$ and outputs $y[n]$ with the general form

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k] \tag{4.5}$$

The frequency response $H(f)$ of LTI systems characterized by equation 4.5 with a fixed sampling rate of f_S is given by the following equation

$$H(f) = \frac{\sum_{k=0}^{M} b_k e^{\frac{-jk2\pi f}{f_s}}}{\sum_{k=0}^{N} a_k e^{\frac{-jk2\pi f}{f_s}}} \tag{4.6}$$

Using equations 4.5 and 4.6 the frequency response for the filter function in equation 4.4 can now be written as

$$H(f) = \frac{1}{12}\sum_{k=0}^{11} e^{\frac{-jk2\pi f}{f_s}} \tag{4.7}$$

The frequency response in equation 4.7 is plotted in Figure 4.6(Bode plot) with fs set at 5760 Hz.

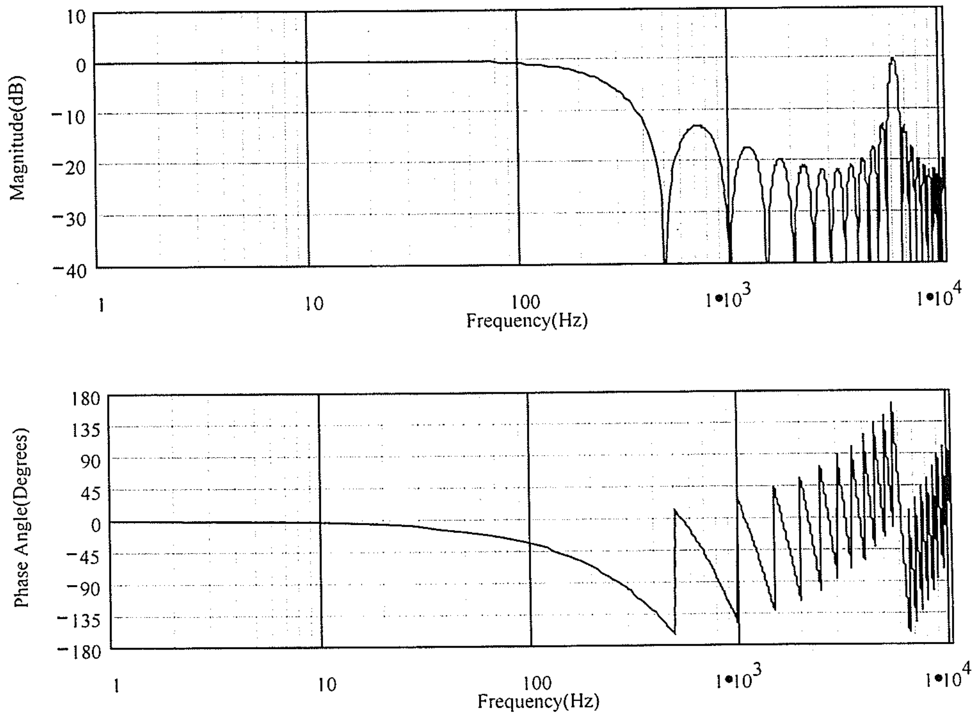


Figure 4.6: Bode plot for the decimation filter.

The Bode magnitude plot in Figure 4.6 reveals that the decimation filter function attenuates the 60Hz frequency component by approximately 2.6% and the 120Hz (2^{nd} harmonic) component by approximately 10%. These quantities are used by the protection elements and the magnitudes are numerically corrected in the LPRO to ensure accuracy. The correction will be included in section 4.3.6.

The Bode phase angle response indicates that the decimation filter delays the 60Hz and 120Hz frequency components by approximately –20 degrees and –100 degrees respectively. This is not a concern since both the 60Hz current and potential signals will be equally affected and the phase angle of the 120Hz current signal is of no significance to the protection elements.

The 96 sample per cycle current and potential signals are "decimating" down to 8 samples per cycle by calculating the filter function in equation 4.4 every 1/8 of a cycle using the 12 most recent samples.

Since the sample decimation process takes place in the DSP, the sample information passed from the A/D converter must be translated from an integer count to a floating point number representing the true secondary currents and potentials at the relay inputs. This is accomplished in the model by multiplying the A/D converter output count by the net current and potential inverse gains from the relay input to the A/D output which are 0.0417 $A_{secondary}$/count and 0.0521 $V_{secondary}$/count respectively.

4.3.5 The Replica Current Filter

For most power system faults, one or more of the faulted phase currents will include a decaying exponential component commonly referred to as "DC offset". DC offset is detrimental to the measuring accuracy of distance elements and must therefore be removed from the current signals. The LPRO relay removes the DC offset component by implementing a proprietary digital filter function selected for its ability to reject the decaying exponential component of the fault current waveform. This filter function with inputs $x[n]$ and output $y[n]$ is referred to as the Replica Filter and is reported by NxtPhase as follows

$$y[n] = \frac{c_1 + c_2}{c_3} x[n] + \frac{c_1 - c_2}{c_3} x[n-1] \tag{4.8}$$

where

$$c_1 = \frac{1}{2\cos\left(\frac{\omega T_s}{2}\right)}$$

$$c_2 = \frac{\omega \tau}{2\sin\left(\frac{\omega T_s}{2}\right)}$$

$$c_3 = \sqrt{1 + (\omega \tau)^2}$$

and where

$$T_s = \frac{1}{8\, f_{sys}} = \frac{1}{480}$$

$$\tau = \frac{X_{T-Line}}{2\,\pi\, f_{sys}\, R_{T-Line}} = \frac{L_{T-Line}}{R_{T-Line}}$$

$\omega = 2\,\pi f$ = angular velocity (radians/second)

f_{sys} = system frequency (hertz)

T_s = sampling period (seconds)

τ = transmission line time constant (seconds)

$X_{T\text{-}Line}$ = transmission line inductive reactance (ohms)

$R_{T\text{-}Line}$= transmission line resistance (ohms)

$L_{T\text{-}Line}$= transmission line inductance (henries)

The replica filter in equation 4.8 is a non-recursive, linear time-invariant, finite impulse response filter with constant coefficients c_1, c_2, and c_3 based on the time constant τ of the protected transmission line and the sampling period T_s. It is beyond the scope of this thesis to explore the theory behind the derivation of this filter's characteristic coefficients. It is relevant however to examine the filter's frequency response to understand the impact of the filter on the current signals.

Simulation of this filter function for a 60Hz signal indicated that the filter had unity gain but advanced the signal in phase by approximately +60 to +70 degrees. The following frequency response for the replica filter can be derived as explained in Section 4.3.4

$$H(f) = \frac{c_1 + c_2}{c_3} + \frac{c_1 - c_2}{c_3} e^{\frac{-j2\pi f}{fs}} \tag{4.9}$$

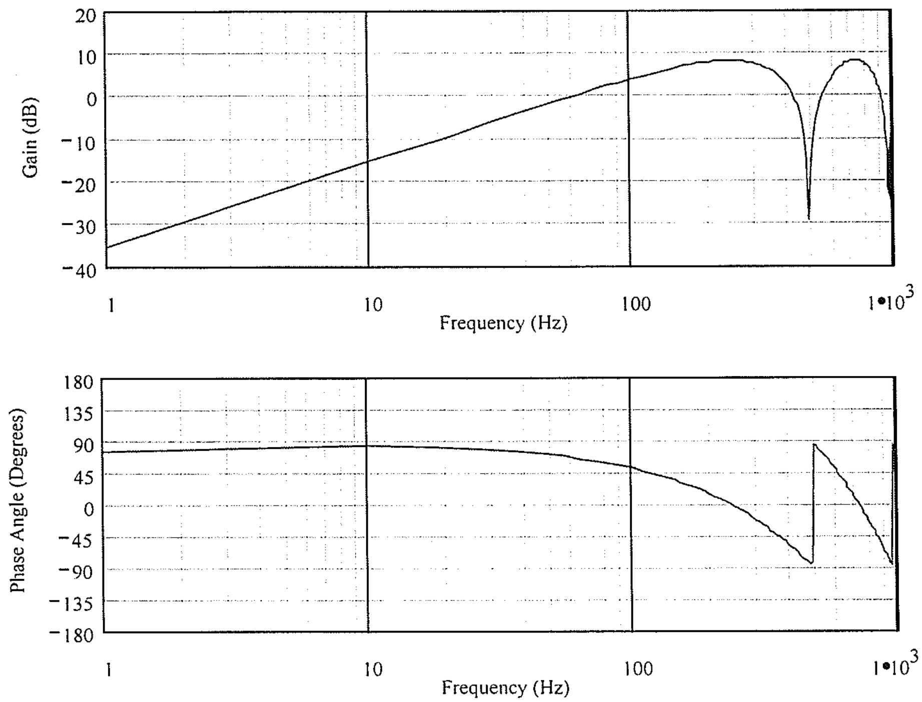


Figure 4.7: Bode plot for the Replica filter with f_{sys}=60Hz, f_s=480Hz, and τ =0.05 seconds.

Figure 4.7 presents the Bode plot for a selected system frequency (f_{sys}) of 60Hz, a transmission line time constant (τ) of 0.050 seconds, and a sampling frequency (f_s) of 480Hz.

From the plot it can be determined that at 60Hz, the replica filter has nearly unity gain (approximately 0.998), however the filter causes the phase angle of the filtered currents to be advanced approximately + 67 degrees. At 120Hz, the filter amplifies signals by a factor of approximately 1.845. The LPRO relay corrects for the phase advancement by retaining the phase angle measured from the unfiltered current signal. The 120Hz magnitude correction is discussed in the following section.

4.3.6 The Discrete Fourier Transform

The LPRO relay uses the discrete Fourier transform (DFT) to extract the fundamental (60Hz) current and voltage signals and the 2^{nd} harmonic (120Hz) current signal from the previously processed power system signals and directly convert those quantities from a discrete time signal to a discrete phasor quantity. The DFT phasor solution $X_{MAG}[n]$ and $X_{PH}[n]$ for a periodic discrete time signal $x[n]$ is determined using the following

$$X_{MAG}(n) = \frac{1}{4\sqrt{2}}\sqrt{S(n)^2 + C(n)^2} \tag{4.10}$$

$$X_{PH}(n) = \arctan\left(\frac{C(n)}{S(n)}\right)$$

where

$$S(n) = \sum_{k=n}^{n-} x[k] \sin(\omega\ K\ \Delta T)$$

$$C(n) = \sum_{k=n}^{n-} x[k] \cos(\omega\ K\ \Delta T)$$

and where

$X_{MAG}[n]$ = RMS magnitude of signal $x[n]$

$X_{PH}[n]$ = phase angle of signal $x[n]$ (radians)

$S(n)$ = DFT sine component

$C(n)$ = DFT cosine component

ω = angular velocity of the desired frequency component (radians/second)

K = sample number (1, 2, 3, ……n)

ΔT = 1/480 (seconds)

NxtPhase uses the non-recursive DFT which calculates the $S(n)$ and $C(n)$ components at every sample followed by the solution components $X_{MAG}[n]$ and $X_{PH}[n]$.

The LPRO software model uses the DFT to calculate the 60Hz current and potential phasors and the 120Hz current phasors. Section 4.3.5 identified that the LPRO's Replica Filter advances the phase angle of the 60Hz current component by 67 degrees and amplifies the 120Hz current magnitude components by 1.845. The phase advance would be detrimental to correct operation of many of the relays functions since it is applied to the current signal only. To avoid this problem, the LPRO relay and model use 2 concurrent DFT processes which extract the current phasor magnitude from the Replica Filtered signal and the current phase angle from the unfiltered current signal. Since there is no Replica Filter required for potentials, the potential phasor can be obtained directly with 1 DFT calculation. The amplification of the 120Hz current component is corrected in the model by multiplying the 120Hz current magnitude by a factor 0.542.

Section 4.3.4 identified that the decimation filter attenuates the 60Hz and 120Hz current and potential signals by 2.6% and 10% respectively. The LPRO relay model corrects for this attenuation in it's DFT calculations by multiplying the phasor magnitudes with the correction factors of 1.026 and 1.10.

4.3.7 Symmetrical Component Extraction

The LPRO relay uses current and potential symmetrical component quantities for numerous elements and functions within the relay such as the neutral overcurrent (50N), negative sequence overcurrent (50-46), directional element, dead line pickup, loss of potential, and distance element supervision [9]. The model calculates these quantities 8 times per cycle using the 60Hz non-Replica filtered current and potential phasors from the DFT process. The following formulae are used to calculate the positive, negative, and zero sequence potential component phasors $\mathbf{V}_1$, $\mathbf{V}_2$, and $\mathbf{V}_0$ respectively.

$$\mathbf{V}_1 = \frac{1}{3}(\mathbf{V}_a + \mathbf{a}\mathbf{V}_b + \mathbf{a}^2\mathbf{V}_c) \tag{4.11}$$

$$\mathbf{V}_2 = \frac{1}{3}(\mathbf{V}_a + \mathbf{a}^2\mathbf{V}_b + \mathbf{a}\mathbf{V}_c)$$

$$\mathbf{V}_0 = \frac{1}{3}(\mathbf{V}_a + \mathbf{V}_b + \mathbf{V}_c)$$

where **a** is the phasor constant $1\underline{/120^{\circ}}$. The current quantities are calculated similarly.

4.3.8 Root-Mean-Square Extraction

The LPRO relay and model uses root-mean-square (RMS) signal magnitudes for the phase overcurrent element (50P), and dead line pickup functions. The RMS magnitude provides an accurate measure of the total current or voltage versus the DFT process which provides a phasor measurement of only a specific frequency component. The RMS current and potential signal magnitudes are calculated at 8 samples/cycle directly from the sample decimation filters. The following generalized formula is used to calculate the RMS quantity X_{RMS} from an 8 sample per cycle discrete signal $x[n]$.

$$X_{RMS} = \sqrt{\frac{1}{8}\sum_{k=n}^{n-7} x^2[k]} \tag{4.12}$$

The design cutoff frequency for the decimation filter is 240 HZ. Consequently the RMS signals calculated from the decimated signals will include only the 60Hz, 120Hz, and 180Hz frequency components. Section 4.3.4 identified that the decimation filter attenuated the 60Hz and 120Hz components by 2.6% and 10% respectively. The LPRO model multiplies the calculated RMS signal by 1.026 to correct only for the dominant 60Hz component. The RMS calculation would therefore be inaccurate in the presence of significant 120Hz and 180Hz components.

4.3.9 Memory Polarising Voltage Filter (Ring Filter)

Faults electrically close to the relay location can reduce the potentials used by the relay to a point where voltage polarized distance and directional elements may not operate reliably. To mitigate this problem the LPRO relay uses a "ring filter" to provide potential signal memory action for voltage polarized elements [9]. The "ring filter" details provided by NxtPhase indicate it is a recursive, discrete time, FIR, adaptive band pass filter with output $y[n]$ for input signal $x[n]$ as follows.

$$y[n] = \left(x[n] - x[n-2] + \frac{A}{C} y[n-1] - \frac{B}{C} y[n-2] \right) C \tag{4.13}$$

where

$$A = 2\, r \cos\left(0.75\, f_{meas} \frac{\pi}{180} \right)$$

$$B = r^2$$

$$C = (1-r)\sqrt{r}$$

and where

r = design constant =0.987

f_{meas} = measured system frequency

The filter's steady state frequency response at 60Hz was analyzed in the simulation environment and found to have unity gain with no phase shift.

One filter is provided for each phase potential discrete signal $v_a[n]$, $v_b[n]$, and $v_c[n]$ (at 480 Hz) as shown in Figure 4.8. Filter design coefficients are selected such that the filter's sinusoidal output voltage has a decaying exponential envelope for a step decrease in the input sinusoid. The filter is adaptive in the sense that it uses measured system frequency (f_{meas}) to adjust the poles of the filter and retain the same output characteristics for changing frequency. The output from each phase filter is operated upon using equation 4.11 for $\mathbf{V_1}$ to obtain the positive sequence memory potential phasor $\mathbf{V_{1mem}}$. It is this potential that is used by the Zone 1 and 2 distance elements and the directional elements for a memory polarizing voltage signal.

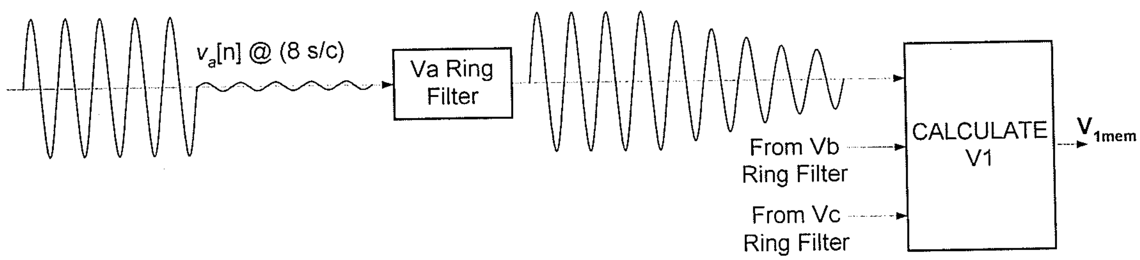


Figure 4.8: The LPRO memory polarizing potential $\mathbf{V_{1mem}}$.

4.3.10 Instantaneous Overcurrent Elements

The model includes the three LPRO instantaneous overcurrent elements for phase overcurrent (50P), neutral overcurrent (50N), and negative sequence overcurrent (50/46). Each of the three instantaneous elements can independently be made directional by enabling control from the directional element (Section 4.3.11). Elements that are directionally controlled are delayed by 5ms before being allowed to operate to ensure a correct directional decision has been made. At each protection processing interval, each element compares the level of its operating quantity to the user applied setting. If the operating quantity exceeds the applied setting, the element is asserted. Elements 50P, 50N, and 50-46 operate on RMS phase current, zero sequence current (*3Io*), and negative sequence current (I_2) respectively. The element logic is modelled as reported in the LPRO User's Manual [9]. The outputs from each element will assert the "Trip" output and can be selected by the user for recording.

4.3.11 Directional Element

The directional element used within the LPRO relay supervises the Zone 1 and 2 distance elements and can be used to directionally control overcurrent elements as well. The directional element is modelled using information from the LPRO User's Manual only [9].

The directional element uses the positive sequence memory voltage phasor $\mathbf{V}_{1mem}$ established in Section 4.3.9, and the positive sequence line current phasor $\mathbf{I}_1$ (Section 4.3.7) to calculate the positive sequence impedance phasor $\mathbf{Z}_{1mem}$ as follows.

$$\mathbf{Z}_{1mem} = \frac{\mathbf{V}_{1mem}}{\mathbf{I}_1} \tag{4.14}$$

As Figure 4.9 shows below, a forward fault is indicated if the angle of phasor $\mathbf{Z}_{1mem}$ lies within +/- 90 degrees of the protected transmission line impedance angle. The output from the directional element model block is a logic "1" indicating a forward direction and a logic "0" indicating a reverse direction or that the element is blocked because the magnitude of either of the two operating quantities $\mathbf{V}_{1mem}$ or $\mathbf{I}_1$ is too low to be reliable.

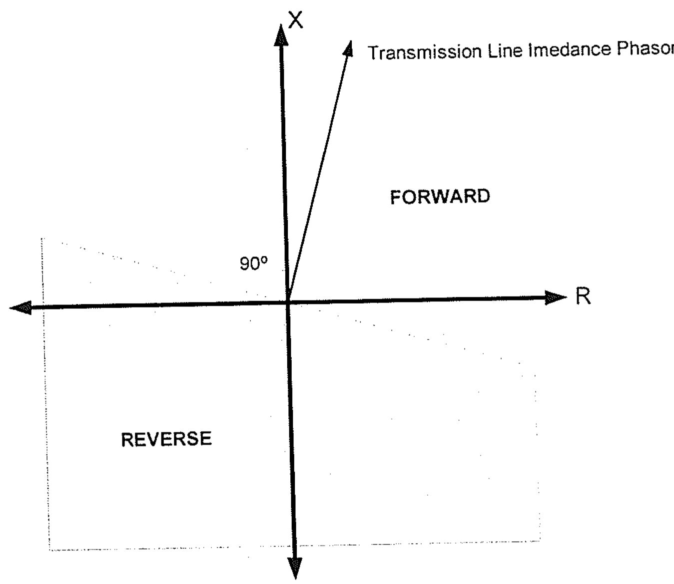


Figure 4.9: The LPRO directional element characteristic.

4.3.12 Phase Distance Elements

The LPRO relay model includes two zones of phase mho distance elements (21P1 and 21P2). The distance elements for both zone 1 and zone 2 are supervised by phase overcurrent (line-to-line current) elements and the directional element. As an option, the phase distance elements can also be supervised by the loss of potential logic (60) discussed in Section 4.3.14. The phase distance element characteristic shape can be adjusted by means of the "mho characteristic angle" setting to provide a mho circle shape (90°), a lens shape (>90°), or a tomato shape (<90°).

The phase distance element is realized in the LPRO relay by comparing the phase angle between the operating phasor (**S1**) and the polarizing phasor (**S2**) defined as follows.

$$\mathbf{S1} = \mathbf{I}_{L\text{-}L} \times \mathbf{R} - \mathbf{V}_{L\text{-}L} \qquad (4.14)$$

$$\mathbf{S2} = \mathbf{V}_{1mem}$$

where

$\mathbf{I_{L\text{-}L}}$ = line to line current phasor (amperes secondary)

R = relay set impedance reach (ohms secondary)

$\mathbf{V_{L\text{-}L}}$ = line to line potential phasor (volts secondary)

$\mathbf{V_{1mem}}$ = memory polarising potential phasor (volts secondary)

The polarising potential phasor $\mathbf{V_{1mem}}$ phase angle must be adjusted in phase angle as required for each phase distance element. The $\mathbf{V_{1mem}}$ phasor obtained in section 4.3.9 is advanced by 30° for A phase, retarded by 90° for B phase and advanced by 150° for C phase.

The mho element will operate if the angle α by which the **S1** phasor leads the **S2** phasor is within the following range determined by the mho characteristic angle (*MCA*) setting.

$$-MCA < \alpha < +MCA \qquad (4.15)$$

The LPRO relay calculates a phasor **Q** as follows.

$$\mathbf{Q} = \mathbf{S1} \times \mathbf{S2^*} \qquad (4.16)$$
$$= q_{real} + j\, q_{imag}$$
$$= |\mathbf{S1}||\mathbf{S2}| \angle \alpha$$

where

$\mathbf{S2^*}$ = complex conjugate of **S2**

q_{real} = real part of phasor **Q**

q_{imag} = imaginary part of phasor **Q**

The phase angle α of **Q** obtained by multiplying phasor **S1** with the **S2*** is the angle by which the **S1** phasor leads the **S2** phasor. The LPRO relay does not however calculate the actual angle α to perform the comparison in 4.15, but instead calculates the cosine of α. The real component q_{real} of phasor **Q** can be expressed as follows.

$$q_{real} = |\mathbf{Q}| \times \cos(\alpha) \qquad (4.17)$$

Rearranging 4.17 we obtain

$$\cos(\alpha) = \frac{q_{real}}{|\mathbf{Q}|}$$

The LPRO relay calculates the cosine of angle α by dividing the real part of phasor **Q** by the magnitude of phasor **Q**. To determine if the mho element should operate, the LPRO compares the cosine of α with the cosine of the mho characteristic angle applied by the user. Operation of the LPRO phase mho distance element occurs if 4.18 below is satisfied.

$$\cos(\alpha) > \cos(MCA) \qquad (4.18)$$

Since the cosine function is an even function, the LPRO must only make 1 comparison test versus 2 comparisons to test for a +/- angle range of 4.15.

The LPRO model incorporates this same method for the phase mho distance elements. The element logic and user setting requirements indicated in the user manual are directly replicated in the model [9].

4.3.13 Ground Distance Elements

The LPRO relay model includes two zones of ground mho distance elements (21G1 and 21G2). The distance elements for both zone 1 and zone 2 are supervised by a phase overcurrent (line current) element, zero sequence overcurrent ($3Io$) element, and the directional element. Optionally, the ground distance elements can also be supervised by the loss of potential logic (60) discussed in Section 4.3.14. The ground distance element characteristic shape can also be adjusted by varying the *MCA* user setting as described in 4.3.12.

The ground distance elements in the LPRO relay operate on the same principle as the phase distance elements described in 4.3.12. The quantities used to determine the **S1** and **S2** phasors however are different as indicated below.

$$\mathbf{S1} = \mathbf{I_G} \times \mathbf{R} - \mathbf{V_{L\text{-}G}} \qquad (4.19)$$
$$\mathbf{S2} = \mathbf{V_{1mem}}$$

where $\mathbf{I_G}$ is the compensated phase current determined as

$$\mathbf{I_G} = \mathbf{I_P} + \mathbf{Ko} \times \mathbf{3Io}$$

and where

$$\mathbf{K_O} = \frac{\mathbf{Z_O} - \mathbf{Z_1}}{3\mathbf{Z_1}}$$

$\mathbf{R}$ = relay set impedance reach (ohms secondary)
$\mathbf{V_{L\text{-}G}}$ = line to ground potential phasor (volts secondary)
$\mathbf{V_{1mem}}$ = memory polarizing potential (volts secondary)
$\mathbf{I_P}$ = measured phase current (amperes secondary)
$\mathbf{K_O}$ = zero sequence current compensation factor
$\mathbf{3Io}$ = zero sequence current (amperes secondary)
$\mathbf{Z_O}$ = transmission line zero sequence impedance (ohms)
$\mathbf{Z_1}$ = transmission line positive sequence impedance (ohms)

The polarising potential phasor $\mathbf{V_{1mem}}$ phase angle must be adjusted in phase angle as required for each ground distance element. The $\mathbf{V_{1mem}}$ phasor obtained in section 4.3.9 is correct as is for A phase, retarded by 120° for B phase and advanced by 120° for C phase.

The ground distance element logic and user setting requirements indicated in the user manual are directly replicated in the model.

4.3.14 Loss of Potential Logic

The LPRO relay uses various potential and current operated elements together with logic decisions to determine if a loss of potential has occurred due to a fuse operation. The loss of potential logic (60) can be enabled to supervise the LPRO's distance elements to prevent false tripping caused by a faulty potential signal which is not associated with a power system fault. This logic and all of its associated user setting requirements are replicated in the model as indicated by the LPRO users manual [9].

4.3.15 Dead Line Pickup

The LPRO relay uses various potential and current operated elements together with logic decisions to determine if a transmission line is faulted immediately after closing the line circuit breaker. The dead line pickup function is a "stand alone" element which can be enabled or disabled as desired. This logic and all of its associated user setting requirements are replicated in the model as indicated by the LPRO users manual [9].

4.3.16 Trip Contact Output

The LPRO model uses one dedicated output channel to represent the relay's trip output contact. As per NxtPhase, this output is modelled to include a pickup delay of 3ms to represent the pickup delay of the electromechanical output relay following its assertion by the digital trip signal in the LPRO. The contact also includes a dropout delay of 100ms as designed into the LPRO relay output contacts [9].

4.4 The LPRO Model in PSCAD

The LPRO model block developed for PSCAD is shown below in Figure 4.10.

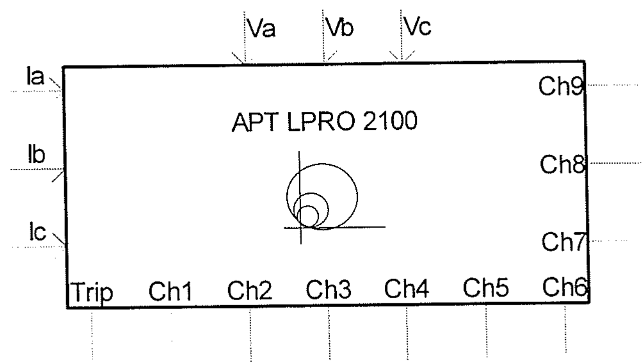


Figure 4.10: The PSCAD LPRO model block.

The model relay signal inputs Ia, Ib, Ic, Va, Vb, and Vc can be connected to either instrument transformer models or to primary level PSCAD signals. The "Trip" output models the electromechanical relay output and is used to initiate actions such as tripping a circuit breaker. Output channels 1-9 are configured by the user to record any number of internal LPRO signal quantities. These quantities can be plotted by connecting the LPRO model channels to "Output Channel" signal recorders in PSCAD.

Settings are applied to the model through the "Edit parameters" selection. The LPRO model settings are broken up into the following 8 menus.

1.) External Ideal CT and/or PT

2.) Set Transmission Line Data.

3.) Set Zone 1 Distance Elements.

4.) Set Zone 2 Distance Elements.

5.) Set Instantaneous Overcurrent Elements.

6.) Set Loss of Potential

7.) Set Dead Line Pickup

8.) Set Recorder Channels.

With the exception of the external ideal CT and/or PT and recorder channels, all settings are the same for the model as for the actual relay. Internal LPRO signal quantities are selected for recording by entering the appropriate signal assignment number listed in Appendix II into the desired channel number in the LPRO model "Set Recorder Channels" menu.

An example of a setting window for " 3.) Set Zone 1 Distance Elements" is shown in Figure 4.11.

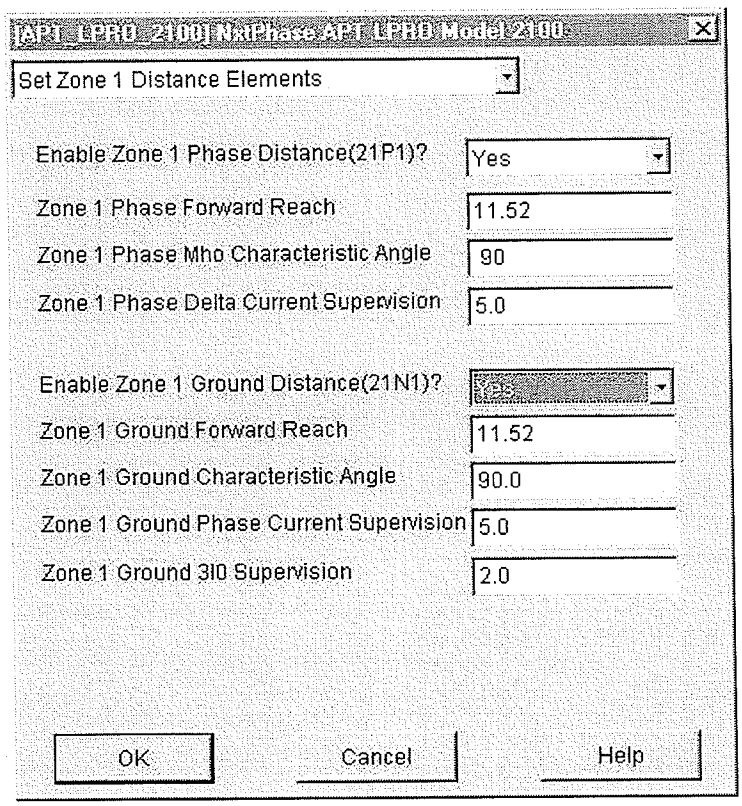


Figure 4.11: LPRO model setting menu for Zone 1 distance elements.

5. LPRO MODEL VALIDATION

5.1 Model Validation Strategy

The model validation strategy must allow for a reasonable assessment to be made of the overall accuracy of the LPRO model in relation to the desired performance objectives stated in Section 4.2.1. The preferred method of validating relay software models is to apply the exact same current and potential signals to both the relay model and physical relay and verify that the response of the model and relay are within the desired design objectives [1].

The LPRO model developed here is easily validated by this method using PSCAD/EMTDC and its real time playback or "RTP" recorder function as illustrated in Figure 5.1 [11]. A model power system is built in PSCAD to generate the required fault current $i[n]$ and potential $v[n]$ discrete signals. These current and potential signals are applied to the LPRO model directly within the simulation case itself. The response of any of the LPRO model's protection elements can than be observed for operating decisions and measured for timing using the PSCAD graphing features.

At the same time the simulation is ran, the current and potential signals applied to the relay model are also applied to the RTP recorder. The RTP recorder is configured to save the current and potential signals to an external file in IEEE COMTRADE format [13]. The COMTRADE signal files can then be played back to the physical relay using a relay test set which supports COMTRADE file playback. The response of the physical relay to the same signals is analysed using the NxtPhase viewing software "RecordBase View™" and the results compared to those of the model.

The number of signals and functions within the relay and model are numerous and it is not within the scope of this investigation to validate each signal and function explicitly. Rather, validation will be based on the correct operation and timing of 5 protection elements within the relay. The five elements include phase overcurrent (50P), neutral overcurrent (50N), negative sequence overcurrent (50-46), zone 1 phase distance (21P), and zone 1 ground distance (21G). These 5 elements use various LPRO signals and functions and collectively will give a reasonable assurance as to whether or not the model developed here is accurate.

The validation testing examines the response of these 5 elements for phase-to-phase faults and single-line-to-ground (SLG) faults placed at various points on a transmission line. Additionally, the response of the same 5 elements is examined for the non-sinusoidal current conditions associated with various degrees of CT saturation.

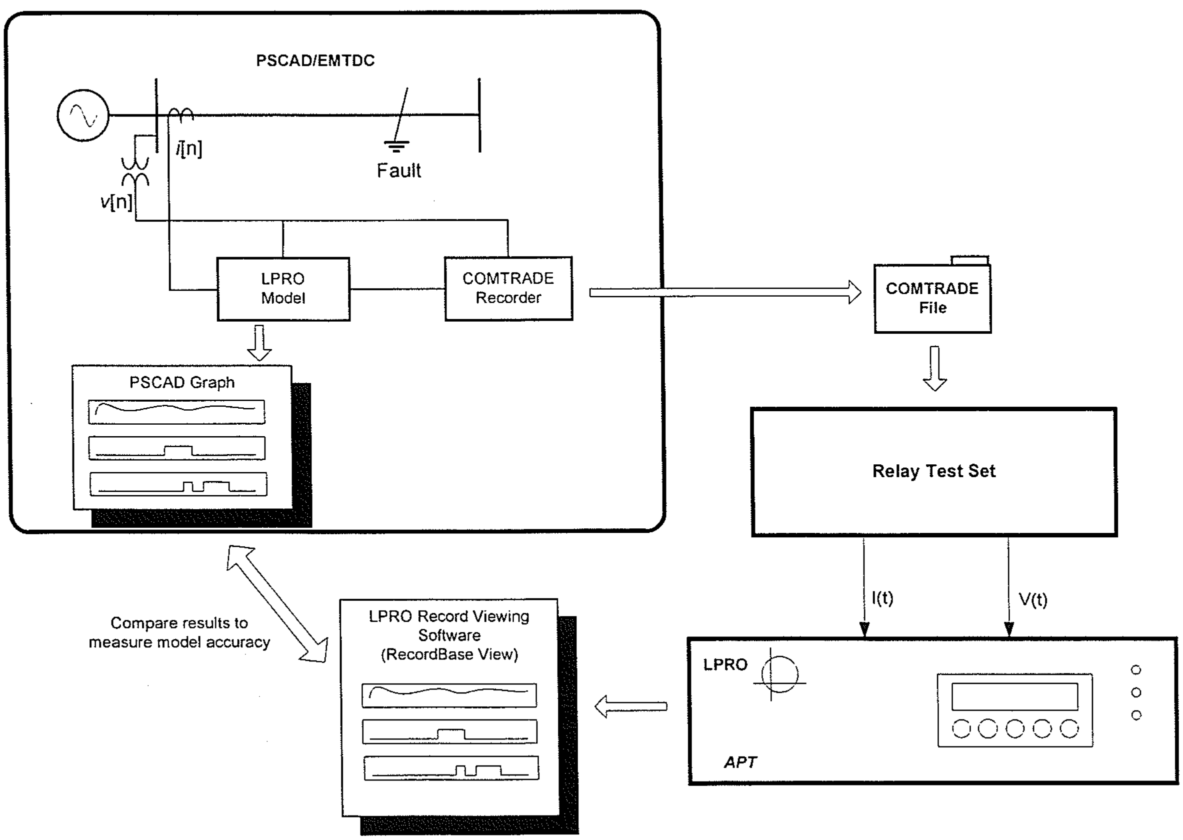


Figure 5.1: The LPRO model validation strategy.

5.2 Model Validation Testing Procedure

The PSCAD/EMTDC model validation simulation case was developed to test the LPRO model using the strategy outlined in Section 5.1. A simple power system model is composed of a single 230kV generator connected to a transmission line and terminating with a fixed 30 megawatt load resistance. A simple CT model is implemented using the PSCAD model for a single phase transformer. Varying degrees of CT saturation can be achieved by increasing the connected burden resistance. No PT model is used, rather the primary voltage signal is adjusted in the LPRO model using the "External ideal PT" function. The PSCAD validation case is shown in Figure 5.2.

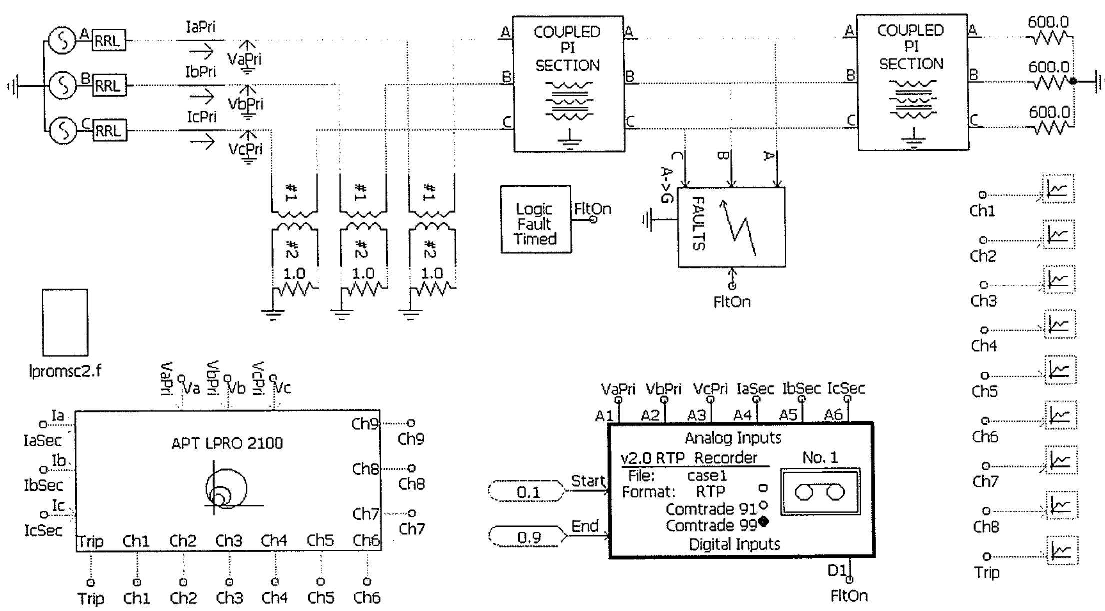


Figure 5.2: The PSCAD validation case.

The transmission line impedance $Z_{T\text{-}LINE}$ used is typical of that for a 230kV transmission line. The generator impedance in this case represents the system impedance *Zsys* and was set equal to the impedance of the transmission line $Z_{T\text{-}LINE}$ for a source impedance ratio (*SIR*) as follows.

$$SIR = \frac{Z_{sys}}{Z_{T-LINE}} = 1.0 \qquad (5.1)$$

Both Z_{sys} and $Z_{T\text{-}LINE}$ were scaled suitably to reduce the CT secondary fault current to a worst case peak of 12.5 amperes RMS to avoid overdriving the current amplifiers in the relay test set.

The transmission line model is split in two parts to allow faults to be placed at various distances along the line by adjusting the impedance of each part accordingly. The PSCAD "Faults" block is used to apply phase-to-phase or SLG faults as required. The RTP recorder generates the required COMTRADE files for playback into the physical relay.

Settings were applied to the LPRO model to enable the 5 protection elements identified in Section 5.1. The zone 1 phase and ground distance elements were set for a reach of 90% of the total impedance of the transmission line. The three overcurrent elements 50P, 50N, and 50-46 were set at 3A, 3A, and 1A respectively. The LPRO model recorder was set to record the outputs from the same 5 elements. These LPRO model elements are viewed and measured for timing using the PSCAD graphical output tools.

A total of 25 simulation cases were run. For cases 1-10, SLG faults were applied on the transmission line at distances from 10% up to 100% of the zone 1 distance element reach setting. The faults were applied at 10% incremental steps. Cases 11-20 applied phase-to-phase faults in a similar manner. Finally cases 21-25 were run for a SLG fault located at 10% of the zone 1 reach for increasing degrees of CT saturation.

All 25 case COMTRADE files were run into the physical LPRO relay (firmware version 2.6b) and the response of the same 5 protection elements monitored. The LPRO elements were viewed and measured for timing using the NxtPhase software package RecordBase View. Figure 5.3 includes both the PSCAD and RecordBase View graphs generated for test case 1 (SLG fault at 10% of Zone 1 reach).

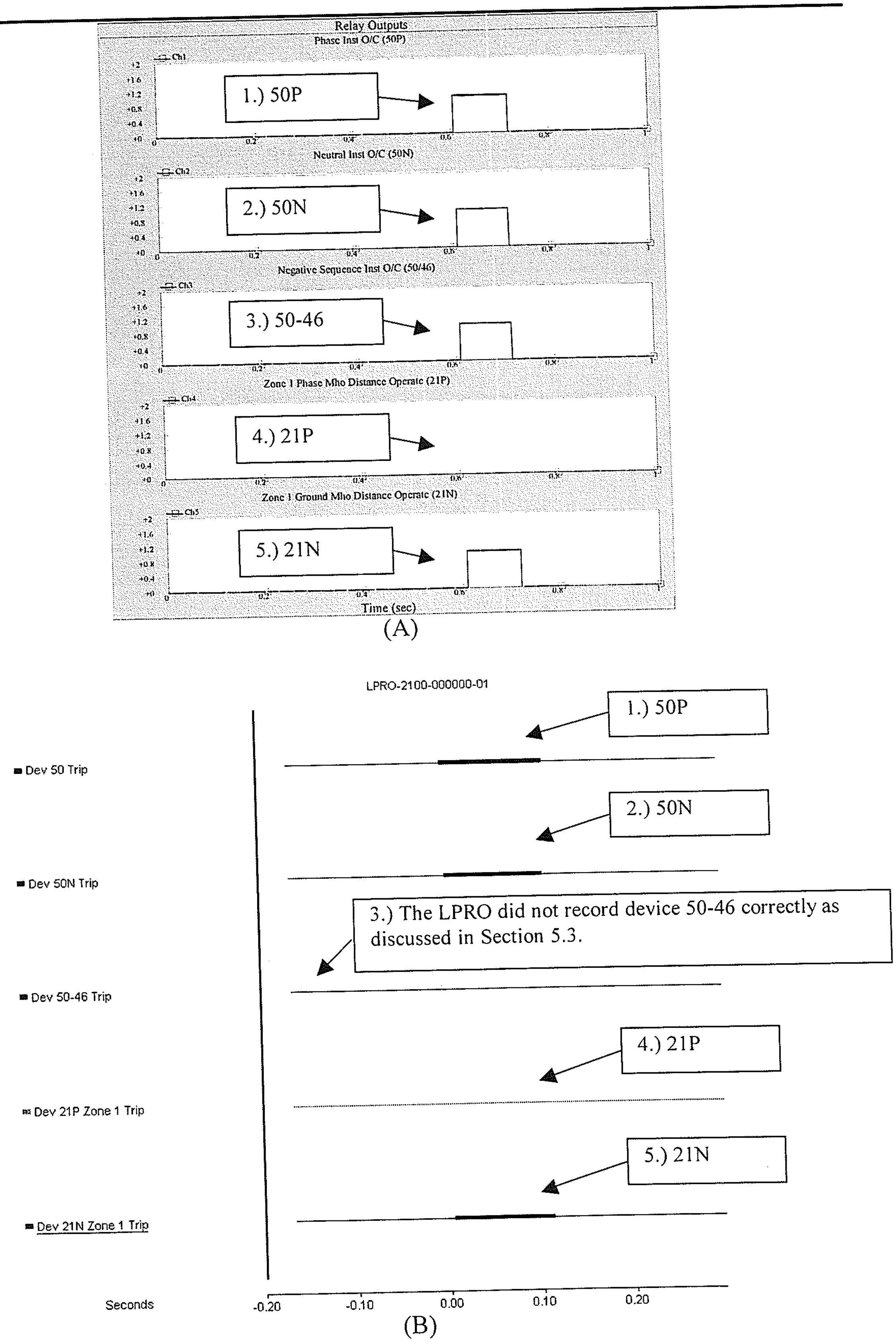


Figure 5.3: (A) PSCAD model and (B) RecordBase View relay results for Case 1.

The PSCAD and RecordBase View graphs in Figure 5.3 can be used to accurately measure timing by zooming in on the rising and falling edges of the signals and using the cursor based measurement tools. All pick up and reset timing measurements are made using the time of fault inception as time zero. Timing measurements made from the PSCAD graphs and RecordBase View graphs were made with an uncertainty of +/- 0.1 milliseconds. Section 4.2.1 stated the desired performance objective for the model is 100% accuracy for element operation decisions and within +/- 1 protection processing interval (+/- 2.1 ms) for element pickup and reset timing when compared to the physical relay for the same input signals. Incorporating the cursor based measurement uncertainty error from both PSCAD (+/- 0.1 ms) and RecordBase View (+/- 0.1ms) tools results in a total permissible error of +/- 2.3 ms. This implies that the model element pick up and reset times must be within +/- 2.3 ms of the relay element times to meet the performance objective.

5.3 Test Results and Discussion

A total of 25 test cases were run and the 5 protection elements monitored resulting in a total of 125 instances where we can observe element operating decisions. Operating decision accuracy is simply determined by verifying that the LPRO model elements and physical relay elements either both operated or both did not operate for each test case. The validation test results indicated correct trip decisions for 124 instances out of 125 for an element operating decision accuracy of 99%.

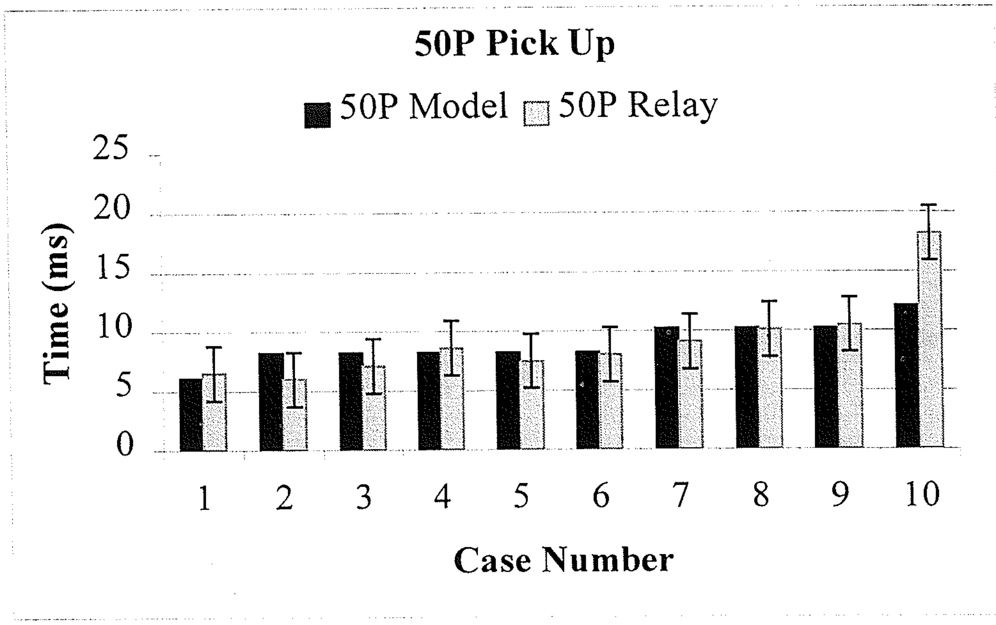
An apparent error with the relay recording or RecordBase View software resulted in no recording of the negative sequence overcurrent element 50-46 for any of the 25 test cases even though the relay event recorder indicated the element had operated. The internal relay event recorder however always captured the pick up time of this element relative to the other elements operating for each test. All other tested elements were successfully recorded for each case, and pick up/reset timing measured for each element. Using relative element timing from the event recorder, the 50-46 pickup time from fault inception was estimated by adding or subtracting the relative time difference to a measured element. Reset timing for this element however cannot be determined by this method and is therefore not included in the test results.

Cases 1 to 10 applied single line to ground faults at increasing distances along the transmission line from the relay position. Case 1 applied the fault at 10% of the zone 1 distance element reach and case 10 applied the fault at 100% of the zone 1 distance element reach. The pickup and reset times for all monitored protection elements were compared to determine if the model response matched that of the relay within the permissible error of +/- 2.3 ms. Figure 5.4 illustrates this comparison for the phase overcurrent element (50P) pickup and reset timing. Error bars included for each test indicate +/- 2.3 ms relative to the 50P relay element. All measured times are elapsed time from the point of fault inception.

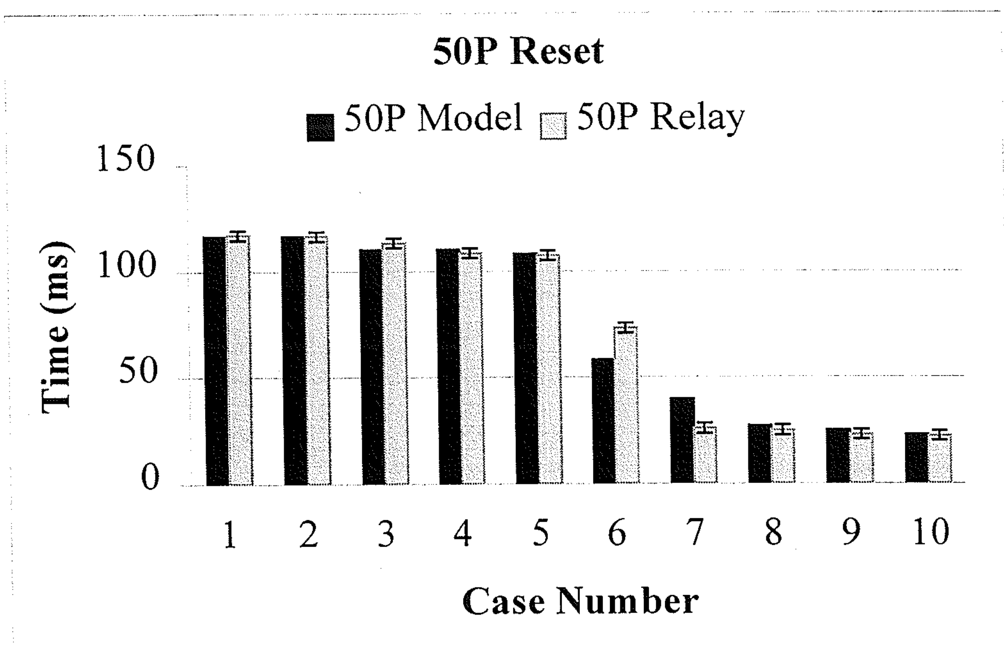
Figure 5.4(a) shows the model pickup timing is accurate within the permissible error with the exception of case 10 where the model tripping time occurred approximately 6.1 ms before the relay trip. Figure 5.4(b) reveals however that for cases 3, 6, and 7, the model 50P reset timing error exceeds the desired limit. In Figure 5.4 we observe acceptable model pickup timing for 9 out of 10 cases for a pick up timing accuracy of 90%. Similarly, acceptable reset timing accuracy for cases 1-10 is 70%.

This method is used to compare the operation of all 5 monitored elements for the 10 SLG fault cases 1-10. Detailed timing results for all elements are included in Appendix III. For SLG cases 1-10, the phase distance element (21P) did not trip in the model or the relay in any of the 10 cases as expected. In total, cases 1-10 had 4 trips out of 40 where the model and relay element pick up times differed by more than 2.3 ms, and 7 resets out of 30 (50-46 reset timing unavailable) where the element reset timing exceeded 2.3 ms. This results in a pick up timing accuracy of 90% and a reset timing accuracy of 77%.

The single instance of an incorrect trip decision occurred in case 10 where the 50N model element did not trip however the 50N relay element did trip. The relatively



(a)



(b)

Figure 5.4: Phase overcurrent 50P (a) pickup timing and (b) reset timing comparison for SLG fault cases 1-10. Error bars indicate +/- 2.3 ms relative to relay pickup times. All times are relative to fault inception.

long pickup time measured for the 50N relay element in this case (20.3 ms) suggests that the relay 50N operating current (3Io) just marginally exceeded the relay setting threshold where it did not exceed the setting threshold in the model.

Cases 11-20 applied phase-to-phase faults at increasing distances along the transmission line from the relay position. Detailed timing results for these cases are also included in Appendix III. For cases 11-20, the ground distance element (21G) and neutral overcurrent (50N) did not trip for the model or the relay in any of the 10 cases as expected. In total, cases 11-20 had 0 trips out of 30 where the model and relay element pickup times differed by more than 2.3 ms, and 6 resets out of 20 where the element reset timing exceeded 2.3 ms. This results in acceptable pickup timing accuracy of 100% and reset timing accuracy of 70%.

Cases 21-25 applied a single line to ground fault at 10% of the zone 1 distance element reach and CT saturation was introduced by increasing the burden resistance connected to the model CTs. Saturation severity increased from the least severe in case 21 to the most severe in case 25. Figure 5.5 illustrates the degree of CT saturation in the A phase CT secondary current for case 23 compared to the same fault in case 1 with no CT saturation.

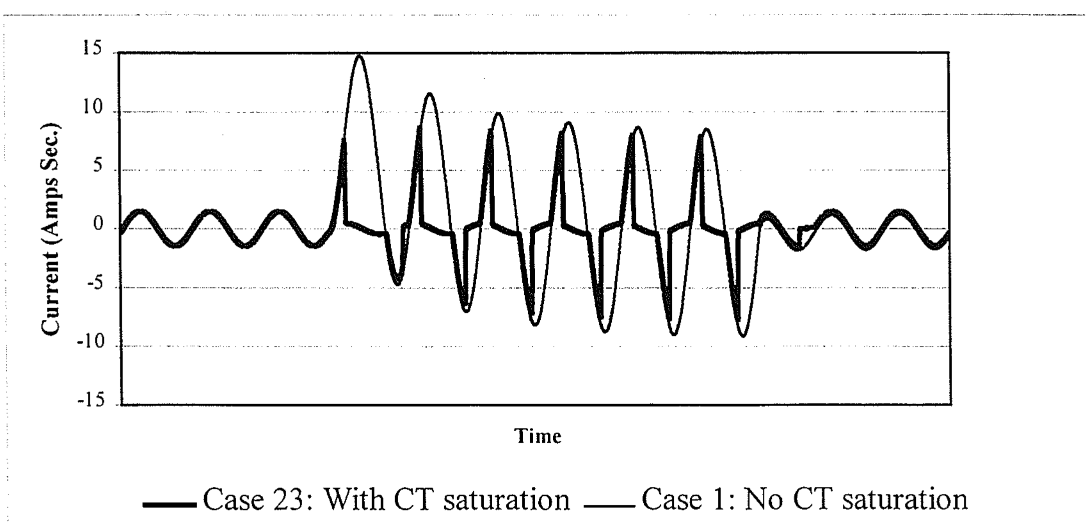


Figure 5.5: CT saturation in Case 23 compared to the same fault with no CT saturation in Case 1.

Detailed timing results for cases 21-25 are included in Appendix III. The phase distance element (21P) did not trip in the model or the relay in any of the 5 cases as expected. The severe CT saturation resulted in slower element operating times for cases 21 to 23 and absolutely no element operations for cases 24 and 25. In total, cases 21-25 had 1 element operation out of 9 where the model and relay element pick up times differed by more than 2.3 ms, and 0 resets out of 7 where the timing difference exceeded 2.3 ms. This results in acceptable pickup timing accuracy of 89% and reset timing accuracy of 100%.

The results for all the elements for cases 1-25 are summarized below in Table 5.1.

Table 5.1: Summary of pickup and reset timing accuracy for cases 1-25.

	Number of Operations per Element											
	Cases 1-10 Element				**Cases 10-20** Element			**Cases 21-25** Element				**Total (%)**
Pickup Timing Accuracy	50P	50N	46	21G	50P	46	21P	50P	50N	46	21G	
$\| t_{model} - t_{relay} \| \leq 2.3$ ms	9	8	9	10	10	10	10	2	2	2	2	94 %
$\| t_{model} - t_{relay} \| > 2.3$ ms	1	2	1	0	0	0	0	0	0	0	1	6 %
Reset Timing Accuracy												
$\| t_{model} - t_{relay} \| \leq 2.3$ ms	7	8	*	8	6	*	8	2	2	*	3	77 %
$\| t_{model} - t_{relay} \| > 2.3$ ms	3	2	*	2	4	*	2	0	0	*	0	23 %

* Data unavailable due to recording error with relay.

The results in Table 5.1 indicate the model timing accuracy on pickup appears to be quite good with 94% of all model elements picking up within +/- 2.3 ms of the relay. Model timing accuracy on drop out however is much less pleasing with only 77% of all model elements resetting within +/- 2.3 ms of the relay. The errors on pickup occurred almost exclusively for cases 1-10 (SLG faults) however the errors on reset are divided nearly equally between cases 1-10 and cases 10-20 (phase-to-phase faults). All 5 protection elements on the model contributed nearly equally to the total number of errors from the model. The errors occurring for test cases 1-25 appear random in nature and not directly attributable to any one element or fault type.

5.4 Error Analysis

The model element operating decision accuracy and pickup/reset timing accuracy test results presented in section 5.3 do not meet the desired accuracy objectives outlined in section 4.2.1. While the results appear promising, they are probably not sufficiently accurate to gain the confidence of the majority of users who may use such a model. This section analyzes the errors occurring in test cases 1-25.

It is entirely reasonable to expect that the errors encountered may be attributed to modelling errors. Before this conclusion can be drawn however we must first exclude other sources of error in the testing procedure. One probable source of error with the injection testing method used is caused by unfaithful reproduction of the analogue relay current and potential signals by the relay test set. Signal playback inaccuracies may result in pickup errors and timing errors since the relay is being injected with a different signal than the model.

The relay fault records for test cases 1-25 were reviewed to examine the injected current and potential signals recorded by the relay. Ideally, these recorded signals should nearly exactly match the signals in their associated COMTRADE source files. The potential signals appeared to be reproduced almost exactly, however significant errors were observed in the recorded current signals. The A phase current from the case 10 COMTRADE file and the case 10 relay fault record are plotted together along with the difference current in Figure 5.5. Clearly the current recorded by the relay appears significantly different from the COMTRADE file current. The recorded relay current appears to have a larger DC offset component. The current error during the fault ranges from 5-10% which may certainly account for the discrepancies in operation and timing observed between the model and relay.

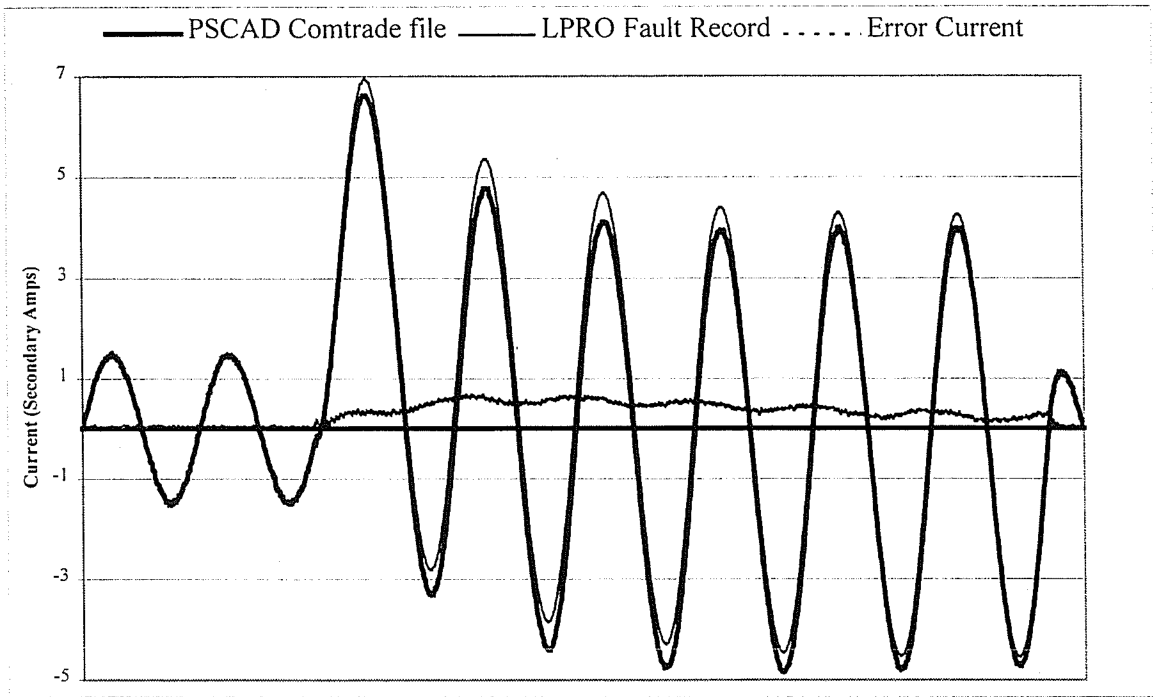


Figure 5.6: Case 10 (SLG fault) A phase current error.

Further simulation testing was conducted to determine how much the current error impacted the testing results for cases 1-25. The current and potential signals recorded by the relay were exported into a text delimited output file. Using the "File Read" function within PSCAD, each fault record file was played back to the LPRO model and the resulting model pickup and reset operations were measured and compared to the relay injection testing results The timing results of the fault record playback testing are included in Appendix IV, and are summarized below in Table 5.2.

The results in Table 5.2 show that the model responds excellently for cases 1-25 with an overall element operation decision accuracy of 100%. All but one pickup operation and all reset operations for the model occurred within +/- 2.3 ms of the relay. Overall accuracy for pickup timing was 99% and for reset timing was 100%. The single timing error occurred in case 21 when the ground distance element (21G) picked up 6.4 ms before the relay element.

Table 5.2: Summary of pickup and reset timing accuracy for cases 1-25 using fault record playback.

	Number of Operations per Element											
	Cases 1-10 Element				**Cases 10-20** Element			**Cases 21-25** Element				**Total (%)**
Pickup Timing Accuracy	50P	50N	46	21G	50P	46	21P	50P	50N	46	21G	
$\lvert t_{model} - t_{relay} \rvert \leq 2.3$ ms	10	10	10	10	10	10	10	2	2	2	2	99 %
$\lvert t_{model} - t_{relay} \rvert > 2.3$ ms	0	0	0	0	0	0	0	0	0	0	1	1 %
Reset Timing Accuracy												
$\lvert t_{model} - t_{relay} \rvert \leq 2.3$ ms	10	10	*	10	10	*	10	2	2	*	3	100 %
$\lvert t_{model} - t_{relay} \rvert > 2.3$ ms	0	0	*	0	0	*	0	0	0	*	0	0%

* Data unavailable due to recording error with relay.

The significant difference in results between section 5.3 and the results obtained with the fault record playback method indicate that the observed current error significantly impacted the results of the validation testing. The source of the current error can not be determined without further investigation and may be attributable to either poor signal reproduction by the relay test set, by the relay's pre-recording processing, or a combination of the two.

If the observed current error is due only to the relay test set, then playing back the fault record signals into the model is a preferable method to validate the model against the relay since both are using nearly the same signals. The test results using the fault record playback method would therefore best represent the accuracy of the model developed. The test results obtained using the fault record playback method indicate that all 5 elements tested in the model appear to be sufficiently accurate for most transient study requirements. The single timing error which occurred for case 21 gives rise to some speculation about a possible modelling error. It is not clear why the pickup timing error occurred in this case.

If the observed current error is caused largely by the relay's pre-recording processing, we must conclude that the model developed here does not accurately incorporate this effect. By playing back the fault record file into the model we have inherently captured all pre-recording signal distortions from the physical relay and run them back through the model. Doing this resulted in almost perfect model response. If the current error is indeed relay induced, and the model was accurate, then the results of the initial validation testing should be as accurate as those obtained using the fault record playback method. Therefore if the current error is entirely caused by the relay, then the results of the validation testing in Section 5.3 would represent the accuracy of the model. A possible source of model error could be from the ideal turns ratio model used for the auxiliary input current transformers. A more accurate electromagnetic model may be necessary.

It must also be emphasized that the testing done here did not attempt to validate the entire LPRO model, but rather only the 5 elements included in the testing. Further validation testing is required for other elements and functions to fully validate the model.

6. CONCLUSIONS

Determining the response of numerical relays to power system disturbances is a difficult task due to the complex architecture of the devices. The use of advanced signal processing techniques and complex protection logic can confuse the most capable protection engineer trying to determine how the relay will respond to a particular system disturbance. Traditional injection testing methods for evaluating relay response to disturbances is indeed effective however the complication and expense associated with the process makes it undesirable except for the most critical applications. Detailed transient software models for numerical relays provide a convenient and inexpensive alternative to the protection engineer for evaluating relay response to system disturbances.

The software model developed for the LPRO relay is a detailed model which closely follows the hardware and firmware architecture of the physical relay. The assistance of the relay manufacturer was essential to developing a detailed and accurate model. The relay user's manual yielded most of the required modelling information on the relay specifications, settings, and protection logic used within the relay. Further proprietary information was obtained from the manufacturer regarding analogue circuit details, signal processing techniques (including digital filters), and protection element implementation. The relay firmware coding however was not made available therefore the model coding could not be made exact. Generally speaking, the amount of effort required to research and develop the model code was significant as reflected by the length of model code.

Particular care must be given when modelling filter functions to ensure they are accurately represented and their effects are fully understood. Filters may cause magnitude or phase angle changes to their output signals which are undesirable and numerically corrected in the relay firmware. Failure to include these magnitude and/or phase corrections will result in a model which functions incorrectly. These details may not be directly communicated by the relay manufacturer and therefore it is prudent to review the filter frequency response to ensure any required corrections are identified and included in the model.

Model validation test results indicated correct element operating decisions for 124 instances out of 125 for a decision accuracy of 99%. The model timing accuracy on pickup appeared to be quite good with 94% of all model elements picking up within +/- 2.3 ms of the relay. Model timing accuracy on reset however was less accurate with only 77% of all model elements resetting within +/- 2.3 ms of the relay.

An analysis of the validation test results revealed that the current signals recorded by the relay had large errors when compared to the COMTRADE source files. These errors could account for the element operating decision and timing errors observed. In order to determine the impact of these errors on the validation test results, the current and potential fault recordings from the relay were exported to an external file and played back to the relay model in PSCAD/EMTDC. For fault record playback simulation, the model responded excellently for cases 1-25 with an overall element operating decision accuracy of 100%. All but one pickup operation and all reset operations for the model occurred within +/- 2.3 ms of the relay. Overall accuracy for pick up timing was 99% and for reset timing was 100%. The single pickup error cannot be readily explained and may point to an inaccuracy in the model.

The accuracy difference between the initial validation test results and the fault record playback results indicate that the current error significantly affected the test results. The observed current error could be due to poor playback quality by the relay test set, errors in the model, or a combination of both. Further work is required to determine the source(s) of the current error.

If the observed current error can be attributed largely to the relay test set, then the results obtained using the fault record playback method indicate that all 5 elements tested in the model appear to be sufficiently accurate for most transient study requirements. If the current errors are due to the relay's pre-recording processing, then the test results obtained through the initial validation tests indicate the model does not come close enough to the performance objective. The testing done here however does not validate the entire LPRO model, but rather only the 5 elements included in the testing. Further validation testing is required for other elements and functions to fully validate the LPRO model developed here.

REFERENCES

[1] P.G. McLaren, K. Mustaphi, G. Benmouyal, S. Chano, A. Girgis, C. Henville, M. Kezunovic, L. Kojovic, R. Martilla, M. Meisinger, G. Michel, M.S. Sachdev, V. Skendzic, T.S. Sidhu, D. Tziouvaras "Software Models for relays", IEEE Transactions on Power Delivery, Vol. 16, No. 2, April 2001, p 238-245.

[2] P.G. McLaren, E.N. Dirks, R.P. Jayasinghe, I. Fernando, G.W. Swift, Z. Zhang "An Accurate Software Model for Off-line Assessment of a Digital Relay", Proceedings of the 1996 Canadian Conference on Electrical and Computer Engineering (CCECE'96), Part 2, May 1996, p 737-740.

[3] Chul-Hwan Kim, Myung-Hee Lee, Raj K. Aggarwal, Allan T. Johns, "Educational Use of EMTP MODELS for the Study of a Distance Relaying Algorithm for Protecting Transmission Lines", IEEE Transactions on Power Systems, Vol. 15, No. 1, February 2000, p 9.

[4] S.H. Horowitz and A.G. Phadke, "Power System Relaying", Research Studies Press Ltd., 1992.

[5] A.G. Phadke and J.S. Thorpe, "Computer relaying for Power Systems", Research Studies Press Ltd., 1988.

[6] J.L. Blackburn and W.A. Elmore, "Protective Relaying Theory and Applications", Chapters 2 & 6, Marcel Dekker Inc., 1988.

[7] T.S. Sidhu, M.S. Sachdev, M. Hfuda, "Computer Simulation of Protective Relay Designs for Evaluating Their Performance", Proceedings of the 1996 Canadian Conference on Electrical and Computer Engineering (CCECE'96), Part 2, May 1996, p 720-723.

[8] T.S. Sidhu, M.A. Hfuda, M.S. Sachdev. "Generating Relay Models for Protection Studies", IEEE Computer Application in Power, Vol. 11, No. 4, Oct. 1998, p 33-38.

[9] APT Power Technologies: *APT LPRO Model 2100 V2.6 Users Manual*, Copyright 2000, APT Power Technologies, Winnipeg, Manitoba, Canada.

[10] A.S. Sedra and K.C. Smith, "Microelectronic Circuits", Holt, Rinehart and Winston, Inc., 1987.

[11] Manitoba HVDC Research Centre: *PSCAD/EMTDC Users Manual*, Copyright 1998, Manitoba HVDC Research Centre, Winnipeg, Manitoba, Canada.

[12] A.V. Oppenheim, A.S. Willsky, I.T. Young, “Signals and Systems”, Prentice-Hall, Inc., 1983.

[13] “IEEE Standard Common Format for Transient Data Exchange (COMTRADE) for Power Systems”, IEEE Standard C37.111-1999.

Appendix I – LPRO Model Fortran 77 Code

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!  PSCAD/EMTDC USER SUBROUTINE
!  APT LPRO 2100 MODEL
!
!  TODD E BUCHHOLZER, 2002 08 17
!  UNIVERSITY OF MANITOBA
!  MASTER OF SCIENCE THESIS
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
      SUBROUTINE U_LPRO(IA,IB,IC,VA,VB,VC,PTRM,CTRM,
     +LZ1MAG,LZ1ANG,LZ0MAG,LZ0ANG,LINEKV,P211EN,P211FR,P211CA,
     +P211DC,P212EN,P212FR,P212CA,P212PD,P212DC,N211EN,N211FR,
     +N211CA,N211PC,N211I0,N212EN,N212FR,N212CA,N212PD,
     +N212PC,N212I0,POEN,PO50PU,PODC,NOEN,NO50PU,NODC,NSEN,
     +NS50PU,NSDC,LOPEN,LOPI1B,LOPI0B,DLPEN,DLP27P,DLP50P,DLP50N,
     +DLPI2E,DLPI2R,REC1,REC2,REC3,REC4,REC5,REC6,REC7,REC8,REC9,
     +TRIP,CHNL1,CHNL2,CHNL3,CHNL4,CHNL5,CHNL6,CHNL7,CHNL8,CHNL9)

!***********************************************
! VARIABLE DESCRIPTIONS
!***********************************************
!
! MODEL INPUTS
!-------------
! IA,IB,IC,VA,VB,VC - SIGNAL SAMPLES FROM EMTDC
!
! RELAY MODEL SETTINGS
!---------------------
! INCTPT - INCLUDE EXTERNAL CT/PT IN LPRO MODEL?
! CTRM, PTRM - SET EXTERNAL CT AND PT RATIOS
! LZ1MAG - SET LINE POS SEQ MAGNITUDE
! LZ1ANG - SET LINE POS SEQ ANGLE
! LZ0MAG - SET LINE ZERO SEQ MAGNITUDE
! LZ0ANG - SET LINE ZERO SEQ ANGLE
! LINEKV - SET LINE KV
! P211EN - SET ENABLE ZONE 1 PHASE?
! P211FR - SET ZONE 1 PHASE FORWARD REACH
! P211CA - SET ZONE 1 PHASE CHAR. ANGLE
! P211DC - SET ZONE 1 DELTA CURRENT SUPERVISION
! P212EN - SET ENABLE ZONE 2 PHASE?
! P212FR - SET ZONE 2 PHASE FORWARD REACH
! P212CA - SET ZONE 2 PHASE CHAR. ANGLE
! P212PD - SET ZONE 2 PHASE PICKUP DELAY
! P212DC - SET ZONE 2 DELTA CURRENT SUPERVISION
! N211EN - SET ENABLE ZONE 1 GROUND?
! N211FR - SET ZONE 1 GROUND FORWARD REACH
! N211CA - SET ZONE 1 GROUND CHAR. ANGLE
! N211PC - SET ZONE 1 PHASE CURRENT SUPERVISION
! N211I0 - SET ZONE 1 3I0 CURRENT SUPERVISION
! N212EN - SET ENABLE ZONE 2 GROUND?
```

```
! N212FR - SET ZONE 2 GROUND FORWARD REACH
! N212CA - SET ZONE 2 GROUND CHAR. ANGLE
! N212PD - SET ZONE 2 GROUND PICKUP DELAY
! N212PC - SET ZONE 2 PHASE CURRENT SUPERVISION
! N212I0 - SET ZONE 2 3I0 CURRENT SUPERVISION
! POEN - SET PHASE OVERCURRENT ENABLE?
! PO50PU - SET 50 PICKUP
! PODC - SET PHASE OC DIRECTIONAL CONTROL?
! NOEN - SET NEUTRAL OVERCURRENT ENABLE?
! NO50PU - SET 50N PICKUP
! NODC - SET NEUTRAL OC DIRECTIONAL ENABLE?
! NSEN - SET NEGATIVE SEQUENCE OC ENABLE?
! NS50PU - SET 46/50 PICKUP
! NSDC - SET NEG SEQ OC DIRECTIONAL CONTROL?
! LOPEN - SET LOSS OF POTENTIAL ENABLE?
! LOPI1B - SET I1 BLOCKING
! LOPI0B - SET 3IO BLOCKING
! DLPEN - SET DEAD LINE PICKUP ENABLE?
! DLP27P - SET 27 VPOS PICKUP
! DLP50P - SET 50 PICKUP
! DLP50N - SET 50N PICKUP
! DLPI2E - ENABLE 2ND HARMONIC RESTRAINT
! DLPI2R - I2/I1 RATIO
!
! INTERNAL MODEL VARIABLES
!-------------------------
! A- COMPLEX OPERATOR 1/_120
! W - ANGULAR FREQUENCY (RADIANS)
! SMPPER - SAMPLING CLOCK PERIOD CONSTANT (1/5760)
! NXTSMP - TIME OF THE NEXT SAMPLE
! DIFTIM - (TIME)-(TIME OF NEXT SAMPLE)
! IAOLD,IBOLD,ICOLD - LAST TIMESTEP IA,IB,IC
! VAOLD,VBOLD,VCOLD - LAST TIMESTEP VA,VB,VC
! ADRES - A/D RESOLUTION PER COUNT
! IGAIN - CURRENT PATH INVERSE GAIN FROM CURRENT INPUT TO DSP
PROCESSING
! VGAIN - POTENTIAL PATH INVERSE GAIN FROM CURRENT INPUT TO DSP
PROCESSING
! SDC - SAMPLE DECIMATION COUNTER
! IASMP(X),IBSMP(X),ICSMP(X) - LAST TWELVE RELAY CURRENT SAMPLES @ 96
S/C
! VASMP(X),VBSMP(X),VCSMP(X) - LAST TWELVE RELAY POTENTIAL SAMPLES @ 96
S/C
! IA8SMP(X),IB8SMP(X),IC8SMP(X) - DEC FILTERED CURRENT SAMPLE STREAM @
8 S/C
! VA8SMP(X),VB8SMP(X),VC8SMP(X) - DEC FILTERED POTENTIAL SAMPLE STREAM
@ 8 S/C
! IARSMP(X),IBRSMP(X),ICRSMP(X) - REPLICA FILTERED CURRENTS @ 8 S/C
! DFTSIN(X), DFTCOS(X) - EIGHT DFT SIN AND COS COEFFICIENTS 60HZ
! D12SIN(X),D12COS(X) - EIGHT DFT SIN AND COS COEFFICIENTS 120HZ
! IA60RS,IB60RS,IC60RS - DFT SIN COMPONENTS FOR RAW CURRENT
! IA60RC,IB60RC,IC60RC - DFT COS COMPONENTS FOR RAW CURRENT
! IA60FS,IB60FS,IC60FS - 60HZ DFT SIN COMPONENTS FOR REPLICA FILTERED
CURRENT
! IA60FC,IB60FC,IC60FC - 60HZ DFT COS COMPONENTS FOR REPLICA FILTERED
CURRENT
```

```
! IA12FS,IB12FS,IC12FS - 120HZ DFT SIN COMPONENTS FOR REPLICA FILTERED
CURRENT
! IA12FC,IB12FC,IC12FC - 120HZ DFT COS COMPONENTS FOR REPLICA FILTERED
CURRENT
! VA60S,VB60S,VC60S - DFT SIN COMPONENTS FOR POTENTIAL
! VA60C,VB60C,VC60C - DFT COS COMPONENTS FOR POTENTIAL
! IA60MG,IB60MG,IC60MG - MAGNITUDE (RMS) OF 60 HZ CURRENT PHASOR
! IA60PH,IB60PH,IC60PH - PHASE ANGLE OF 60HZ CURRENT PHASOR
! VA60MG,VB60MG,VC60MG - MAGNITUDE (RMS) OF 60 HZ POTENTIAL PHASOR
! VA60PH,VB60PH,VC60PH - PHASE ANGLE OF 60HZ POTENTIAL PHASOR
! IA12MG,IB12MG,IC12MG - MAGNITUDE (RMS) OF 120 HZ CURRENT PHASORS
! I1MG,I2MG,I0MG - POSITIVE/NEGATIVE/ZERO 60HZ I SEQ. MAGNITUDES
! I1PH,I2PH,I0PH - POSITIVE/NEGATIVE/ZERO 60HZ I SEQ. PHASE ANGLE
! V1MG,V2MG,V0MG - POSITIVE/NEGATIVE/ZERO 60HZ V SEQ. MAGNITUDES
! V1PH(X),V2PH,V0PH - POSITIVE/NEGATIVE/ZERO 60HZ V SEQ. PHASE ANGLE
! IAPHSR,IBPHSR,ICPHSR - 60HZ CURRENT PHASORS
! VAPHSR,VBPHSR,VCPHSR - 60HZ POTENTIAL PHASORS
! I1PHSR,I2PHSR,I0PHSR - 60HZ CURRENT SEQUENCE COMPONENT PHASORS
! V1PHSR,V2PHSR,V0PHSR - 60HZ POTENTIAL SEQUENCE COMPONENTS PHASORS
! IARMS,IBRMS,ICRMS - TRUE RMS OF RAW CURRENT SIGNAL @ 8 S/C
! VARMS,VBRMS,VCRMS - TRUE RMS OF POTENTIAL SIGNAL @ 8 S/C
! V1FR - POSITIVE SEQUENCE VOLTAGE MEASURED FREQUENCY
! MEMR,MEMTHE,MEMA,MEMB,MEMC - POTENTIAL MEMORY FILTER PARAMETERS
! VAPOL(X),VBPOL(X),VCPOL(X) - MEMORY PHASE POTENTIALS
! VAPOLS,VBPOLS,VCPOLS - DFT SIN COMPONENTS FOR MEMORY PHASE POTENTIALS
! VAPOLC,VBPOLC,VCPOLC - DFT COS COMPONENTS FOR MEMORY PHASE POTENTIALS
! VAPOMG,VBPOMG,VCPOMG - DFT MAGNITUDE OF MEMORY PHASE POTENTIALS
! VAPOPH,VBPOPH,VCPOPH - DFT PHASE ANGLE OF MEMORY PHASE POTENTIALS
! VAMEPH,VBMEPH,VCMEPH - MEMORY POLARIZED VOLTAGE PHASORS
! V1MEM - POSITIVE SEQUENCE MEMORY POLARIZING POTENTIAL
! ZPOSPR - POSITIVE SEQUENCE IMPEDANCE PHASOR (DIRECTIONAL ELEMENT)
! ZPOSAN - POSITIVE SEQUENCE IMPEDANCE PHASOR ANGLE
! ZPOSDR - POSITIVE SEQUENCE IMPEDANCE DIRECTION (1=FORWARD)
! DIRLHA - DIRECTIONAL CHARACTERISTIC LEFT HAND ANGLE
! DIRRHA - DIRECTIONAL CHARACTERISTIC RIGHT HAND ANGLE
! V1ME59 - DIRECTIONAL ELEMENT UNDERVOLTAGE SUPERVISION
! I1ME50 - DIRECTIONAL ELEMENT UNDERCURRENT SUPERVISION
! IRMSMX - MAXIMUM RMS PHASE CURRENT
! IOC50P - INST. PHASE OVERCURRENT ELEMENT
! PRGREC(x) - PRORAMMABLE RECORDER CHANNEL X
! RECx - USER SELECTED QUANTITY FOR CHANNEL X
! CHNLx - RECORDER DATA CHANNEL TO EMTDC
! I2pI1p - RATIO OF I2 TO I1 FOR PHASE A (DLP)
! LOP27 - LOP MINIMUM PHASE VOLTAGE
! LOP27P - LOP UNDERVOLTAGE PICKUP (0.75PU)
! DI1DT - LOP RATE OF CHANGE OF CURRENT PER CYCLE
! DV1DT - LOP RATE OF CHANGE OF VOLTAGE PER CYCLE
! LOP501 - LOP 50 I1 BLOCKING
! LOP500 - LOP 50 I0 BLOCKING
! DLTV1P - LOP DV1DT > 3V/C
! DLTV1N - LOP DV1DT < -3V/C
! ILMAX - PHASE CURRENT SUPERVISION
! RZ1PST - ZONE 1 PHASE SET REACH COMPLEX
! RZ1GST - ZONE 1 GROUND SET REACH COMPLEX
! RZ2PST - ZONE 2 PHASE SET REACH COMPLEX
! RZ2GST - ZONE 2 GROUND SET REACH COMPLEX
! IAB,IBC,ICA - L-L CURRENT PHASORS
```

```
! VAB,VBC,VCA - L-L VOLTAGE PHASORS
! K0 - GROUND DISTANCE CURRENT COMPENSATION FACTOR
! IA21G,IB21G,IC21G - GROUND DISTANCE COMPENSATED CURRENTS
! S1PnpP - ZONE n PHASE p DISTANCE S1 PHASOR (IR-V)
! S2p - PHASE p S2 PHASOR (VMEMORY)
! S12zPp - S1xS2* ZONE z PHASE p
! S12zPp - S1xS2* ZONE z GROUND PHASE p
! ZnPpAG - ZONE n PHASE p OPERATING ANGLE COSINE (S1-S2)
! ZnGpAG - ZONE n GROUND PHASE p OPERATING ANGLE COSINE (S1-S2)
!
! PICK UP/DROP OUT TIMERS
!-----------------------
! T##IP(x) - TIMER ## INPUT STATE (INDEX 0=PRESENT, 1=LAST TIME STEP)
! T##DIP - TIMER ## INPUT TRANSITION (0=NONE,1=RISING,-1=FALLING)
! T##OP - TIMER ## OUPUT STATE
! T##SET - THE TIME AT WHICH THE INPUT WAS SET
! T##RST - THE TIME AT WHICH THE INPUT WAS RESET
! T##DST - TIMER ## (PRESENT TIME-T##SET)
! T##DRT - TIMER ## (PRESENT TIME-T##RST)
! T##PUS - TIMER ## PICK UP DELAY SETTING
! T##DOS - TIMER ## DROP OUT DELAY SETTING
!
!
! EXTERNAL MODEL FUNCTIONS
!------------------------
! RNDUP - ROUNDS UP A/D COUNT TO NEXT HIGHEST INTEGER
! ULIMIT - HARD LIMITER FUNCTION

!
!****************************************************
! INCLUDE FILES
!****************************************************
      INCLUDE 'nd.h'
      INCLUDE 'emtstor.h'
      INCLUDE 'emtconst.h'
      INCLUDE 'fnames.h'
      INCLUDE 's1.h'

!****************************************************
! VARIABLE DECLARATIONS
!****************************************************
!
! MODEL INPUTS
!-------------
      REAL IA,IB,IC,VA,VB,VC
!
! MODEL OUTPUTS
!--------------
      REAL CHNL1,CHNL2,CHNL3,CHNL4,CHNL5,CHNL6,CHNL7,CHNL8,CHNL9
      REAL TRIP
!
! RELAY MODEL SETTINGS
!---------------------
      REAL CTRM,PTRM
      INTEGER P211EN,P212EN,N211EN,N212EN,POEN,NOEN
      INTEGER PODC,NODC,NSEN,NSDC,LOPEN,DLPEN,DLPI2E
      INTEGER REC1,REC2,REC3,REC4,REC5,REC6,REC7,REC8,REC9
```

```
      REAL LZ1MAG,LZ1ANG,LZ0MAG,LZ0ANG,LINEKV,P211FR,P211CA
      REAL P211DC,P212FR,P212CA,P212PD,P212DC,N211FR
      REAL N211CA,N211PC,N211I0,N212FR,N212CA,N212PD
      REAL N212PC,N212I0,PO50PU,NO50PU
      REAL NS50PU,LOPI1B,LOPI0B,DLP27P,DLP50P,DLP50N
      REAL DLPI2R


!
! INTERNAL MODEL VARIABLES
!-------------------------
      REAL ADRES,RNDUP,IGAIN,VGAIN,SMPPER,DIFTIM,NXTSMP
      REAL W,LZ1R,LZ1X,LZ0X,LZ0R,TS,TSET,AF,BF,CF,RF1,RF2
      REAL IASMP(0:11),IBSMP(0:11),ICSMP(0:11)
      REAL VASMP(0:11),VBSMP(0:11),VCSMP(0:11),CMPX14
      REAL IA8SMP(0:7),IB8SMP(0:7),IC8SMP(0:7)
      REAL VA8SMP(0:7),VB8SMP(0:7),VC8SMP(0:7)
      REAL IAOLD,IBOLD,ICOLD,VAOLD,VBOLD,VCOLD
      REAL IARSMP(0:7),IBRSMP(0:7),ICRSMP(0:7)
      REAL DFTSIN(0:7),DFTCOS(0:7),D12SIN(0:7),D12COS(0:7)
      REAL IA60RC,IB60RC,IC60RC,IA60RS,IB60RS,IC60RS
      REAL IA60FC,IB60FC,IC60FC,IA60FS,IB60FS,IC60FS
      REAL IA12FC,IB12FC,IC12FC,IA12FS,IB12FS,IC12FS
      REAL VA60C,VB60C,VC60C,VA60S,VB60S,VC60S
      REAL IA60MG,IB60MG,IC60MG,IA60PH,IB60PH,IC60PH
      REAL VA60MG,VB60MG,VC60MG,VA60PH,VB60PH,VC60PH
      REAL IA12MG,IB12MG,IC12MG,I1MG(0:1),I2MG,I0MG
      REAL V1MG(0:1),V2MG,V0MG,V1PH(0:1)
      REAL IARMS,IBRMS,ICRMS,VARMS,VBRMS,VCRMS
      REAL V1FR,MEMR,MEMTHE,MEMA,MEMB,MEMC
      REAL VAPOL(0:7),VBPOL(0:7),VCPOL(0:7)
      REAL VAPOLS,VBPOLS,VCPOLS,VAPOLC,VBPOLC,VCPOLC
      REAL VAPOMG,VBPOMG,VCPOMG,VAPOPH,VBPOPH,VCPOPH,V1MEMG
      REAL ZPOSAN,DIRRHA,DIRLHA,IRMSMX,PRGREC(0:60)
      REAL LOP27,LOP27P,DI1DT,DV1DT,ILMAX
      REAL Z1PCAN,Z1GCAN,Z2PCAN,Z2GCAN
      REAL Z1PAAG,Z1PBAG,Z1PCAG
      REAL Z1GAAG,Z1GBAG,Z1GCAG
      REAL Z2PAAG,Z2PBAG,Z2PCAG
      REAL Z2GAAG,Z2GBAG,Z2GCAG
      REAL I2AI1A,I2BI1B,I2CI1C,I21MAX,IA60MR,IB60MR,IC60MR
      COMPLEX IAPHSR,IBPHSR,ICPHSR,VAPHSR,VBPHSR,VCPHSR
      COMPLEX I1PHSR,I2PHSR,I0PHSR,V1MEM,ZPOSPR,A
      COMPLEX V1PHSR,V2PHSR,V0PHSR,VAMEPH,VBMEPH,VCMEPH
      COMPLEX RZ1PST,RZ1GST,RZ2PST,RZ2GST,LZ1CPX,LZ0CPX
      COMPLEX IAB,IBC,ICA,VAB,VBC,VCA,K0,IA21G,IB21G,IC21G
      COMPLEX S1P1AP,S1P1BP,S1P1CP,S1G1AP,S1G1BP,S1G1CP
      COMPLEX S1P2AP,S1P2BP,S1P2CP,S1G2AP,S1G2BP,S1G2CP
      COMPLEX S121PA,S121PB,S121PC,S121GA,S121GB,S121GC
      COMPLEX S122PA,S122PB,S122PC,S122GA,S122GB,S122GC
      COMPLEX V1MEMB,V1MEMC,S2AG,S2BG,S2CG,CONA,CONB,CONC
      COMPLEX S2AP,S2BP,S2CP,IAPHRW,IBPHRW,ICPHRW
      INTEGER MYINDX,I,ZPOSDR
      INTEGER V1ME59,I1ME50,IOC50P,IOC50N,IOC46
      INTEGER AND163,OR164,AND165,AND174,AND176
      INTEGER OR178,OR179,AND180,OR181,OR182,AND183
      INTEGER OR184,AND188,OR187,OR199,AND185,OR189(0:1)
```

```
      INTEGER AND186,AND195,LOP501,LOP500,DLTV1P,DLTV1N
      INTEGER Z1PAZ,Z1PBZ,Z1PCZ,Z1PIAB,Z1PIBC,Z1PICA
      INTEGER AND122,AND123,AND124,OR125,AND126
      INTEGER Z1GAZ,Z1GBZ,Z1GCZ,Z1GIA,Z1GIB,Z1GIC,Z1GI0
      INTEGER AND127,AND128,AND129,OR130,AND131
      INTEGER Z2PAZ,Z2PBZ,Z2PCZ,Z2PIAB,Z2PIBC,Z2PICA
      INTEGER AND132,AND133,AND134,OR135,AND136
      INTEGER Z2GAZ,Z2GBZ,Z2GCZ,Z2GIA,Z2GIB,Z2GIC,Z2GI0
      INTEGER AND137,AND138,AND139,OR140,AND141,TRPMAX
      EXTERNAL RNDUP,ULIMIT,PITOPI
      LOGICAL TIMEZERO

!
! PICK UP/DROP OUT TIMERS
!------------------------
      INTEGER T01IP(0:1),T01OP,T01DIP
      REAL T01PUS,T01DOS,T01SET,T01RST,T01DST,T01DRT
      INTEGER T02IP(0:1),T02OP,T02DIP
      REAL T02PUS,T02DOS,T02SET,T02RST,T02DST,T02DRT
      INTEGER T03IP(0:1),T03OP,T03DIP
      REAL T03PUS,T03DOS,T03SET,T03RST,T03DST,T03DRT
      INTEGER T04IP(0:1),T04OP,T04DIP
      REAL T04PUS,T04DOS,T04SET,T04RST,T04DST,T04DRT
      INTEGER T05IP(0:1),T05OP,T05DIP
      REAL T05PUS,T05DOS,T05SET,T05RST,T05DST,T05DRT
      INTEGER T06IP(0:1),T06OP,T06DIP
      REAL T06PUS,T06DOS,T06SET,T06RST,T06DST,T06DRT
      INTEGER T07IP(0:1),T07OP,T07DIP
      REAL T07PUS,T07DOS,T07SET,T07RST,T07DST,T07DRT
      INTEGER T08IP(0:1),T08OP,T08DIP
      REAL T08PUS,T08DOS,T08SET,T08RST,T08DST,T08DRT
      INTEGER T09IP(0:1),T09OP,T09DIP
      REAL T09PUS,T09DOS,T09SET,T09RST,T09DST,T09DRT
      INTEGER T10IP(0:1),T10OP,T10DIP
      REAL T10PUS,T10DOS,T10SET,T10RST,T10DST,T10DRT
      INTEGER T11IP(0:1),T11OP,T11DIP
      REAL T11PUS,T11DOS,T11SET,T11RST,T11DST,T11DRT
      INTEGER T12IP(0:1),T12OP,T12DIP
      REAL T12PUS,T12DOS,T12SET,T12RST,T12DST,T12DRT
      INTEGER T13IP(0:1),T13OP,T13DIP
      REAL T13PUS,T13DOS,T13SET,T13RST,T13DST,T13DRT
      INTEGER T14IP(0:1),T14OP,T14DIP
      REAL T14PUS,T14DOS,T14SET,T14RST,T14DST,T14DRT
      INTEGER T15IP(0:1),T15OP,T15DIP
      REAL T15PUS,T15DOS,T15SET,T15RST,T15DST,T15DRT
      INTEGER T16IP(0:1),T16OP,T16DIP
      REAL T16PUS,T16DOS,T16SET,T16RST,T16DST,T16DRT
      INTEGER T17IP(0:1),T17OP,T17DIP
      REAL T17PUS,T17DOS,T17SET,T17RST,T17DST,T17DRT

!*********************************************
! RETRIEVE STORED DATA
!*********************************************
      MYINDX=NSTORF
      NSTORF=NSTORF+229

      ADRES=STORF(MYINDX)
```

```
      SMPPER=STORF(MYINDX+1)
      NXTSMP=STORF(MYINDX+2)
      W=STORF(MYINDX+3)
      IGAIN=STORF(MYINDX+4)
      VGAIN=STORF(MYINDX+5)
      SDC=STORF(MYINDX+6)
      TS=STORF(MYINDX+7)
      TSET=STORF(MYINDX+8)
      AF=STORF(MYINDX+9)
      BF=STORF(MYINDX+10)
      CF=STORF(MYINDX+11)
      RF1=STORF(MYINDX+12)
      RF2=STORF(MYINDX+13)
!
!     DFT COEFFICIENTS
!---------------------
      DO 5 I=0,7
      DFTSIN(I)=STORF(MYINDX+14+I)
      DFTCOS(I)=STORF(MYINDX+22+I)
5     CONTINUE
!
!     TIMER STATES
!----------------
      T01IP(1)=STORF(MYINDX+30)
      T02IP(1)=STORF(MYINDX+31)
!
!     V1 PHASE
!------------
      V1PH(1)=STORF(MYINDX+32)
!
!     7 STORED DECIMATED SAMPLES
!------------------------------
      DO 6 I=0,7
      IA8SMP(I)=STORF(MYINDX+33+I)
      IB8SMP(I)=STORF(MYINDX+41+I)
      IC8SMP(I)=STORF(MYINDX+49+I)
      VA8SMP(I)=STORF(MYINDX+57+I)
      VB8SMP(I)=STORF(MYINDX+65+I)
      VC8SMP(I)=STORF(MYINDX+73+I)
!
!     7 STORED REPLICA FILTER SAMPLES
!-----------------------------------
      IARSMP(I)=STORF(MYINDX+81+I)
      IBRSMP(I)=STORF(MYINDX+89+I)
      ICRSMP(I)=STORF(MYINDX+97+I)
!
!      7 STORED POLARIZATION VOLTAGES
!-----------------------------------
      VAPOL(I)=STORF(MYINDX+105+I)
      VBPOL(I)=STORF(MYINDX+113+I)
      VCPOL(I)=STORF(MYINDX+121+I)
6     CONTINUE
!
!     STORED UNFILTERED SAMPLES
!-----------------------------
      DO 7 I=0,11
      IASMP(I)=STORF(MYINDX+129+I)
```

```
      IBSMP(I)=STORF(MYINDX+141+I)
      ICSMP(I)=STORF(MYINDX+153+I)
      VASMP(I)=STORF(MYINDX+165+I)
      VBSMP(I)=STORF(MYINDX+177+I)
      VCSMP(I)=STORF(MYINDX+189+I)
7     CONTINUE
      SDC=STORF(MYINDX+201)
      IAOLD=STORF(MYINDX+202)
      IBOLD=STORF(MYINDX+203)
      ICOLD=STORF(MYINDX+204)
      VAOLD=STORF(MYINDX+205)
      VBOLD=STORF(MYINDX+206)
      VCOLD=STORF(MYINDX+207)
      I1MG(1)=STORF(MYINDX+208)
      V1MG(1)=STORF(MYINDX+209)
      OR189(1)=STORF(MYINDX+210)
      T03IP(1)=STORF(MYINDX+211)
      T04IP(1)=STORF(MYINDX+212)
      T05IP(1)=STORF(MYINDX+213)
      T06IP(1)=STORF(MYINDX+214)
      T07IP(1)=STORF(MYINDX+215)
      T08IP(1)=STORF(MYINDX+216)
      T09IP(1)=STORF(MYINDX+217)
      T10IP(1)=STORF(MYINDX+218)
      T11IP(1)=STORF(MYINDX+219)
      T12IP(1)=STORF(MYINDX+220)
      T13IP(1)=STORF(MYINDX+221)
      T14IP(1)=STORF(MYINDX+222)
      T15IP(1)=STORF(MYINDX+223)
      T16IP(1)=STORF(MYINDX+224)
      T17IP(1)=STORF(MYINDX+225)
      Z1PCAN=STORF(MYINDX+226)
      Z1GCAN=STORF(MYINDX+227)
      Z2PCAN=STORF(MYINDX+228)
      Z2GCAN=STORF(MYINDX+229)


!*****************************************************
! VARIABLE INITIALIZATION FOR FIRST TIMESTEP ONLY
!*****************************************************
      IF ( TIMEZERO ) THEN
        SDC=0
      ADRES=.0006105
      SMPPER=.0001736111
        NXTSMP=0.0
        A=(-0.5,0.8660254)
        CONA=(0.866,0.5)
        CONB=(0,-1.0)
        CONC=(-0.866,0.5)
        IAOLD=2.5
        IBOLD=2.5
        ICOLD=2.5
        VAOLD=2.5
        VBOLD=2.5
        VCOLD=2.5
        IGAIN=0.04170
        VGAIN=0.05211
```

```
        LOP27P=0.75*LINEKV*1000.0/(1.73205*PTRM)
        Z1PCAN=COS(P211CA)
        Z1GCAN=COS(N211CA)
        Z2PCAN=COS(P212CA)
        Z2GCAN=COS(N212CA)
      LZ1X=(SIN(LZ1ANG))*LZ1MAG
      LZ1R=(COS(LZ1ANG))*LZ1MAG
        LZ1CPX=CMPLX(LZ1R,LZ1X)
      LZ0X=(SIN(LZ0ANG))*LZ0MAG
      LZ0R=(COS(LZ0ANG))*LZ0MAG
        LZ0CPX=CMPLX(LZ0R,LZ0X)
        RZ1PST=(P211FR/ABS(LZ1CPX))*LZ1CPX
        RZ1GST=(N211FR/ABS(LZ1CPX))*LZ1CPX
        RZ2PST=(P212FR/ABS(LZ1CPX))*LZ1CPX
        RZ2GST=(N212FR/ABS(LZ1CPX))*LZ1CPX
        K0=(LZ0CPX-LZ1CPX)/(3.0*LZ1CPX)

!
!     REPLICA FILTER CONSTANTS
!-----------------------------
      W=2.0*PI_*60.0
            TS=0.002083333
      TSET=LZ1X/(2.0*PI_*60.0*LZ1R)
        AF=1.0/(2.0*COS(W*TS/2.0))
        BF=(W*TSET)/(2.0*SIN(W*TS/2.0))
      CF=SQRT(1.0+(W*TSET)**2.0)
      RF1=(AF+BF)/CF
      RF2=(AF-BF)/CF

!
!     DFT COEFFICIENTS
!---------------------
       DO 9 I=0,7
       DFTSIN(I)=SIN(W*(8-I)/480.0)
       DFTCOS(I)=COS(W*(8-I)/480.0)
       D12SIN(I)=SIN(2.0*W*(8-I)/480.0)
       D12COS(I)=COS(2.0*W*(8-I)/480.0)
9      CONTINUE
!
!     TIMERS
!----------------------
      T01SET=9999999.9
      T01RST=0.0
      T02SET=9999999.9
      T02RST=0.0
      T03SET=9999999.9
      T03RST=0.0
      T04SET=9999999.9
      T04RST=0.0
      T05SET=9999999.9
      T05RST=0.0
      T06SET=9999999.9
      T06RST=0.0
      T07SET=9999999.9
      T07RST=0.0
      T08SET=9999999.9
      T08RST=0.0
```

```
      T09SET=9999999.9
      T09RST=0.0
      T10SET=9999999.9
      T10RST=0.0
      T11SET=9999999.9
      T11RST=0.0
      T12SET=9999999.9
      T12RST=0.0
      T13SET=9999999.9
      T13RST=0.0
      T14SET=9999999.9
      T14RST=0.0
      T15SET=9999999.9
      T15RST=0.0
      T16SET=9999999.9
      T16RST=0.0
      T17SET=9999999.9
      T17RST=0.0
      END IF

!
!*******************************************************
! IDEAL EXTERNAL CT AND PT
!*******************************************************
!
      IA=IA*1000.0/CTRM
      IB=IB*1000.0/CTRM
      IC=IC*1000.0/CTRM
      VA=VA*1000.0/PTRM
      VB=VB*1000.0/PTRM
      VC=VC*1000.0/PTRM

!=======================================================
! LPRO MODEL
!=======================================================
!
!***************************
! CURRENT INPUT CT & RESISTOR
!***************************
!
      IA=IA/21.51
      IB=IB/21.51
      IC=IC/21.51
!
!****************
! VOLTAGE INPUT PT
!****************
!
      VA=VA/48.8
      VB=VB/48.8
      VC=VC/48.8

!
!********************************************
! ANTI-ALIASING FILTER
!********************************************
```

```
!
      IA=CMPX14(1,1.82,0.59,8331.503,IA)
      IB=CMPX14(1,1.82,0.59,8331.503,IB)
      IC=CMPX14(1,1.82,0.59,8331.503,IC)
      VA=CMPX14(1,1.82,0.59,8331.503,VA)
      VB=CMPX14(1,1.82,0.59,8331.503,VB)
      VC=CMPX14(1,1.82,0.59,8331.503,VC)

!**********************************************************
! INPUT AMP REFERENCE SHIFT & LOWER LIMIT & GAIN PATH ADJUSTMENT
!**********************************************************
!
      IA=(IA*.1731)+2.5
      IB=(IB*.1731)+2.5
      IC=(IC*.1731)+2.5
      VA=(VA*.3144)+2.5
      VB=(VB*.3144)+2.5
      VC=(VC*.3144)+2.5

      IA=ULIMIT(6.0,0.1,IA)
      IB=ULIMIT(6.0,0.1,IB)
      IC=ULIMIT(6.0,0.1,IC)
      VA=ULIMIT(6.0,0.1,VA)
      VB=ULIMIT(6.0,0.1,VB)
      VC=ULIMIT(6.0,0.1,VC)
!
!****************************************
! RELAY PROCESSING CONTROL (96 x PER CYCLE)
!****************************************
!
      DIFTIM=TIME-NXTSMP
!
!>>>>>SAMPLE CONTROL(D1)
      IF (DIFTIM.GE.0) THEN

!****************************************
! A/D CONVERTER
!****************************************
!
! INTERPOLATING SAMPLER @ 96 SAMPLES/CYCLE
!----------------------------------------
      IASMP(0)=((DELT-DIFTIM)*(IA-IAOLD)/DELT)+IAOLD
      IBSMP(0)=((DELT-DIFTIM)*(IB-IBOLD)/DELT)+IBOLD
      ICSMP(0)=((DELT-DIFTIM)*(IC-ICOLD)/DELT)+ICOLD
      VASMP(0)=((DELT-DIFTIM)*(VA-VAOLD)/DELT)+VAOLD
      VBSMP(0)=((DELT-DIFTIM)*(VB-VBOLD)/DELT)+VBOLD
      VCSMP(0)=((DELT-DIFTIM)*(VC-VCOLD)/DELT)+VCOLD
!
! QUANTIZE AND ROUND UP THE SAMPLES
!---------------------------------
!
      IASMP(0)=(((IASMP(0)-2.5)/2.5)*4096.0)-0.5
      IBSMP(0)=(((IBSMP(0)-2.5)/2.5)*4096.0)-0.5
      ICSMP(0)=(((ICSMP(0)-2.5)/2.5)*4096.0)-0.5
      VASMP(0)=(((VASMP(0)-2.5)/2.5)*4096.0)-0.5
      VBSMP(0)=(((VBSMP(0)-2.5)/2.5)*4096.0)-0.5
      VCSMP(0)=(((VCSMP(0)-2.5)/2.5)*4096.0)-0.5
```

```
!
      IASMP(0)=RNDUP(IASMP(0))
      IBSMP(0)=RNDUP(IBSMP(0))
      ICSMP(0)=RNDUP(ICSMP(0))
      VASMP(0)=RNDUP(VASMP(0))
      VBSMP(0)=RNDUP(VBSMP(0))
      VCSMP(0)=RNDUP(VCSMP(0))
!
! LIMIT A/D OUTPUT
!----------------
!
      IASMP(0)=ULIMIT(4095.0,-4096.0,IASMP(0))
      IBSMP(0)=ULIMIT(4095.0,-4096.0,IBSMP(0))
      ICSMP(0)=ULIMIT(4095.0,-4096.0,ICSMP(0))
      VASMP(0)=ULIMIT(4095.0,-4096.0,VASMP(0))
      VBSMP(0)=ULIMIT(4095.0,-4096.0,VBSMP(0))
      VCSMP(0)=ULIMIT(4095.0,-4096.0,VCSMP(0))
!
!*******************************************
! SAMPLE DECIMATION TO 8 SAMPLES/CYCLE
!*******************************************
!
! CONVERT SAMPLE TO FLOATING POINT NUMBER (SECONDARY LEVEL)
!---------------------------------------------------------
      IASMP(0)=IASMP(0)*IGAIN
      IBSMP(0)=IBSMP(0)*IGAIN
      ICSMP(0)=ICSMP(0)*IGAIN
      VASMP(0)=VASMP(0)*VGAIN
      VBSMP(0)=VBSMP(0)*VGAIN
      VCSMP(0)=VCSMP(0)*VGAIN
!
! 8 SAMPLE PER CYCLE PROCESSING INSIDE THIS LOOP
!-----------------------------------------------
      SDC=SDC+1

      IF (SDC.EQ.12) THEN
      SDC=0

!                                                         !
! SAMPLE DECIMATION FILTER
!-------------------------
      IA8SMP(0)=0.0
      IB8SMP(0)=0.0
      IC8SMP(0)=0.0
      VA8SMP(0)=0.0
      VB8SMP(0)=0.0
      VC8SMP(0)=0.0

      DO 10 I=0,11
      IA8SMP(0)=1.0/12.0*IASMP(I)+IA8SMP(0)
      IB8SMP(0)=1.0/12.0*IBSMP(I)+IB8SMP(0)
      IC8SMP(0)=1.0/12.0*ICSMP(I)+IC8SMP(0)
      VA8SMP(0)=1.0/12.0*VASMP(I)+VA8SMP(0)
      VB8SMP(0)=1.0/12.0*VBSMP(I)+VB8SMP(0)
      VC8SMP(0)=1.0/12.0*VCSMP(I)+VC8SMP(0)
10    CONTINUE
```

```
!***************************************
! CURRENT REPLICA FILTER
!***************************************
      IARSMP(0)=RF1*IA8SMP(0)+RF2*IA8SMP(1)
      IBRSMP(0)=RF1*IB8SMP(0)+RF2*IB8SMP(1)
      ICRSMP(0)=RF1*IC8SMP(0)+RF2*IC8SMP(1)

!****************************************
! DFT - EXTRACT 60HZ COMPONENTS
!****************************************
      IA60RS=0
      IB60RS=0
      IC60RS=0
      IA60RC=0
      IB60RC=0
      IC60RC=0

      IA60FS=0
      IB60FS=0
      IC60FS=0
      IA60FC=0
      IB60FC=0
      IC60FC=0

      VA60S=0
      VB60S=0
      VC60S=0
      VA60C=0
      VB60C=0
      VC60C=0
!
      DO 15 I=0,7
!
!     RAW CURRENT FOR PHASE ANGLE
!     ---------------------------
      IA60RS=IA8SMP(I)*DFTSIN(I)+IA60RS
      IB60RS=IB8SMP(I)*DFTSIN(I)+IB60RS
      IC60RS=IC8SMP(I)*DFTSIN(I)+IC60RS
      IA60RC=IA8SMP(I)*DFTCOS(I)+IA60RC
      IB60RC=IB8SMP(I)*DFTCOS(I)+IB60RC
      IC60RC=IC8SMP(I)*DFTCOS(I)+IC60RC
!
!     REPLICA CURRENT FOR MAGNITUDE
!     -----------------------------
      IA60FS=IARSMP(I)*DFTSIN(I)+IA60FS
      IB60FS=IBRSMP(I)*DFTSIN(I)+IB60FS
      IC60FS=ICRSMP(I)*DFTSIN(I)+IC60FS
      IA60FC=IARSMP(I)*DFTCOS(I)+IA60FC
      IB60FC=IBRSMP(I)*DFTCOS(I)+IB60FC
      IC60FC=ICRSMP(I)*DFTCOS(I)+IC60FC
!
!     POTENTIALS
!     -----------------------------
      VA60S=VA8SMP(I)*DFTSIN(I)+VA60S
      VB60S=VB8SMP(I)*DFTSIN(I)+VB60S
      VC60S=VC8SMP(I)*DFTSIN(I)+VC60S
      VA60C=VA8SMP(I)*DFTCOS(I)+VA60C
```

```
      VB60C=VB8SMP(I)*DFTCOS(I)+VB60C
      VC60C=VC8SMP(I)*DFTCOS(I)+VC60C
15    CONTINUE

!
!     CALCULATE 60HZ I & V RMS PHASOR & MAGNITUDE
!-----------------------------------------------
      IA60MG=1.026/(4.0*SQRT(2.0))*SQRT(IA60FS**2+IA60FC**2)
      IB60MG=1.026/(4.0*SQRT(2.0))*SQRT(IB60FS**2+IB60FC**2)
      IC60MG=1.026/(4.0*SQRT(2.0))*SQRT(IC60FS**2+IC60FC**2)

      IA60MR=1.026/(4.0*SQRT(2.0))*SQRT(IA60RS**2+IA60RC**2)
      IB60MR=1.026/(4.0*SQRT(2.0))*SQRT(IB60RS**2+IB60RC**2)
      IC60MR=1.026/(4.0*SQRT(2.0))*SQRT(IC60RS**2+IC60RC**2)
!
      IA60PH=ATAN2(IA60RC,IA60RS)
      IB60PH=ATAN2(IB60RC,IB60RS)
      IC60PH=ATAN2(IC60RC,IC60RS)
!
      VA60MG=1.026/(4.0*SQRT(2.0))*SQRT(VA60S**2+VA60C**2)
      VB60MG=1.026/(4.0*SQRT(2.0))*SQRT(VB60S**2+VB60C**2)
      VC60MG=1.026/(4.0*SQRT(2.0))*SQRT(VC60S**2+VC60C**2)
!
      VA60PH=ATAN2(VA60C,VA60S)
      VB60PH=ATAN2(VB60C,VB60S)
      VC60PH=ATAN2(VC60C,VC60S)

      IAPHSR=CMPLX(IA60MG*COS(IA60PH),IA60MG*SIN(IA60PH))
      IBPHSR=CMPLX(IB60MG*COS(IB60PH),IB60MG*SIN(IB60PH))
      ICPHSR=CMPLX(IC60MG*COS(IC60PH),IC60MG*SIN(IC60PH))
      IAPHRW=CMPLX(IA60MR*COS(IA60PH),IA60MR*SIN(IA60PH))
      IBPHRW=CMPLX(IB60MR*COS(IB60PH),IB60MR*SIN(IB60PH))
      ICPHRW=CMPLX(IC60MR*COS(IC60PH),IC60MR*SIN(IC60PH))
      VAPHSR=CMPLX(VA60MG*COS(VA60PH),VA60MG*SIN(VA60PH))
      VBPHSR=CMPLX(VB60MG*COS(VB60PH),VB60MG*SIN(VB60PH))
      VCPHSR=CMPLX(VC60MG*COS(VC60PH),VC60MG*SIN(VC60PH))

      PRGREC(1)=IA60MG*CTRM
      PRGREC(2)=IB60MG*CTRM
      PRGREC(3)=IC60MG*CTRM
      PRGREC(4)=VA60MG*PTRM
      PRGREC(5)=VB60MG*PTRM
      PRGREC(6)=VC60MG*PTRM

!*****************************************
! DFT - EXTRACT 120HZ I MAGNITUDE
!*****************************************
      IF (DLPI2E.EQ.1)THEN

      IA12FS=0
      IB12FS=0
      IC12FS=0
      IA12FC=0
      IB12FC=0
      IC12FC=0

      DO 17 I=0,7
```

```
      IA12FS=IARSMP(I)*D12SIN(I)+IA12FS
      IB12FS=IBRSMP(I)*D12SIN(I)+IB12FS
      IC12FS=ICRSMP(I)*D12SIN(I)+IC12FS
      IA12FC=IARSMP(I)*D12COS(I)+IA12FC
      IB12FC=IBRSMP(I)*D12COS(I)+IB12FC
      IC12FC=ICRSMP(I)*D12COS(I)+IC12FC
17    CONTINUE

!
!     CALCULATE 120HZ I RMS PHASOR MAGNITUDE
!-------------------------------------------------
      IA12MG=(1.10*0.542)/(4.0*SQRT(2.0))*
     +SQRT(IA12FS**2+IA12FC**2)
      IB12MG=(1.10*0.542)/(4.0*SQRT(2.0))*
     +SQRT(IB12FS**2+IB12FC**2)
      IC12MG=(1.10*0.542)/(4.0*SQRT(2.0))*
     +SQRT(IC12FS**2+IC12FC**2)

      PRGREC(7)=IA12MG*CTRM
      PRGREC(8)=IB12MG*CTRM
      PRGREC(9)=IC12MG*CTRM

      END IF
!
!*******************************************************
! CALCULATE 60HZ I & V SYMMETRICAL COMPONENT(RMS) PHASORS
!*******************************************************
      I1PHSR=1.0/3.0*(IAPHRW+A*IBPHRW+(A**2.0)*ICPHRW)
      I2PHSR=1.0/3.0*(IAPHRW+(A**2.0)*IBPHRW+A*ICPHRW)
      I0PHSR=1.0/3.0*(IAPHRW+IBPHRW+ICPHRW)
      V1PHSR=1.0/3.0*(VAPHSR+A*VBPHSR+(A**2.0)*VCPHSR)
      V2PHSR=1.0/3.0*(VAPHSR+(A**2.0)*VBPHSR+A*VCPHSR)
      V0PHSR=1.0/3.0*(VAPHSR+VBPHSR+VCPHSR)

      I1MG(0)=ABS(I1PHSR)
      I2MG=ABS(I2PHSR)
      I0MG=3.0*ABS(I0PHSR)
      V1MG(0)=(INT(ABS(V1PHSR)*100.0))/100.0
      V2MG=ABS(V2PHSR)
      V0MG=ABS(V0PHSR)

      PRGREC(10)=I1MG(0)*CTRM
      PRGREC(11)=I2MG*CTRM
      PRGREC(12)=I0MG*CTRM
      PRGREC(13)=V1MG(0)*PTRM
      PRGREC(14)=V2MG*PTRM
      PRGREC(15)=V0MG*PTRM

!*********************************
! CALCULATE 60HZ I & V TRUE RMS
!*********************************
      IARMS=0
      IBRMS=0
      ICRMS=0
      VARMS=0
      VBRMS=0
      VCRMS=0
```

```
      DO 18 I=0,7
      IARMS=IARMS+(IA8SMP(I))**2.0
      IBRMS=IBRMS+(IB8SMP(I))**2.0
      ICRMS=ICRMS+(IC8SMP(I))**2.0
      VARMS=VARMS+(VA8SMP(I))**2.0
      VBRMS=VBRMS+(VB8SMP(I))**2.0
      VCRMS=VCRMS+(VC8SMP(I))**2.0
18    CONTINUE

      IARMS=1.026*SQRT(IARMS/8.0)
      IBRMS=1.026*SQRT(IBRMS/8.0)
      ICRMS=1.026*SQRT(ICRMS/8.0)
      VARMS=1.026*SQRT(VARMS/8.0)
      VBRMS=1.026*SQRT(VBRMS/8.0)
      VCRMS=1.026*SQRT(VCRMS/8.0)

      PRGREC(16)=IARMS*CTRM
      PRGREC(17)=IBRMS*CTRM
      PRGREC(18)=ICRMS*CTRM
      PRGREC(19)=VARMS*PTRM
      PRGREC(20)=VBRMS*PTRM
      PRGREC(21)=VCRMS*PTRM

!***********************************************
! POSITIVE SEQUENCE MEMORY POLARIZING POTENTIAL
!***********************************************
!
!     DETERMINE FREQUENCY
!-----------------------
      V1FR=60.0+(V1PH(0)-V1PH(1))/(1.0/480.0)
!
!     DETERMINE FILTER COEFFICIENTS
!---------------------------------
      MEMR=0.987
      MEMTHE=(360.0*PI_*V1FR)/86400.0
      MEMA=2.0*MEMR*COS(MEMTHE)
      MEMB=MEMR**2.0
      MEMC=(1.0-MEMR)*SQRT(MEMR)
!
!     POLARIZATION FILTER
!------------------------
      VAPOL(0)=(VA8SMP(0)-VA8SMP(2)+(MEMA/MEMC)*VAPOL(1)-(MEMB/MEMC)*
     +VAPOL(2))*MEMC
      VBPOL(0)=(VB8SMP(0)-VB8SMP(2)+(MEMA/MEMC)*VBPOL(1)-(MEMB/MEMC)*
     +VBPOL(2))*MEMC
      VCPOL(0)=(VC8SMP(0)-VC8SMP(2)+(MEMA/MEMC)*VCPOL(1)-(MEMB/MEMC)*
     +VCPOL(2))*MEMC
!
!     EXTRACT 60HZ RMS PHASOR USING DFT
!--------------------------------------
      VAPOLS=0
      VBPOLS=0
      VCPOLS=0
      VAPOLC=0
      VBPOLC=0
      VCPOLC=0
```

```
      DO 19 I=0,7
      VAPOLS=VAPOL(I)*DFTSIN(I)+VAPOLS
      VBPOLS=VBPOL(I)*DFTSIN(I)+VBPOLS
      VCPOLS=VCPOL(I)*DFTSIN(I)+VCPOLS
      VAPOLC=VAPOL(I)*DFTCOS(I)+VAPOLC
      VBPOLC=VBPOL(I)*DFTCOS(I)+VBPOLC
      VCPOLC=VCPOL(I)*DFTCOS(I)+VCPOLC
19    CONTINUE
!
      VAPOMG=1.026/(4.0*SQRT(2.0))*SQRT(VAPOLS**2+VAPOLC**2)
      VBPOMG=1.026/(4.0*SQRT(2.0))*SQRT(VBPOLS**2+VBPOLC**2)
      VCPOMG=1.026/(4.0*SQRT(2.0))*SQRT(VCPOLS**2+VCPOLC**2)
!
      VAPOPH=ATAN2(VAPOLC,VAPOLS)
      VBPOPH=ATAN2(VBPOLC,VBPOLS)
      VCPOPH=ATAN2(VCPOLC,VCPOLS)
!
      VAMEPH=CMPLX(VAPOMG*COS(VAPOPH),VAPOMG*SIN(VAPOPH))
      VBMEPH=CMPLX(VBPOMG*COS(VBPOPH),VBPOMG*SIN(VBPOPH))
      VCMEPH=CMPLX(VCPOMG*COS(VCPOPH),VCPOMG*SIN(VCPOPH))
!
!     EXTRACT POSTIVE SEQUENCE PHASOR
!-----------------------------------
      V1MEM=1.0/3.0*(VAMEPH+A*VBMEPH+(A**2)*VCMEPH)
      V1MEMG=ABS(V1MEM)
      PRGREC(22)=V1MEMG*PTRM
!*************************************************
! PROTECTION ELEMENTS
!*************************************************
!
!     DIRECTIONAL ELEMENT (OR164)
!-----------------------------
      ZPOSPR=V1MEM/I1PHSR
      ZPOSAN=ATAN2(AIMAG(ZPOSPR),REAL(ZPOSPR))
      DIRRHA=0.0-(PI_/2.0-LZ1ANG)
      DIRLHA=PI_-(PI_/2.0-LZ1ANG)
      IF((ZPOSAN.GT.DIRRHA).AND.(ZPOSAN.LT.DIRLHA))THEN
        ZPOSDR=1
      ELSE
        ZPOSDR=0
      END IF

      IF(V1MEMG.GE.2.0) THEN
         V1ME59=1
      ELSE
         V1ME59=0
      END IF

      IF(I1MG(0).GE.0.2) THEN
         I1ME50=1
      ELSE
         I1ME50=0
      END IF

      IF((ZPOSDR.EQ.1).AND.(V1ME59.EQ.1))THEN
        AND163=1
      ELSE
```

```
          AND163=0
        END IF

        IF((AND163.EQ.1).OR.(V1ME59.EQ.0).OR.(I1ME50.EQ.0))THEN
          OR164=1
        ELSE
          OR164=0
        END IF
        PRGREC(23)=OR164
!
!       PHASE INST OVERCURRENT ELEMENT (T01OP)
!---------------------------------------

        IRMSMX=MAX(IARMS,IBRMS,ICRMS)
        IF(IRMSMX.GE.PO50PU)THEN
          IOC50P=1
        ELSE
          IOC50P=0
        END IF
        IF(POEN.EQ.1) THEN
        IF((IOC50P.EQ.1).AND.(OR164.EQ.1)) THEN
          AND165=1
        ELSE
          AND165=0
        END IF

        IF(PODC.EQ.0)THEN
          AND165=IOC50P
        END IF
!
!       PICKUP TIMER ENABLED IF DIRECTIONAL
!---------------------------------------
        IF(PODC.EQ.1)THEN
!
!       TIMER INPUT SIGNAL
!----------------------
          T01IP(0)=AND165
!
!       TIMER SETTINGS
!------------------
          T01PUS=0.050
          T01DOS=0.0
          T01DIP=T01IP(0)-T01IP(1)
             IF(T01DIP.EQ.1)THEN
               T01SET=TIME
               T01RST=9999999.9
             END IF
             IF(T01DIP.EQ.-1)THEN
               T01SET=9999999.9
               T01RST=TIME
             END IF
             T01DST=TIME-T01SET
             T01DRT=TIME-T01RST
             IF(T01DST.GE.T01PUS)THEN
               T01OP=1
             END IF
             IF(T01DRT.GE.T01DOS)THEN
```

```
            T01OP=0
          END IF

      ELSE
        T01OP=AND165
      END IF
      END IF
      PRGREC(24)=T01OP
!
!     NEUTRAL INST OVERCURRENT ELEMENT (T02OP)
!---------------------------------------------

      IF(IOMG.GE.NO50PU)THEN
        IOC50N=1
      ELSE
        IOC50N=0
      END IF
      IF(NOEN.EQ.1) THEN
      IF((IOC50N.EQ.1).AND.(OR164.EQ.1)) THEN
        AND174=1
      ELSE
        AND174=0
      END IF

      IF(NODC.EQ.0)THEN
        AND174=IOC50N
      END IF
!
!     PICKUP TIMER ENABLED IF DIRECTIONAL
!-----------------------------------------
      IF(NODC.EQ.1)THEN
!
!     TIMER INPUT SIGNAL
!-----------------------
        T02IP(0)=AND174
!
!     TIMER SETTINGS
!-------------------
        T02PUS=0.050
        T02DOS=0.0
        T02DIP=T02IP(0)-T02IP(1)
           IF(T02DIP.EQ.1)THEN
             T02SET=TIME
             T02RST=9999999.9
           END IF
           IF(T02DIP.EQ.-1)THEN
             T02SET=9999999.9
             T02RST=TIME
           END IF
           T02DST=TIME-T02SET
           T02DRT=TIME-T02RST
           IF(T02DST.GE.T02PUS)THEN
             T02OP=1
           END IF
           IF(T02DRT.GE.T02DOS)THEN
             T02OP=0
           END IF
```

```
      ELSE
        T02OP=AND174
      END IF
      END IF
      PRGREC(25)=T02OP
!
!     I2 INST OVERCURRENT ELEMENT (T03OP)
!---------------------------------------------
      IF(NSEN.EQ.1) THEN
      IF(I2MG.GE.NS50PU)THEN
        IOC46=1
      ELSE
        IOC46=0
      END IF
      IF((IOC46.EQ.1).AND.(OR164.EQ.1)) THEN
        AND176=1
      ELSE
        AND176=0
      END IF

      IF(NSDC.EQ.0)THEN
        AND176=IOC46
      END IF
!
!     PICKUP TIMER ENABLED IF DIRECTIONAL
!----------------------------------------
      IF(NSDC.EQ.1)THEN
!
!     TIMER INPUT SIGNAL
!-----------------------
        T03IP(0)=AND176
!
!     TIMER SETTINGS
!-------------------
        T03PUS=0.050
        T03DOS=0.0
        T03DIP=T03IP(0)-T03IP(1)
           IF(T03DIP.EQ.1)THEN
             T03SET=TIME
             T03RST=9999999.9
           END IF
           IF(T03DIP.EQ.-1)THEN
             T03SET=9999999.9
             T03RST=TIME
           END IF
           T03DST=TIME-T03SET
           T03DRT=TIME-T03RST
           IF(T03DST.GE.T03PUS)THEN
             T03OP=1
           END IF
           IF(T03DRT.GE.T03DOS)THEN
             T03OP=0
           END IF

      ELSE
        T03OP=AND176
```

```
      END IF
      END IF

      PRGREC(26)=T03OP

!
!     DEAD LINE PICKUP
!----------------------
      IF (DLPEN.EQ.1)THEN

      IF (V1MG(0).LE.DLP27P)THEN

!
!     TIMER INPUT SIGNAL
!-----------------------
        T15IP(0)=1
!
!     TIMER SETTINGS
!-------------------
        T15PUS=2.0
        T15DOS=0.0
        T15DIP=T15IP(0)-T15IP(1)
           IF(T15DIP.EQ.1)THEN
             T15SET=TIME
             T15RST=9999999.9
           END IF
           IF(T15DIP.EQ.-1)THEN
             T15SET=9999999.9
             T15RST=TIME
           END IF
           T15DST=TIME-T15SET
           T15DRT=TIME-T15RST
           IF(T15DST.GE.T15PUS)THEN
             T15OP=1
           END IF
           IF(T15DRT.GE.T15DOS)THEN
             T15OP=0
           END IF
      END IF

      IF (IRMSMX.GE.0.2)THEN
      OR178=1
      ELSE
      OR178=0
      END IF

!
!     TIMER INPUT SIGNAL
!-----------------------
        T16IP(0)=OR178
!
!     TIMER SETTINGS
!-------------------
        T16PUS=0.200
        T16DOS=0.010
        T16DIP=T16IP(0)-T16IP(1)
           IF(T16DIP.EQ.1)THEN
```

```
          T16SET=TIME
          T16RST=9999999.9
        END IF
        IF(T16DIP.EQ.-1)THEN
          T16SET=9999999.9
          T16RST=TIME
        END IF
        T16DST=TIME-T16SET
        T16DRT=TIME-T16RST
        IF(T16DST.GE.T16PUS)THEN
          T16OP=1
        END IF
        IF(T16DRT.GE.T16DOS)THEN
          T16OP=0
        END IF

      IF((IRMSMX.GE.DLP50P).OR.(I0MG.GE.DLP50N))THEN
      OR179=1
      ELSE
      OR179=0
      END IF

      I2AI1A=IA12MG/IA60MG
      I2BI1B=IB12MG/IB60MG
      I2CI1C=IC12MG/IC60MG
      I21MAX=MAX(I2AI1A,I2BI1B,I2CI1C)

      IF(I21MAX.GE.DLPI2R)THEN
      OR181=1
      ELSE
      OR181=0
      END IF

      IF((IOC50P.EQ.1).OR.(IOC50N.EQ.1))THEN
      OR182=1
      ELSE
      OR182=0
      END IF

      IF ((OR182.EQ.0).AND.(OR181.EQ.1).AND.(DLPI2E.EQ.1))THEN
      AND183=1
      ELSE
      AND183=0
      END IF

      IF((T15OP.EQ.1).AND.(OR178.EQ.1).AND.(T16OP.EQ.0).AND.
     +(OR179.EQ.1).AND.(AND183.EQ.0))THEN
      AND180=1
      ELSE
      AND180=0
      END IF

      END IF
      PRGREC(27)=AND180
!
!     LOSS OF POTENTIAL LOGIC (LOP60)
!------------------------------------
```

```
      IF(LOPEN.EQ.1)THEN
      LOP27=MIN(VARMS,VBRMS,VCRMS)
      DI1DT=8.0*ABS((I1MG(0)-I1MG(1))/TS)
      DV1DT=8.0*(V1MG(0)-V1MG(1))/TS

      IF (LOP27.LE.LOP27P) THEN
         OR184=1
      ELSE
         OR184=0
      ENDIF
      IF(I1MG(0).GE.LOPI1B) THEN
         LOP501=1
      ELSE
         LOP501=0
      END IF
      IF(I0MG.GE.LOPI0B) THEN
         LOP500=1
      ELSE
         LOP500=0
      END IF
      IF(DI1DT.GT.0.1)THEN
         T06IP(0)=1
      ELSE
         T06IP(0)=0
      END IF
      IF(DV1DT.GT.3.0)THEN
         DLTV1P=1
      ELSE
         DLTV1P=0
      END IF
      IF(DV1DT.LT.-3.0)THEN
         DLTV1N=1
      ELSE
         DLTV1N=0
      END IF
      ILMAX=MAX(IA60MG,IB60MG,IC60MG)
      IF(ILMAX.GT.0.2)THEN
         OR199=1
      ELSE
         OR199=0
      END IF
!
!     TIMER INPUT SIGNAL
!-----------------------
!        SEE ABOVE
!
!     TIMER SETTINGS
!-------------------
        T06PUS=0.0
        T06DOS=0.020
        T06DIP=T06IP(0)-T06IP(1)
           IF(T06DIP.EQ.1)THEN
             T06SET=TIME
             T06RST=9999999.9
           END IF
           IF(T06DIP.EQ.-1)THEN
             T06SET=9999999.9
```

```
          T06RST=TIME
        END IF
        T06DST=TIME-T06SET
        T06DRT=TIME-T06RST
        IF(T06DST.GE.T06PUS)THEN
          T06OP=1
        END IF
        IF(T06DRT.GE.T06DOS)THEN
          T06OP=0
        END IF
      IF((LOP501.EQ.1).OR.(LOP500.EQ.1))THEN
        OR187=1
      ELSE
        OR187=0
      END IF
      IF((DLTV1N.EQ.1).AND.(T06OP.EQ.0).AND.(OR187.EQ.0))THEN
        AND188=1
      ELSE
        AND188=0
      END IF
!
!     TIMER INPUT SIGNAL
!-----------------------
      T07IP(0)=AND188
!
!     TIMER SETTINGS
!-------------------
        T07PUS=0.005
        T07DOS=0.012
        T07DIP=T07IP(0)-T07IP(1)
           IF(T07DIP.EQ.1)THEN
             T07SET=TIME
             T07RST=9999999.9
           END IF
           IF(T07DIP.EQ.-1)THEN
             T07SET=9999999.9
             T07RST=TIME
           END IF
           T07DST=TIME-T07SET
           T07DRT=TIME-T07RST
           IF(T07DST.GE.T07PUS)THEN
             T07OP=1
           END IF
           IF(T07DRT.GE.T07DOS)THEN
             T07OP=0
           END IF
      IF((OR189(1).EQ.1).AND.(OR184.EQ.1).AND.(DLTV1P.EQ.0))THEN
         AND185=1
      ELSE
         AND185=0
      END IF
      IF((DLTV1P.EQ.0).AND.(OR199.EQ.1))THEN
         AND186=1
      ELSE
         AND186=0
      END IF
      IF((AND185.EQ.1).OR.(T07OP.EQ.1))THEN
```

```
         OR189(0)=1
      ELSE
         OR189(0)=0
      END IF
      IF((OR184.EQ.1).AND.(OR189(0).EQ.1).AND.(AND186.EQ.1))THEN
         AND195=1
      ELSE
         AND195=0
      END IF
!
!     TIMER INPUT SIGNAL
!-----------------------
      T08IP(0)=AND195
!
!     TIMER SETTINGS
!-------------------
        T08PUS=0.000
        T08DOS=0.017
        T08DIP=T08IP(0)-T08IP(1)
           IF(T08DIP.EQ.1)THEN
             T08SET=TIME
             T08RST=9999999.9
           END IF
           IF(T08DIP.EQ.-1)THEN
             T08SET=9999999.9
             T08RST=TIME
           END IF
           T08DST=TIME-T08SET
           T08DRT=TIME-T08RST
           IF(T08DST.GE.T08PUS)THEN
             T08OP=1
           END IF
           IF(T08DRT.GE.T08DOS)THEN
             T08OP=0
           END IF
      ELSE
      T08OP=0
      END IF

      PRGREC(28)=T08OP
!
!     DISTANCE ELEMENTS
!----------------------
      IAB=IAPHSR-IBPHSR
      IBC=IBPHSR-ICPHSR
      ICA=ICPHSR-IAPHSR
      VAB=VAPHSR-VBPHSR
      VBC=VBPHSR-VCPHSR
      VCA=VCPHSR-VAPHSR
      IA21G=IAPHSR+(K0*3.0*I0PHSR)
      IB21G=IBPHSR+(K0*3.0*I0PHSR)
      IC21G=ICPHSR+(K0*3.0*I0PHSR)
      V1MEMB=V1MEM*A*A
      V1MEMC=V1MEM*A
      S2AG=CONJG(V1MEM)
      S2BG=CONJG(V1MEMB)
      S2CG=CONJG(V1MEMC)
```

```
      S2AP=V1MEM*CONA
      S2BP=V1MEM*CONB
      S2CP=V1MEM*CONC
      S2AP=CONJG(S2AP)
      S2BP=CONJG(S2BP)
      S2CP=CONJG(S2CP)
!
!     ZONE 1 PHASE
!-----------------
      IF(P211EN.EQ.1)THEN

      S1P1AP=(IAB*RZ1PST)-VAB
      S1P1BP=(IBC*RZ1PST)-VBC
      S1P1CP=(ICA*RZ1PST)-VCA

      S121PA=S1P1AP*S2AP
      S121PB=S1P1BP*S2BP
      S121PC=S1P1CP*S2CP

      Z1PAAG=REAL(S121PA)/ABS(S121PA)
      Z1PBAG=REAL(S121PB)/ABS(S121PB)
      Z1PCAG=REAL(S121PC)/ABS(S121PC)

      PRGREC(29)=ACOS(Z1PAAG)*180.0/PI_
      PRGREC(30)=ACOS(Z1PBAG)*180.0/PI_
      PRGREC(31)=ACOS(Z1PCAG)*180.0/PI_

      IF(Z1PAAG.GE.Z1PCAN)THEN
       Z1PAZ=1
      ELSE
       Z1PAZ=0
      END IF

      IF(Z1PBAG.GE.Z1PCAN)THEN
       Z1PBZ=1
      ELSE
       Z1PBZ=0
      END IF
      IF(Z1PCAG.GE.Z1PCAN)THEN
       Z1PCZ=1
      ELSE
       Z1PCZ=0
      END IF

      IF(ABS(IAB).GE.P211DC)THEN
      Z1PIAB=1
      ELSE
      Z1PIAB=0
      END IF

      IF(ABS(IBC).GE.P211DC)THEN
      Z1PIBC=1
      ELSE
      Z1PIBC=0
      END IF

      IF(ABS(ICA).GE.P211DC)THEN
```

```
      Z1PICA=1
      ELSE
      Z1PICA=0
      END IF

      IF((Z1PAZ.EQ.1).AND.(Z1PIAB.EQ.1).AND.(OR164.EQ.1))THEN
      AND122=1
      ELSE
      AND122=0
      END IF

      IF((Z1PBZ.EQ.1).AND.(Z1PIBC.EQ.1).AND.(OR164.EQ.1))THEN
      AND123=1
      ELSE
      AND123=0
      END IF

      IF((Z1PCZ.EQ.1).AND.(Z1PICA.EQ.1).AND.(OR164.EQ.1))THEN
      AND124=1
      ELSE
      AND124=0
      END IF

      IF((AND122.EQ.1).OR.(AND123.EQ.1).OR.(AND124.EQ.1))THEN
      OR125=1
      ELSE
      OR125=0
      END IF

      IF((OR125.EQ.1).AND.(T08OP.EQ.0))THEN
      AND126=1
      ELSE
      AND126=0
      END IF

!
!     TIMER INPUT SIGNAL
!-----------------------
      T09IP(0)=AND126
!
!     TIMER SETTINGS
!-------------------
        T09PUS=0.000
        T09DOS=0.004
        T09DIP=T09IP(0)-T09IP(1)
           IF(T09DIP.EQ.1)THEN
             T09SET=TIME
             T09RST=9999999.9
           END IF
           IF(T09DIP.EQ.-1)THEN
             T09SET=9999999.9
             T09RST=TIME
           END IF
           T09DST=TIME-T09SET
           T09DRT=TIME-T09RST
           IF(T09DST.GE.T09PUS)THEN
             T09OP=1
```

```
          END IF
          IF(T09DRT.GE.T09DOS)THEN
            T09OP=0
          END IF
     ELSE
     T09OP=0

     END IF

     PRGREC(32)=T09OP

!
!    ZONE 1 GROUND
!------------------
     IF(N211EN.EQ.1)THEN

     S1G1AP=(IA21G*RZ1GST)-VAPHSR
     S1G1BP=(IB21G*RZ1GST)-VBPHSR
     S1G1CP=(IC21G*RZ1GST)-VCPHSR

     S121GA=S1G1AP*S2AG
     S121GB=S1G1BP*S2BG
     S121GC=S1G1CP*S2CG

     Z1GAAG=(REAL(S121GA)/ABS(S121GA))
     Z1GBAG=(REAL(S121GB)/ABS(S121GB))
     Z1GCAG=(REAL(S121GC)/ABS(S121GC))

     PRGREC(33)=ACOS(Z1GAAG)*180.0/PI_
     PRGREC(34)=ACOS(Z1GBAG)*180.0/PI_
     PRGREC(35)=ACOS(Z1GCAG)*180.0/PI_

     IF(Z1GAAG.GE.Z1GCAN)THEN
      Z1GAZ=1
     ELSE
      Z1GAZ=0
     END IF

     IF(Z1GBAG.GE.Z1GCAN)THEN
      Z1GBZ=1
     ELSE
      Z1GBZ=0
     END IF
     IF(Z1GCAG.GE.Z1GCAN)THEN
      Z1GCZ=1
     ELSE
      Z1GCZ=0
     END IF

     IF(ABS(IAPHSR).GE.N211PC)THEN
     Z1GIA=1
     ELSE
     Z1GIA=0
     END IF

     IF(ABS(IBPHSR).GE.N211PC)THEN
     Z1GIB=1
```

```
      ELSE
      Z1GIB=0
      END IF

      IF(ABS(ICPHSR).GE.N211PC)THEN
      Z1GIC=1
      ELSE
      Z1GIC=0
      END IF

      IF(IOMG.GE.N211IO)THEN
      Z1GIO=1
      ELSE
      Z1GIO=0
      END IF

      IF((Z1GAZ.EQ.1).AND.(Z1GIA.EQ.1).AND.(Z1GIO.EQ.1)
     +.AND.(OR164.EQ.1))THEN
      AND127=1
      ELSE
      AND127=0
      END IF

      IF((Z1GBZ.EQ.1).AND.(Z1GIB.EQ.1).AND.(Z1GIO.EQ.1)
     +.AND.(OR164.EQ.1))THEN
      AND128=1
      ELSE
      AND128=0
      END IF

      IF((Z1GCZ.EQ.1).AND.(Z1GIC.EQ.1).AND.(Z1GIO.EQ.1)
     +.AND.(OR164.EQ.1))THEN
      AND129=1
      ELSE
      AND129=0
      END IF

      IF((AND127.EQ.1).OR.(AND128.EQ.1).OR.(AND129.EQ.1))THEN
      OR130=1
      ELSE
      OR130=0
      END IF

      IF((OR130.EQ.1).AND.(T08OP.EQ.0))THEN
      AND131=1
      ELSE
      AND131=0
      END IF
!
!     TIMER INPUT SIGNAL
!-----------------------
      T10IP(0)=AND131
!
!     TIMER SETTINGS
!-------------------
         T10PUS=0.000
         T10DOS=0.004
```

```
      T10DIP=T10IP(0)-T10IP(1)
        IF(T10DIP.EQ.1)THEN
          T10SET=TIME
          T10RST=9999999.9
        END IF
        IF(T10DIP.EQ.-1)THEN
          T10SET=9999999.9
          T10RST=TIME
        END IF
        T10DST=TIME-T10SET
        T10DRT=TIME-T10RST
        IF(T10DST.GE.T10PUS)THEN
          T10OP=1
        END IF
        IF(T10DRT.GE.T10DOS)THEN
          T10OP=0
        END IF
      ELSE
      T10OP=0

      END IF

      PRGREC(36)=T10OP
!
!     ZONE 2 PHASE
!-----------------
      IF(P212EN.EQ.1)THEN

      S1P2AP=(IAB*RZ2PST)-VAB
      S1P2BP=(IBC*RZ2PST)-VBC
      S1P2CP=(ICA*RZ2PST)-VCA

      S122PA=S1P2AP*S2AP
      S122PB=S1P2BP*S2BP
      S122PC=S1P2CP*S2CP

      Z2PAAG=(REAL(S122PA)/ABS(S122PA))
      Z2PBAG=(REAL(S122PB)/ABS(S122PB))
      Z2PCAG=(REAL(S122PC)/ABS(S122PC))

      PRGREC(37)=ACOS(Z2PAAG)*180.0/PI_
      PRGREC(38)=ACOS(Z2PBAG)*180.0/PI_
      PRGREC(39)=ACOS(Z2PCAG)*180.0/PI_

      IF(Z2PAAG.GE.Z2PCAN)THEN
       Z2PAZ=1
      ELSE
       Z2PAZ=0
      END IF

      IF(Z2PBAG.GE.Z2PCAN)THEN
       Z2PBZ=1
      ELSE
       Z2PBZ=0
      END IF
      IF(Z2PCAG.GE.Z2PCAN)THEN
       Z2PCZ=1
```

```
      ELSE
       Z2PCZ=0
      END IF

      IF(ABS(IAB).GE.P212DC)THEN
      Z2PIAB=1
      ELSE
      Z2PIAB=0
      END IF

      IF(ABS(IBC).GE.P212DC)THEN
      Z2PIBC=1
      ELSE
      Z2PIBC=0
      END IF

      IF(ABS(ICA).GE.P212DC)THEN
      Z2PICA=1
      ELSE
      Z2PICA=0
      END IF

      IF((Z2PAZ.EQ.1).AND.(Z2PIAB.EQ.1).AND.(OR164.EQ.1))THEN
      AND132=1
      ELSE
      AND132=0
      END IF

      IF((Z2PBZ.EQ.1).AND.(Z2PIBC.EQ.1).AND.(OR164.EQ.1))THEN
      AND133=1
      ELSE
      AND133=0
      END IF

      IF((Z2PCZ.EQ.1).AND.(Z2PICA.EQ.1).AND.(OR164.EQ.1))THEN
      AND134=1
      ELSE
      AND134=0
      END IF

      IF((AND132.EQ.1).OR.(AND133.EQ.1).OR.(AND134.EQ.1))THEN
      OR135=1
      ELSE
      OR135=0
      END IF

      IF((OR135.EQ.1).AND.(T08OP.EQ.0))THEN
      AND136=1
      ELSE
      AND136=0
      END IF

!
!     TIMER INPUT SIGNAL
!-----------------------
      T11IP(0)=AND136
!
```

```
!      TIMER SETTINGS
!-------------------
         T11PUS=0.000
         T11DOS=0.004
         T11DIP=T11IP(0)-T11IP(1)
            IF(T11DIP.EQ.1)THEN
              T11SET=TIME
              T11RST=9999999.9
            END IF
            IF(T11DIP.EQ.-1)THEN
              T11SET=9999999.9
              T11RST=TIME
            END IF
            T11DST=TIME-T11SET
            T11DRT=TIME-T11RST
            IF(T11DST.GE.T11PUS)THEN
              T11OP=1
            END IF
            IF(T11DRT.GE.T11DOS)THEN
              T11OP=0
            END IF

       PRGREC(40)=T11OP

!
!      TIMER INPUT SIGNAL
!----------------------
       T12IP(0)=T11OP
!
!      TIMER SETTINGS
!-------------------
         T12PUS=P212PD
         T12DOS=0.00
         T12DIP=T12IP(0)-T12IP(1)
            IF(T12DIP.EQ.1)THEN
              T12SET=TIME
              T12RST=9999999.9
            END IF
            IF(T12DIP.EQ.-1)THEN
              T12SET=9999999.9
              T12RST=TIME
            END IF
            T12DST=TIME-T12SET
            T12DRT=TIME-T12RST
            IF(T12DST.GE.T12PUS)THEN
              T12OP=1
            END IF
            IF(T12DRT.GE.T12DOS)THEN
              T12OP=0
            END IF

       ELSE
       T12OP=0
       END IF

       PRGREC(41)=T12OP
```

```
!
!     ZONE 2 GROUND
!------------------
      IF(N212EN.EQ.1)THEN

      S1G2AP=(IA21G*RZ2GST)-VAPHSR
      S1G2BP=(IB21G*RZ2GST)-VBPHSR
      S1G2CP=(IC21G*RZ2GST)-VCPHSR

      S122GA=S1G2AP*S2AG
      S122GB=S1G2BP*S2BG
      S122GC=S1G2CP*S2CG

      Z2GAAG=(REAL(S122GA)/ABS(S122GA))
      Z2GBAG=(REAL(S122GB)/ABS(S122GB))
      Z2GCAG=(REAL(S122GC)/ABS(S122GC))

      PRGREC(42)=ACOS(Z2GAAG)*180.0/PI_
      PRGREC(43)=ACOS(Z2GBAG)*180.0/PI_
      PRGREC(44)=ACOS(Z2GCAG)*180.0/PI_

      IF(Z2GAAG.GE.Z2GCAN)THEN
       Z2GAZ=1
      ELSE
       Z2GAZ=0
      END IF

      IF(Z2GBAG.GE.Z2GCAN)THEN
       Z2GBZ=1
      ELSE
       Z2GBZ=0
      END IF
      IF(Z2GCAG.GE.Z2GCAN)THEN
       Z2GCZ=1
      ELSE
       Z2GCZ=0
      END IF

      IF(ABS(IAPHSR).GE.N212PC)THEN
      Z2GIA=1
      ELSE
      Z2GIA=0
      END IF

      IF(ABS(IBPHSR).GE.N212PC)THEN
      Z2GIB=1
      ELSE
      Z2GIB=0
      END IF

      IF(ABS(ICPHSR).GE.N212PC)THEN
      Z2GIC=1
      ELSE
      Z2GIC=0
      END IF

      IF(IOMG.GE.N212IO)THEN
```

```
      Z2GIO=1
      ELSE
      Z2GIO=0
      END IF

      IF((Z2GAZ.EQ.1).AND.(Z2GIA.EQ.1).AND.(Z2GIO.EQ.1)
     +.AND.(OR164.EQ.1))THEN
      AND137=1
      ELSE
      AND137=0
      END IF

      IF((Z2GBZ.EQ.1).AND.(Z2GIB.EQ.1).AND.(Z2GIO.EQ.1)
     +.AND.(OR164.EQ.1))THEN
      AND138=1
      ELSE
      AND138=0
      END IF

      IF((Z2GCZ.EQ.1).AND.(Z2GIC.EQ.1).AND.(Z2GIO.EQ.1)
     +.AND.(OR164.EQ.1))THEN
      AND139=1
      ELSE
      AND139=0
      END IF

      IF((AND137.EQ.1).OR.(AND138.EQ.1).OR.(AND139.EQ.1))THEN
      OR140=1
      ELSE
      OR140=0
      END IF

      IF((OR140.EQ.1).AND.(T08OP.EQ.0))THEN
      AND141=1
      ELSE
      AND141=0
      END IF
!
!     TIMER INPUT SIGNAL
!-----------------------
      T13IP(0)=AND141
!
!     TIMER SETTINGS
!-------------------
        T13PUS=0.000
        T13DOS=0.004
        T13DIP=T13IP(0)-T13IP(1)
           IF(T13DIP.EQ.1)THEN
             T13SET=TIME
             T13RST=9999999.9
           END IF
           IF(T13DIP.EQ.-1)THEN
             T13SET=9999999.9
             T13RST=TIME
           END IF
           T13DST=TIME-T13SET
           T13DRT=TIME-T13RST
```

```
          IF(T13DST.GE.T13PUS)THEN
            T13OP=1
          END IF
          IF(T13DRT.GE.T13DOS)THEN
            T13OP=0
          END IF

     PRGREC(45)=T13OP
!
!    TIMER INPUT SIGNAL
!-----------------------
     T14IP(0)=T13OP
!
!    TIMER SETTINGS
!-------------------
       T14PUS=N212PD
       T14DOS=0.000
       T14DIP=T14IP(0)-T14IP(1)
          IF(T14DIP.EQ.1)THEN
            T14SET=TIME
            T14RST=9999999.9
          END IF
          IF(T14DIP.EQ.-1)THEN
            T14SET=9999999.9
            T14RST=TIME
          END IF
          T14DST=TIME-T14SET
          T14DRT=TIME-T14RST
          IF(T14DST.GE.T14PUS)THEN
            T14OP=1
          END IF
          IF(T14DRT.GE.T14DOS)THEN
            T14OP=0
          END IF

     ELSE
     T14OP=0

     END IF

     PRGREC(46)=T14OP

!****************************************
! MEASURED PHASE AND GROUND IMPEDANCES
!****************************************
     PRGREC(47)=ABS(VAB/IAB)
     PRGREC(48)=ABS(VBC/IBC)
     PRGREC(49)=ABS(VCA/ICA)
     PRGREC(50)=ABS(VAPHSR/IA21G)
     PRGREC(51)=ABS(VBPHSR/IB21G)
     PRGREC(52)=ABS(VCPHSR/IC21G)

!*****************************************
! SHIFT ALL STORED VALUES
!*****************************************
!
!    SHIFT LOP MEMORY ITEMS
```

```
!--------------------------
      I1MG(1)=I1MG(0)
      V1MG(1)=V1MG(0)
      OR189(1)=OR189(0)
!
!     SHIFT STORED DFT CONSTANTS
!------------------------------
      DFTCOS(8)=DFTCOS(7)
      DFTSIN(8)=DFTSIN(7)
      D12COS(8)=D12COS(7)
      D12SIN(8)=D12SIN(7)
      DO 20 I=7,1,-1
      DFTCOS(I)=DFTCOS(I-1)
      DFTSIN(I)=DFTSIN(I-1)
      D12COS(I)=D12COS(I-1)
      D12SIN(I)=D12SIN(I-1)
20    CONTINUE
      DFTCOS(0)=DFTCOS(8)
      DFTSIN(0)=DFTSIN(8)
      D12COS(0)=D12COS(8)
      D12SIN(0)=D12SIN(8)
!
!     SHIFT STORED TIMER STATES
!-----------------------------
      T01IP(1)=T01IP(0)
      T02IP(1)=T02IP(0)
      T03IP(1)=T03IP(0)
      T04IP(1)=T04IP(0)
      T05IP(1)=T05IP(0)
      T06IP(1)=T06IP(0)
      T07IP(1)=T07IP(0)
      T08IP(1)=T08IP(0)
      T09IP(1)=T09IP(0)
      T10IP(1)=T10IP(0)
      T11IP(1)=T11IP(0)
      T12IP(1)=T12IP(0)
      T13IP(1)=T13IP(0)
      T14IP(1)=T14IP(0)
      T15IP(1)=T15IP(0)
      T16IP(1)=T16IP(0)
!
!     SHIFT 1 STORED V1 PHASE
!---------------------------
      V1PH(1)=V1PH(0)

      DO 30 I=7,1,-1
!
!     SHIFT 7 STORED DECIMATED SAMPLES
!-----------------------------------
      IA8SMP(I)=IA8SMP(I-1)
      IB8SMP(I)=IB8SMP(I-1)
      IC8SMP(I)=IC8SMP(I-1)
      VA8SMP(I)=VA8SMP(I-1)
      VB8SMP(I)=VB8SMP(I-1)
      VC8SMP(I)=VC8SMP(I-1)
!
!     SHIFT 7 STORED REPLICA FILTER SAMPLES
```

```
!------------------------------------------
      IARSMP(I)=IARSMP(I-1)
      IBRSMP(I)=IBRSMP(I-1)
      ICRSMP(I)=ICRSMP(I-1)
!
!     SHIFT 7 STORED POLARIZATION VOLTAGES
!------------------------------------------
      VAPOL(I)=VAPOL(I-1)
      VBPOL(I)=VBPOL(I-1)
      VCPOL(I)=VCPOL(I-1)
30    CONTINUE

!>>>>>>>END SAMPLE CONTROL 8 S/C
      END IF
!************************************
! SHIFT 11 STORED UNFILTERED SAMPLES
!************************************
      DO 40 I=11,1,-1
      IASMP(I)=IASMP(I-1)
      IBSMP(I)=IBSMP(I-1)
      ICSMP(I)=ICSMP(I-1)
      VASMP(I)=VASMP(I-1)
      VBSMP(I)=VBSMP(I-1)
      VCSMP(I)=VCSMP(I-1)
40    CONTINUE
      NXTSMP=TIME-DIFTIM+SMPPER

!>>>>>>>END SAMPLE CONTROL 96 S/C
      END IF

!*****************************************
! TRIPPING LOGIC
!*****************************************
      TRPMAX=MAX0(T01OP,T02OP,T03OP,AND180,T09OP,T10OP,T12OP,T14OP)
      IF(TRPMAX.EQ.1)THEN
      T17IP(0)=1
      ELSE
      T17IP(0)=0
      END IF

!
!     TIMER SETTINGS(OUTPUT RELAY DELAY=3ms)
!------------------------------------------
        T17PUS=0.003
        T17DOS=0.100
        T17DIP=T17IP(0)-T17IP(1)
           IF(T17DIP.EQ.1)THEN
             T17SET=TIME
             T17RST=9999999.9
           END IF
           IF(T17DIP.EQ.-1)THEN
             T17SET=9999999.9
             T17RST=TIME
           END IF
           T17DST=TIME-T17SET
           T17DRT=TIME-T17RST
           IF(T17DST.GE.T17PUS)THEN
```

```
          T17OP=1
        END IF
        IF(T17DRT.GE.T17DOS)THEN
          T17OP=0
        END IF
      T17IP(1)=T17IP(0)

!*******************************************************
! CONFIGURE USER RECORDING
!*******************************************************
      TRIP=T17OP
      CHNL1=PRGREC(REC1)
      CHNL2=PRGREC(REC2)
      CHNL3=PRGREC(REC3)
      CHNL4=PRGREC(REC4)
      CHNL5=PRGREC(REC5)
      CHNL6=PRGREC(REC6)
      CHNL7=PRGREC(REC7)
      CHNL8=PRGREC(REC8)
      CHNL9=PRGREC(REC9)
!
!     STORE COPY OF SIGNALS FOR NEXT SAMPLE
!-----------------------------------------
      IAOLD=IA
      IBOLD=IB
      ICOLD=IC
      VAOLD=VA
      VBOLD=VB
      VCOLD=VC
!
!***************************************************
! STORE DATA FOR THE NEXT TIME STEP & RETURN TO MAIN
!***************************************************
      STORF(MYINDX)=ADRES
      STORF(MYINDX+1)=SMPPER
      STORF(MYINDX+2)=NXTSMP
      STORF(MYINDX+3)=W
      STORF(MYINDX+4)=IGAIN
      STORF(MYINDX+5)=VGAIN
      STORF(MYINDX+6)=SDC
      STORF(MYINDX+7)=TS
      STORF(MYINDX+8)=TSET
      STORF(MYINDX+9)=AF
      STORF(MYINDX+10)=BF
      STORF(MYINDX+11)=CF
      STORF(MYINDX+12)=RF1
      STORF(MYINDX+13)=RF2
!
!     DFT COEFFICIENTS
!     ----------------
      DO 55 I=0,7
      STORF(MYINDX+14+I)=DFTSIN(I)
      STORF(MYINDX+22+I)=DFTCOS(I)
55     CONTINUE
!
!     TIMER STATES
!----------------
```

```
      STORF(MYINDX+30)=T01IP(1)
      STORF(MYINDX+31)=T02IP(1)
!
!     V1 PHASE
!-------------
      STORF(MYINDX+32)=V1PH(1)
!
!     7 STORED DECIMATED SAMPLES
!------------------------------
      DO 65 I=0,7
      STORF(MYINDX+33+I)=IA8SMP(I)
      STORF(MYINDX+41+I)=IB8SMP(I)
      STORF(MYINDX+49+I)=IC8SMP(I)
      STORF(MYINDX+57+I)=VA8SMP(I)
      STORF(MYINDX+65+I)=VB8SMP(I)
      STORF(MYINDX+73+I)=VC8SMP(I)
!
!     7 STORED REPLICA FILTER SAMPLES
!-----------------------------------
      STORF(MYINDX+81+I)=IARSMP(I)
      STORF(MYINDX+89+I)=IBRSMP(I)
      STORF(MYINDX+97+I)=ICRSMP(I)
!
!      7 STORED POLARIZATION VOLTAGES
!------------------------------------
      STORF(MYINDX+105+I)=VAPOL(I)
      STORF(MYINDX+113+I)=VBPOL(I)
      STORF(MYINDX+121+I)=VCPOL(I)
65    CONTINUE
!
!     STORED UNFILTERED SAMPLES
!-----------------------------
      DO 75 I=0,11
      STORF(MYINDX+129+I)=IASMP(I)
      STORF(MYINDX+141+I)=IBSMP(I)
      STORF(MYINDX+153+I)=ICSMP(I)
      STORF(MYINDX+165+I)=VASMP(I)
      STORF(MYINDX+177+I)=VBSMP(I)
      STORF(MYINDX+189+I)=VCSMP(I)
75    CONTINUE
      STORF(MYINDX+201)=SDC
      STORF(MYINDX+202)=IAOLD
      STORF(MYINDX+203)=IBOLD
      STORF(MYINDX+204)=ICOLD
      STORF(MYINDX+205)=VAOLD
      STORF(MYINDX+206)=VBOLD
      STORF(MYINDX+207)=VCOLD
      STORF(MYINDX+208)=I1MG(1)
      STORF(MYINDX+209)=V1MG(1)
      STORF(MYINDX+210)=OR189(1)
      STORF(MYINDX+211)=T03IP(1)
      STORF(MYINDX+212)=T04IP(1)
      STORF(MYINDX+213)=T05IP(1)
      STORF(MYINDX+214)=T06IP(1)
      STORF(MYINDX+215)=T07IP(1)
      STORF(MYINDX+216)=T08IP(1)
      STORF(MYINDX+217)=T09IP(1)
```

```
      STORF(MYINDX+218)=T10IP(1)
      STORF(MYINDX+219)=T11IP(1)
      STORF(MYINDX+220)=T12IP(1)
      STORF(MYINDX+221)=T13IP(1)
      STORF(MYINDX+222)=T14IP(1)
      STORF(MYINDX+223)=T15IP(1)
      STORF(MYINDX+224)=T16IP(1)
      STORF(MYINDX+225)=T17IP(1)
      STORF(MYINDX+226)=Z1PCAN
      STORF(MYINDX+227)=Z1GCAN
      STORF(MYINDX+228)=Z2PCAN
      STORF(MYINDX+229)=Z2GCAN

!
      RETURN
      END



!
!**************************************
! FUNCTION RNDUP
! --ROUNDS A SAMPLE UP TO NEXT INTEGER
!**************************************
!
      REAL FUNCTION RNDUP(X)
      REAL X,A
!
      A=ABS(X-INT(X))
      IF(A.LT.0.1) THEN
      RNDUP=INT(X)
        ELSE
        IF (X.LT.0) THEN
          RNDUP=INT(X)
        ELSE
          RNDUP=INT(X)+1
        END IF
      END IF
      END


!
!**************************************
! FUNCTION ULIMIT
! --USER LIMIT FUNCTION
!**************************************
!
      REAL FUNCTION ULIMIT(HI,LO,X)
      REAL HI,LO,X
!
      IF((X.LE.HI).AND.(X.GE.LO)) THEN
      ULIMIT=X
        ELSE
        IF (X.GT.HI) THEN
             ULIMIT=HI
          END IF
          IF (X.LT.LO)THEN
```

```
        ULIMIT=LO
  END IF
END IF
END
```

Appendix II – LPRO Model Recorder Signal Assignments

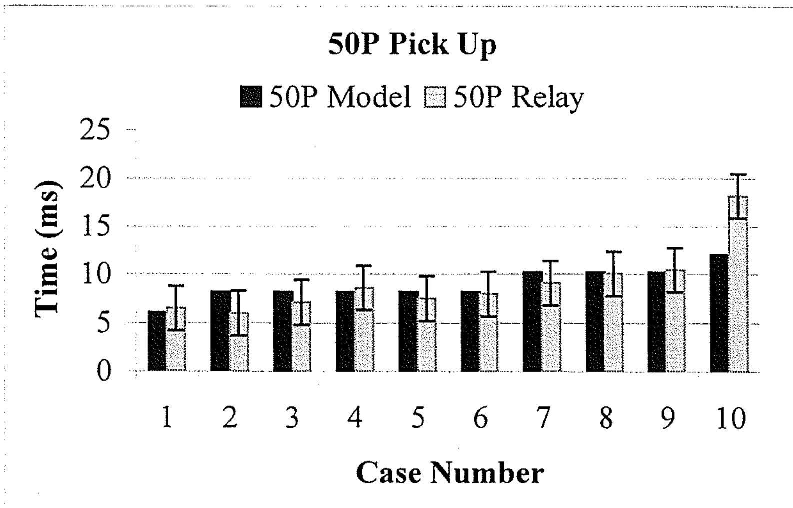
Signal Assignment	Signal Quantity	Signal Units
1	Ia 60Hz Magnitude	Amps primary
2	Ib 60Hz Magnitude	Amps primary
3	Ic 60Hz Magnitude	Amps primary
4	Va 60Hz Magnitude	Volts primary
5	Vb 60Hz Magnitude	Volts primary
6	Vc 60Hz Magnitude	Volts primary
7	Ia 120Hz Magnitude	Amps primary
8	Ib 120Hz Magnitude	Amps primary
9	Ic 120Hz Magnitude	Amps primary
10	I1 Magnitude	Amps primary
11	I2 Magnitude	Amps primary
12	3Io Magnitude	Amps primary
13	V1 Magnitude	Volts primary
14	V2 Magnitude	Volts primary
15	Vo Magnitude	Volts primary
16	Ia RMS	Amps primary
17	Ib RMS	Amps primary
18	Ic RMS	Amps primary
19	Va RMS	Volts primary
20	Vb RMS	Volts primary
21	Vc RMS	Volts primary
22	V1 memory magnitude	Volts primary
23	Directional Element	1=Forward/0=Block
24	Phase Inst O/C (50P)	1=Trip/0=Reset
25	Neutral Inst O/C (50N)	1=Trip/0=Reset
26	Negative Sequence Inst O/C (50/46)	1=Trip/0=Reset
27	Dead Line Pickup	1=Trip/0=Reset
28	Loss of Potential (LOP)	1=LOP/0=Reset
29	Zone 1 A Phase S1-S2 Angle	Degrees
30	Zone 1 B Phase S1-S2 Angle	Degrees
31	Zone 1 C Phase S1-S2 Angle	Degrees
32	Zone 1 Phase Operate	1=Trip/0=Reset
33	Zone 1 A Ground S1-S2 Angle	Degrees
34	Zone 1 B Ground S1-S2 Angle	Degrees
35	Zone 1 C Ground S1-S2 Angle	Degrees
36	Zone 1 Ground Operate	1=Trip/0=Reset
37	Zone 2 A Phase S1-S2 Angle	Degrees
38	Zone 2 B Phase S1-S2 Angle	Degrees

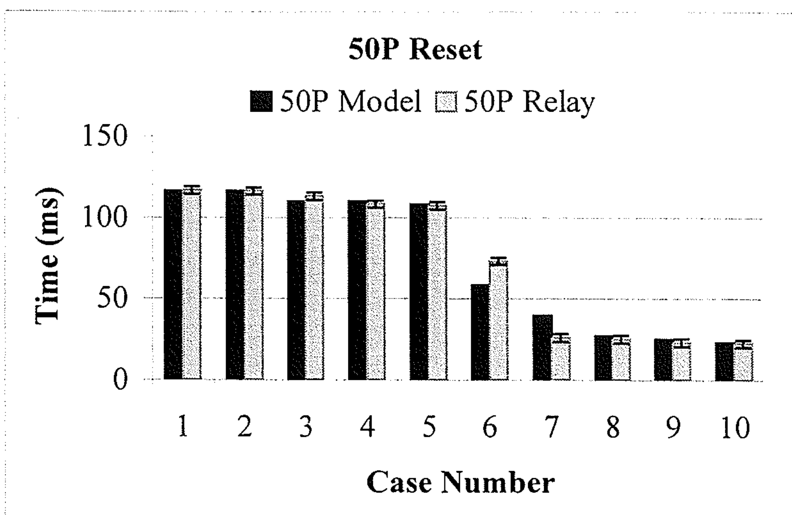
Signal Assignment	Signal Quantity	Signal Units
40	Zone 2 Phase Operate (T11)	1=Trip/0=Reset
41	Zone 2 Phase Operate (T12)	1=Trip/0=Reset
42	Zone 2 A Ground S1-S2 Angle	Degrees
43	Zone 2 B Ground S1-S2 Angle	Degrees
44	Zone 2 C Ground S1-S2 Angle	Degrees
45	Zone 2 Ground Operate (T11)	1=Trip/0=Reset
46	Zone 2 Ground Operate (T12)	1=Trip/0=Reset
47	Phase A measured Z	Ohms secondary
48	Phase B measured Z	Ohms secondary
49	Phase C measured Z	Ohms secondary
50	Phase A measured Z (using Ko)	Ohms secondary
51	Phase B measured Z (using Ko)	Ohms secondary
52	Phase C measured Z (using Ko)	Ohms secondary

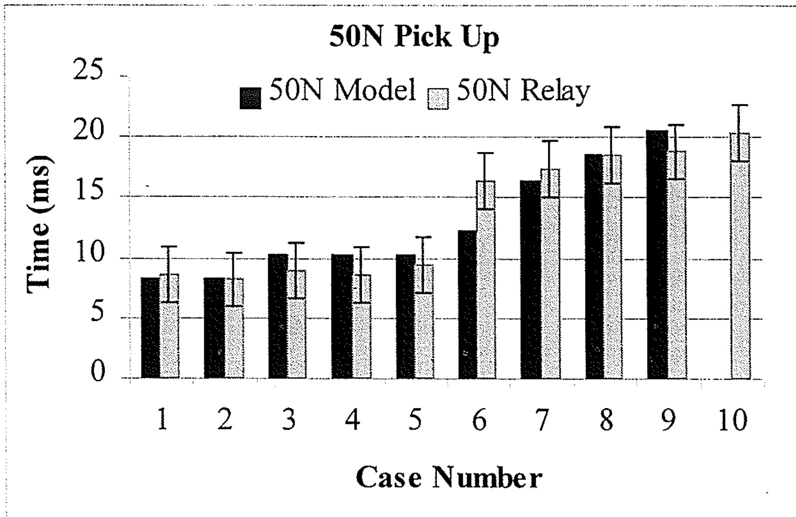
Appendix III – LPRO Model Validation Test Results

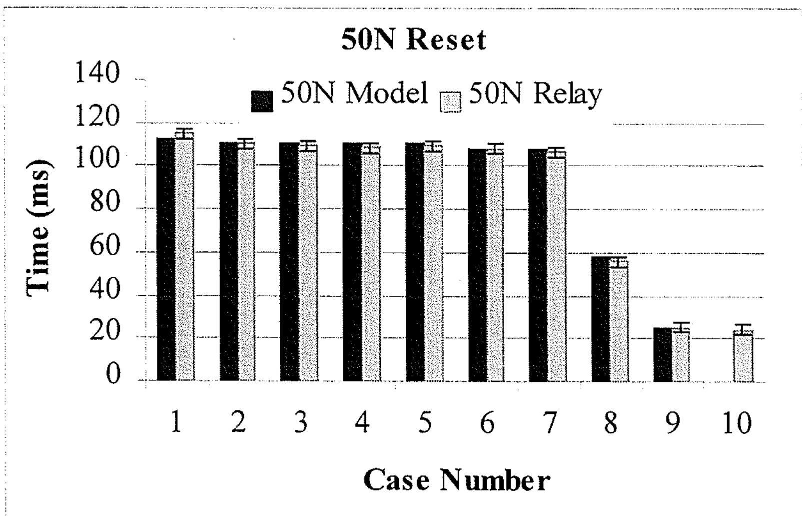
Cases 1-10

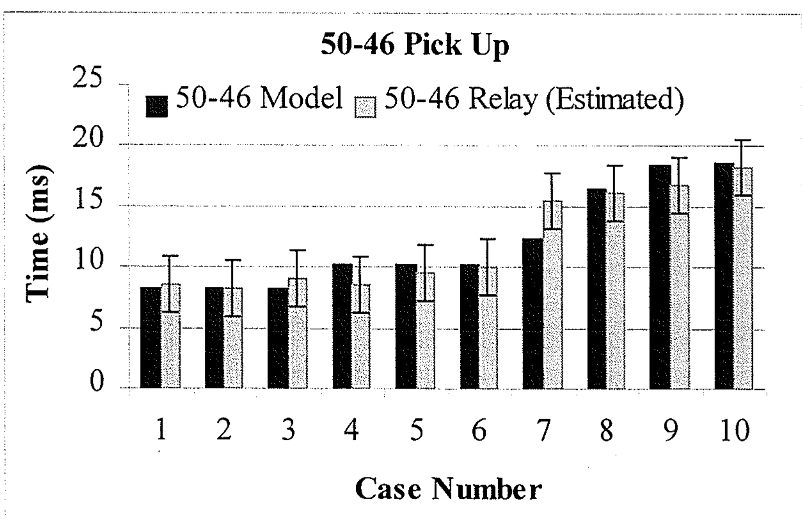
Single-line-to ground faults along the transmission line. Error bars indicate +/- 2.3 ms relative to the operation of the relay element.

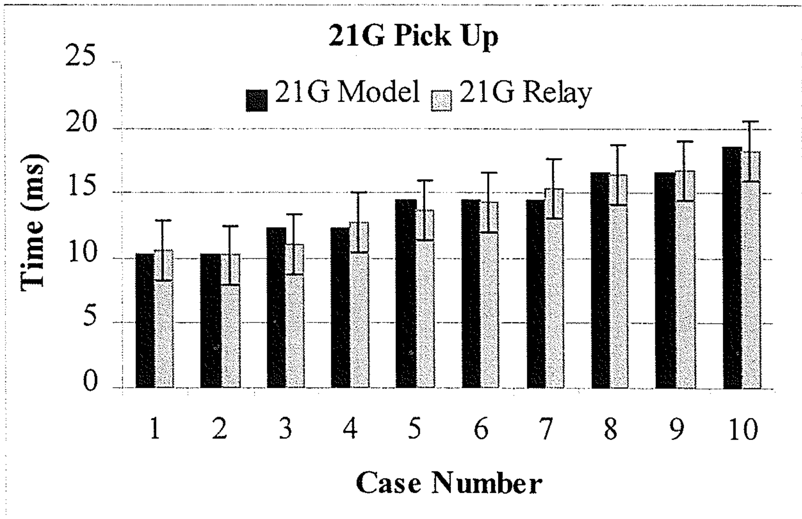


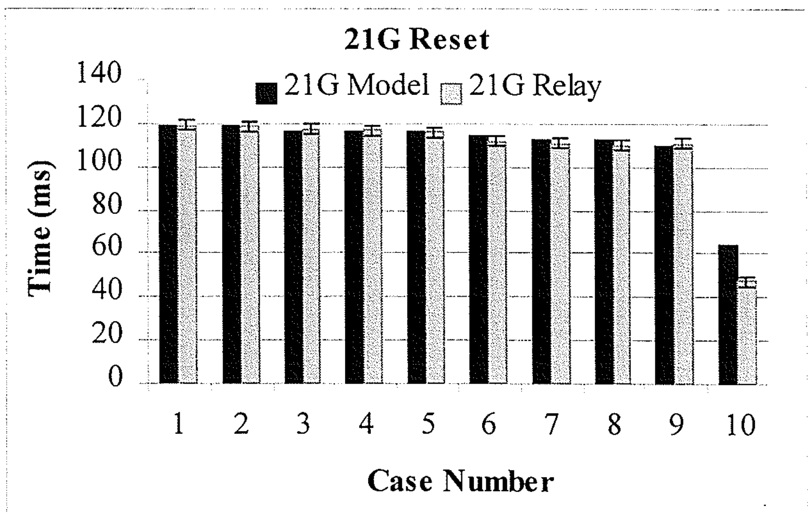



50N Pick Up
50N Model
50N Relay
Time (ms)
25
20
15
10
5
0
1 2 3 4 5 6 7 8 9 10
Case Number


50N Reset
50N Model
50N Relay
Time (ms)
140
120
100
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10
Case Number

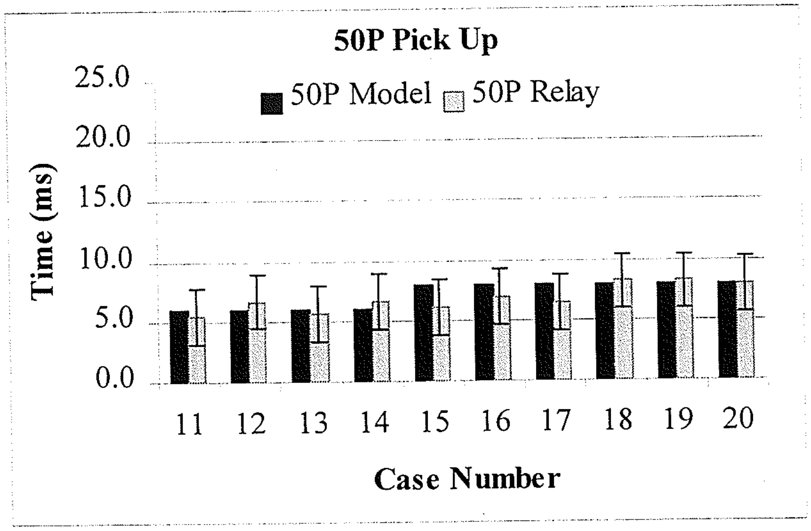

50-46 Pick Up
50-46 Model
50-46 Relay (Estimated)
Time (ms)
25
20
15
10
5
0
1 2 3 4 5 6 7 8 9 10
Case Number

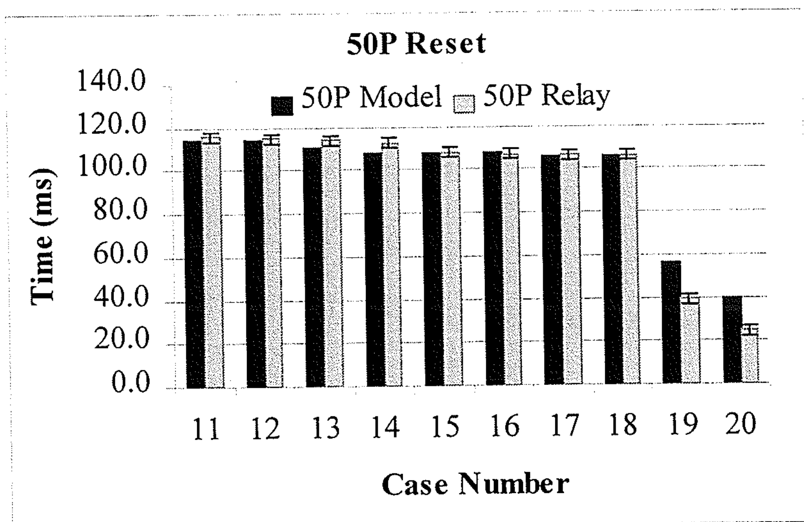

21G Pick Up
21G Model
21G Relay
Time (ms)
25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
Case Number

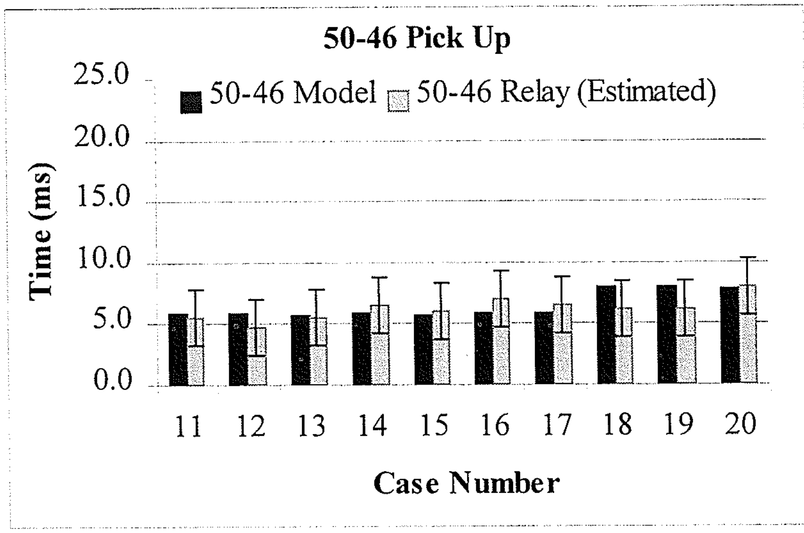

21G Reset
21G Model
21G Relay
Time (ms)
140
120
100
80
60
40
20
0
1
2
3
4
5
6
7
8
9
10
Case Number

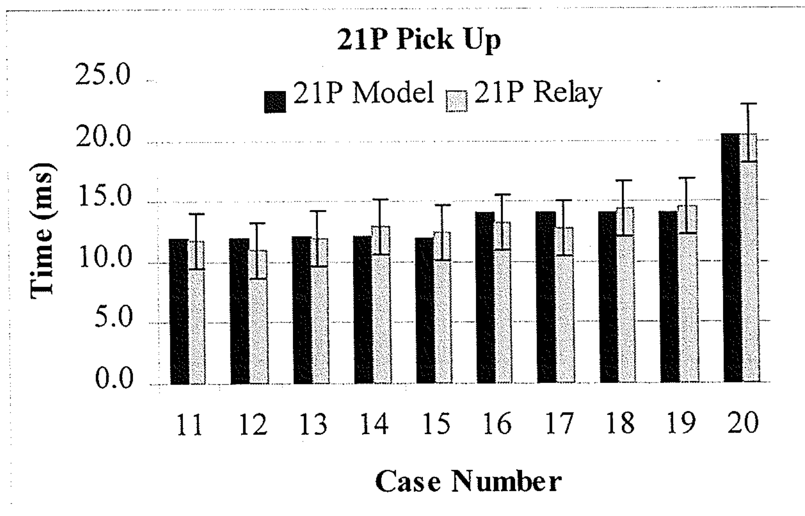
Cases 11-20

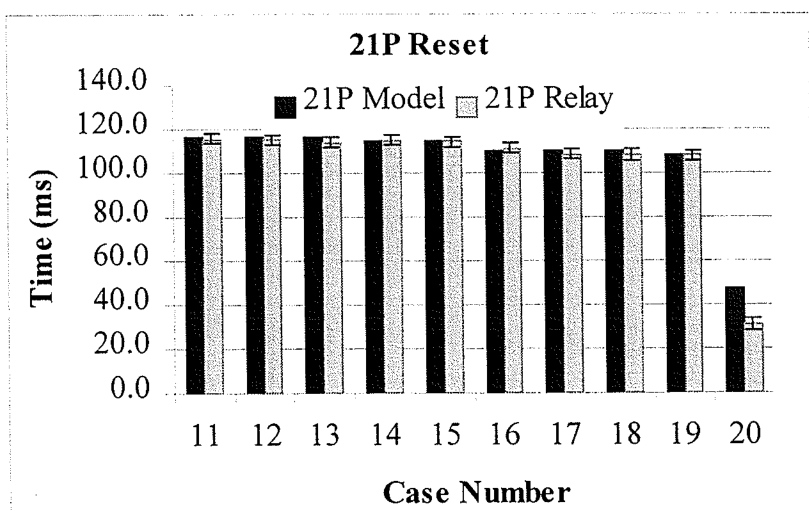
Phase-phase faults along the transmission line. Error bars indicate +/- 2.3 ms relative to the operation of the relay element.





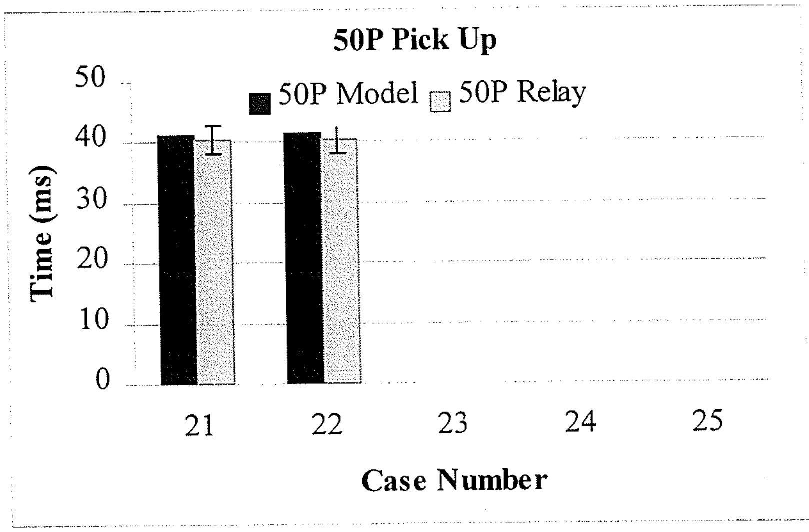

50-46 Pick Up
50-46 Model
50-46 Relay (Estimated)
Time (ms)
25.0
20.0
15.0
10.0
5.0
0.0
11
12
13
14
15
16
17
18
19
20
Case Number

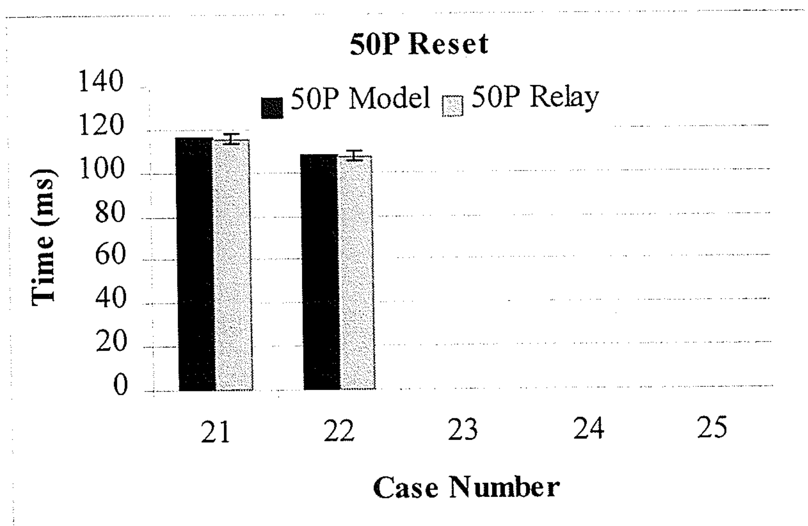

21P Pick Up
21P Model
21P Relay
Time (ms)
25.0
20.0
15.0
10.0
5.0
0.0
11
12
13
14
15
16
17
18
19
20
Case Number

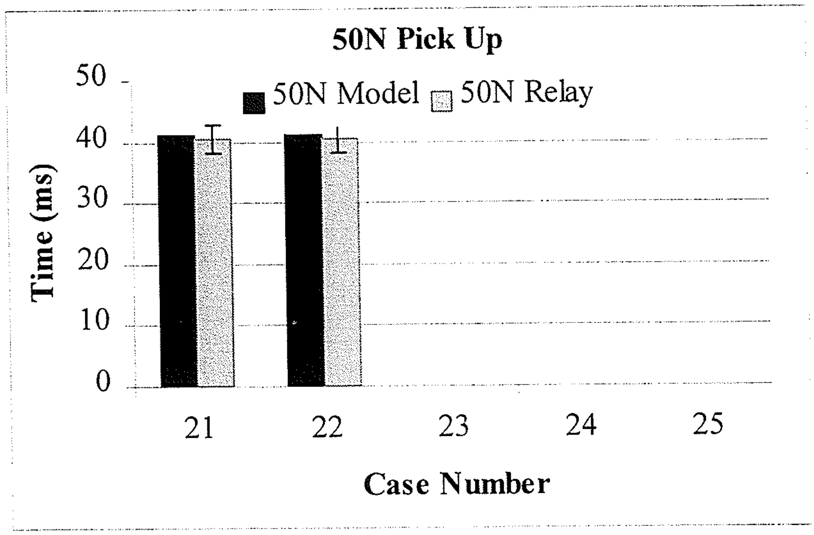

21P Reset
21P Model
21P Relay
Time (ms)
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
11
12
13
14
15
16
17
18
19
20
Case Number

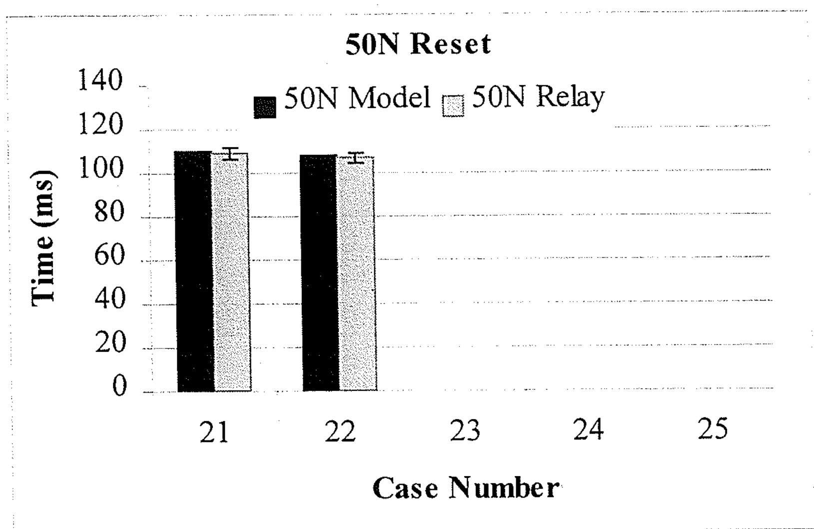
Cases 21-25

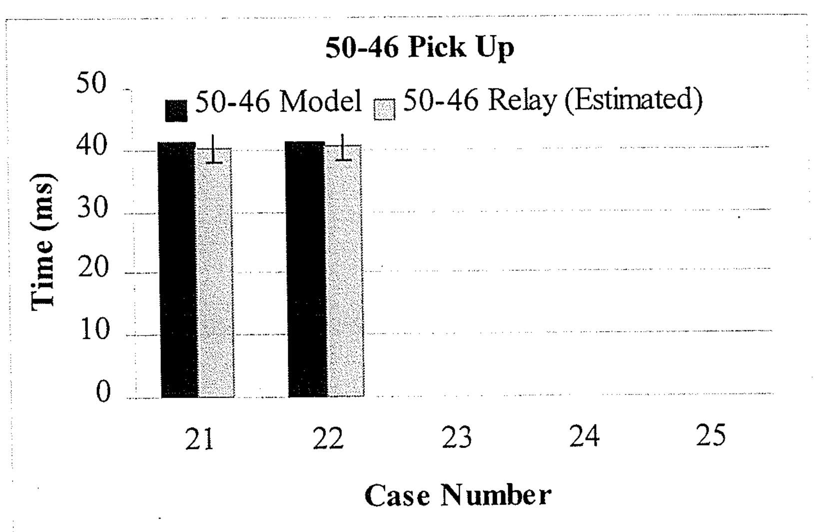
Single-line-to-ground faults at 10% of zone 1 reach with increasing CT saturation. Error bars indicate +/- 2.3 ms relative to the operation of the relay element.

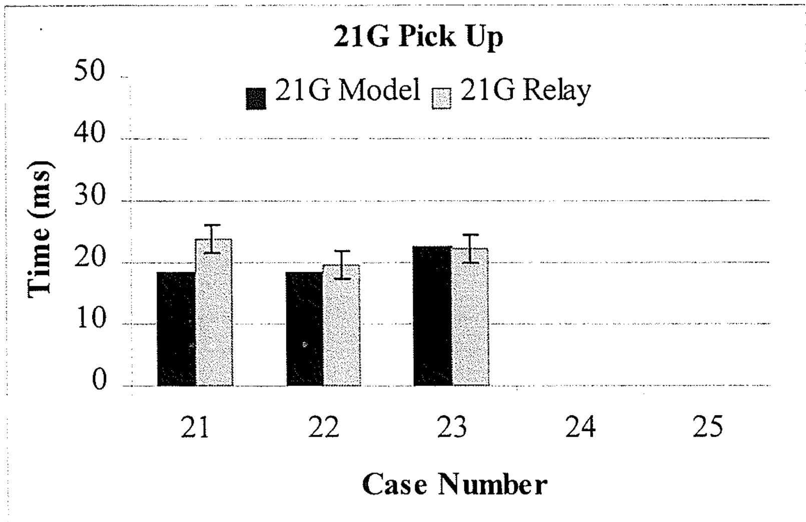


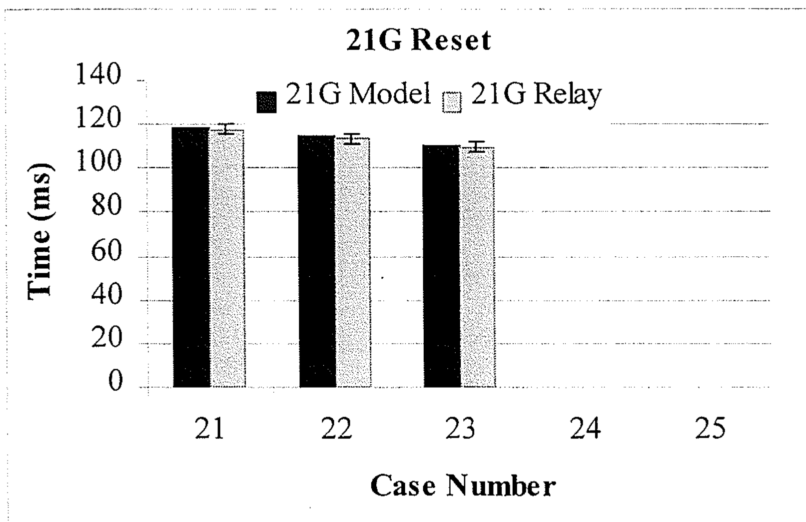



50N Pick Up
50N Model
50N Relay
Time (ms)
50
40
30
20
10
0
21
22
23
24
25
Case Number


50N Reset
50N Model
50N Relay
Time (ms)
140
120
100
80
60
40
20
0
21
22
23
24
25
Case Number


50-46 Pick Up
50-46 Model
50-46 Relay (Estimated)
Time (ms)
50
40
30
20
10
0
21
22
23
24
25
Case Number

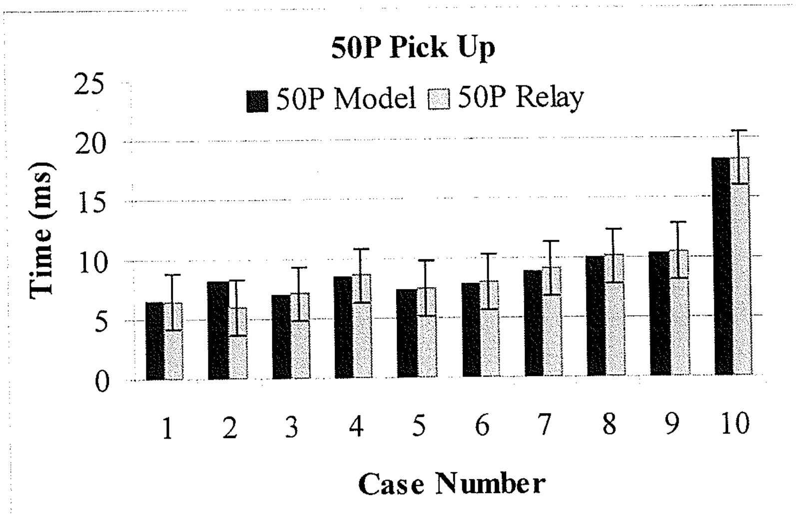

21G Pick Up
21G Model 21G Relay
Time (ms)
50
40
30
20
10
0
21
22
23
24
25
Case Number

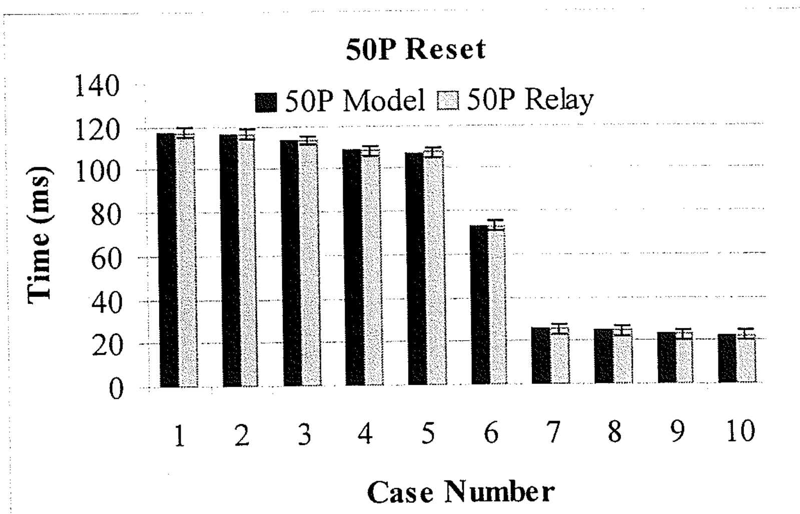

21G Reset
21G Model 21G Relay
Time (ms)
140
120
100
80
60
40
20
0
21
22
23
24
25
Case Number

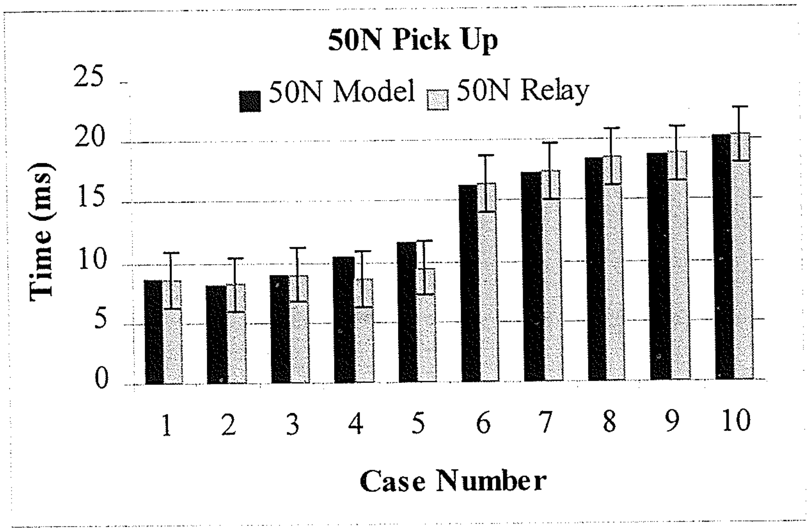
Appendix IV – LPRO Model Validation Test Results Using the Fault Record Playback Method

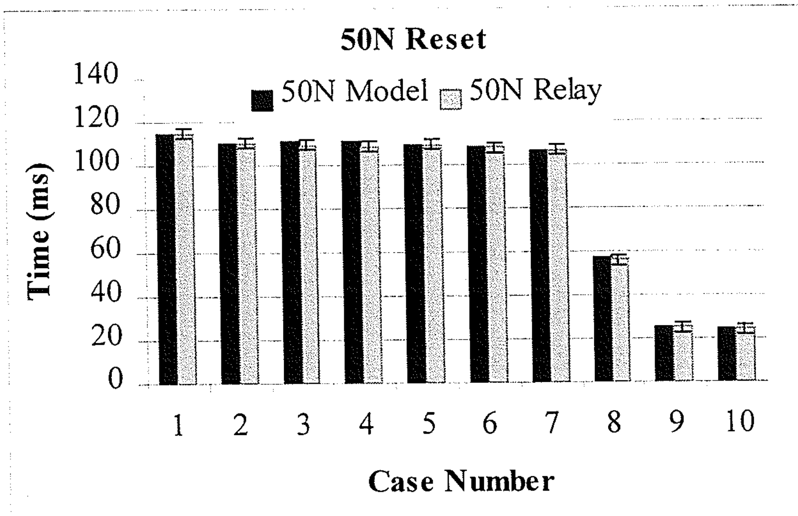
Cases 1-10

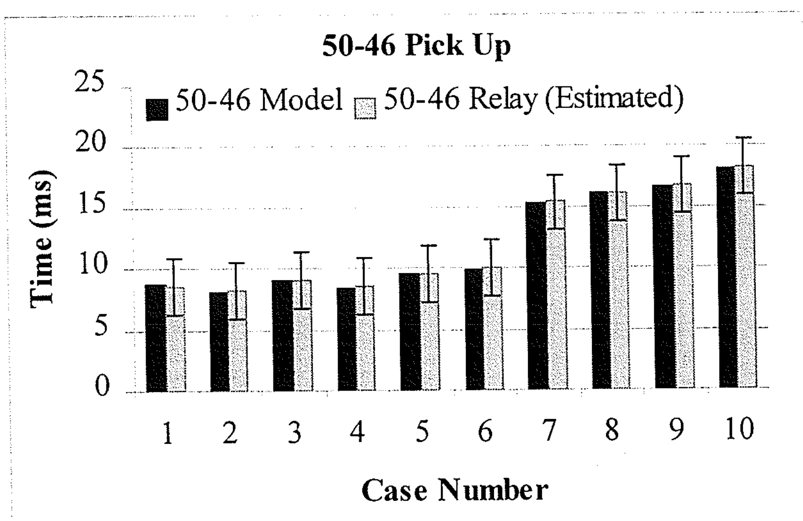
Single-line-to ground faults along the transmission line. Error bars indicate +/- 2.3 ms relative to the operation of the relay element.

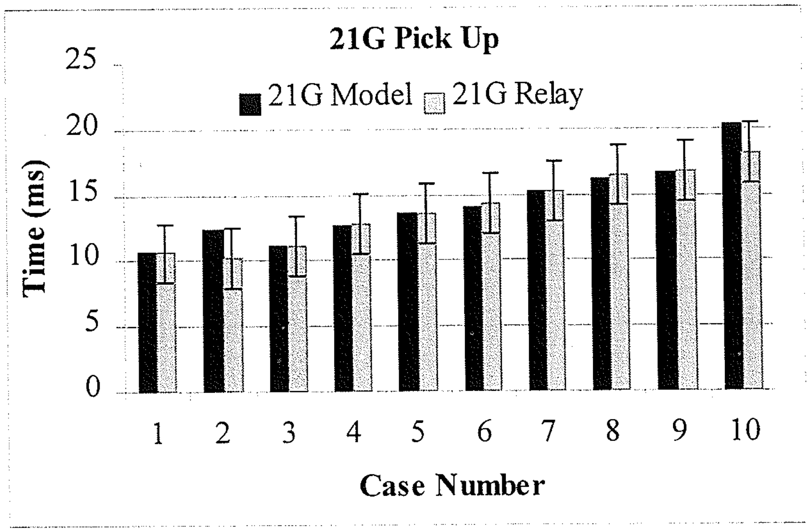


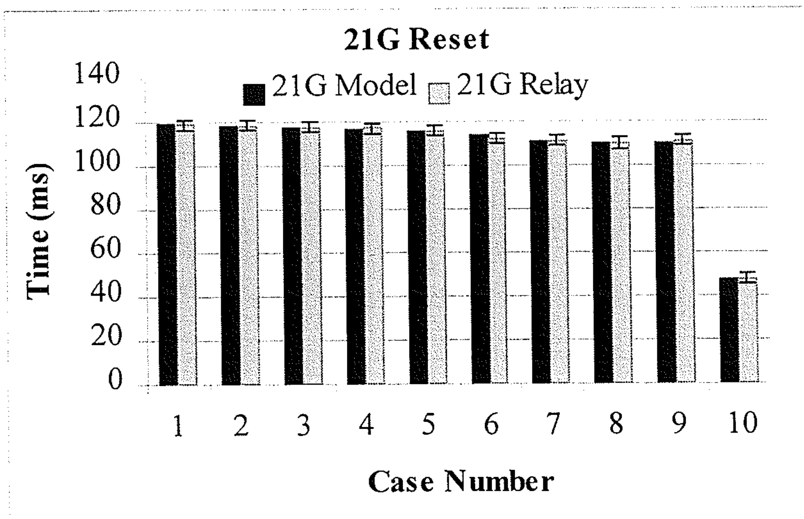



50N Pick Up
50N Model
50N Relay
Time (ms)
25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
Case Number


50N Reset
50N Model
50N Relay
Time (ms)
140
120
100
80
60
40
20
0
1
2
3
4
5
6
7
8
9
10
Case Number

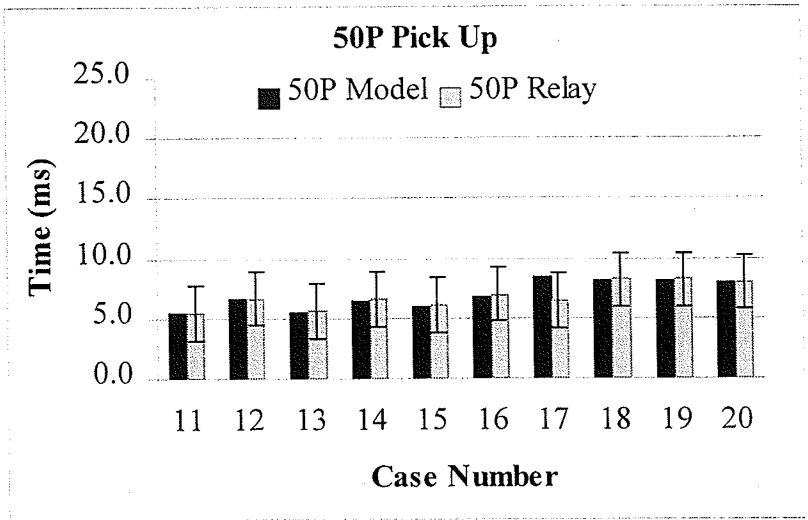

50-46 Pick Up
50-46 Model
50-46 Relay (Estimated)
Time (ms)
25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
Case Number

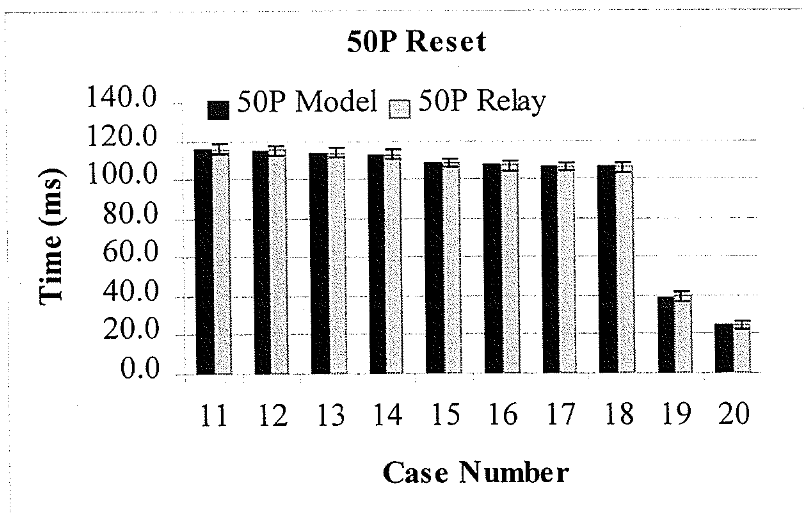

21G Pick Up
21G Model
21G Relay
Time (ms)
25
20
15
10
5
0
1 2 3 4 5 6 7 8 9 10
Case Number

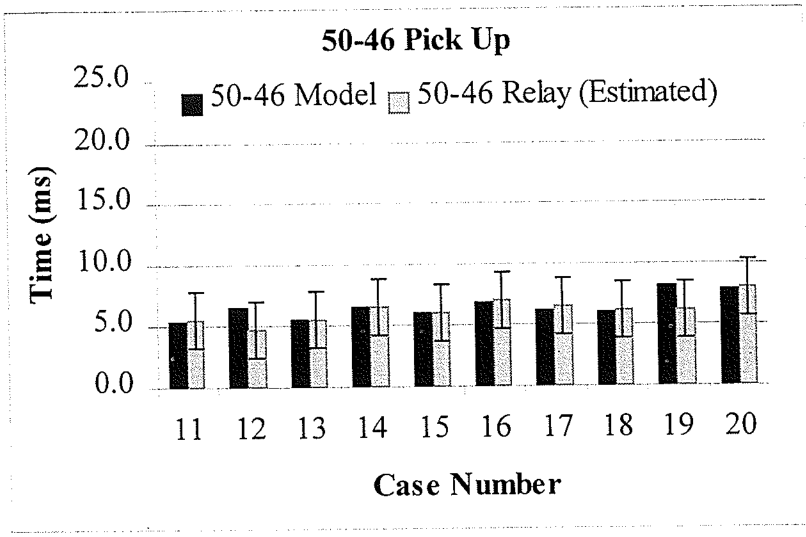

21G Reset
21G Model
21G Relay
Time (ms)
140
120
100
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10
Case Number

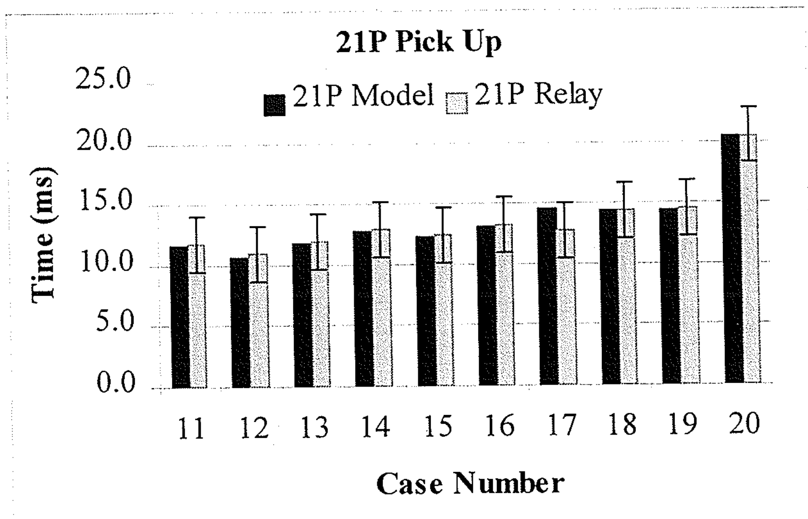
Cases 11-20

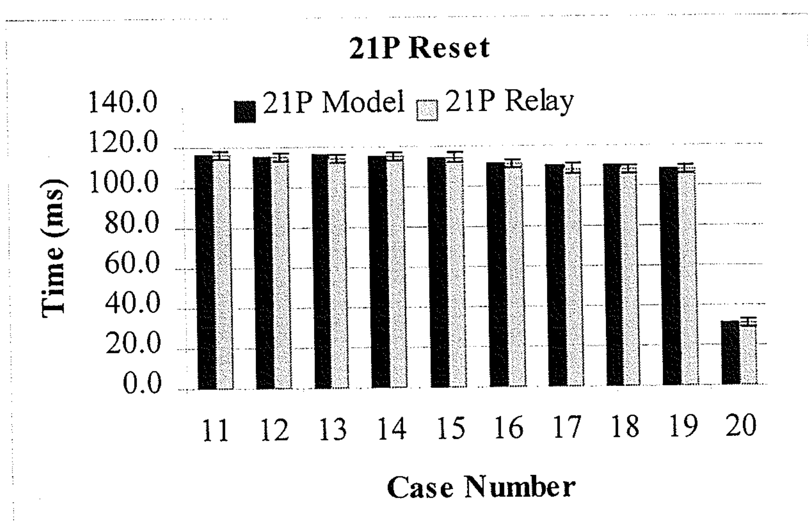
Phase-phase faults along the transmission line. Error bars indicate +/- 2.3 ms relative to the operation of the relay element.





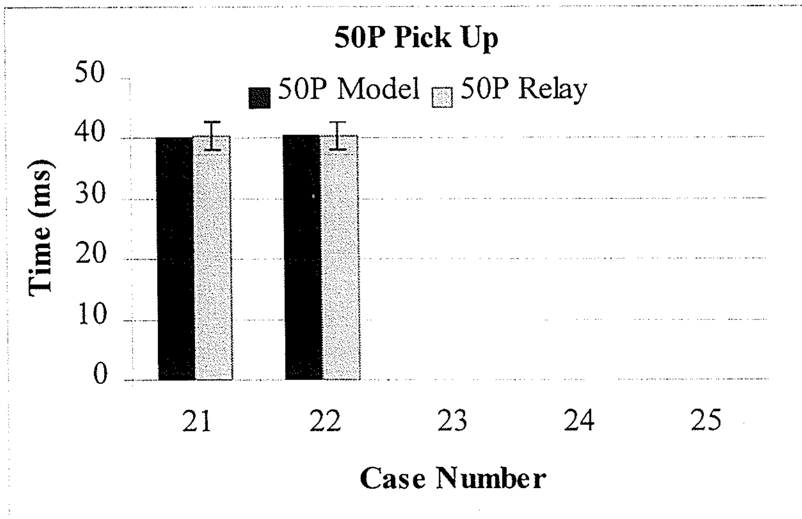

50-46 Pick Up
50-46 Model
50-46 Relay (Estimated)
Time (ms)
25.0
20.0
15.0
10.0
5.0
0.0
11 12 13 14 15 16 17 18 19 20
Case Number

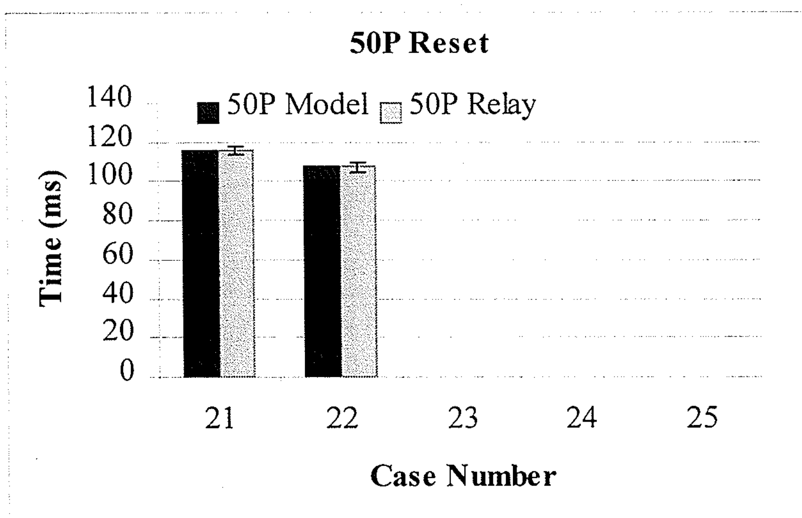

21P Pick Up
21P Model
21P Relay
Time (ms)
25.0
20.0
15.0
10.0
5.0
0.0
11 12 13 14 15 16 17 18 19 20
Case Number

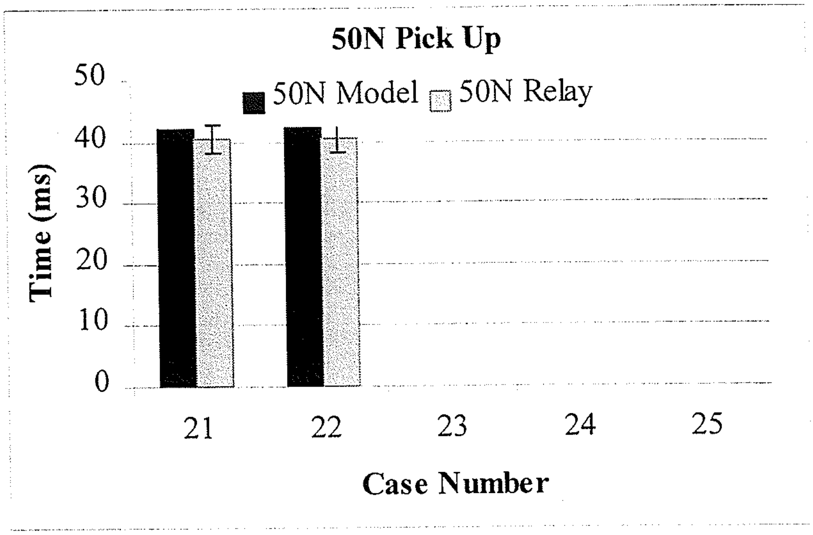

21P Reset
21P Model
21P Relay
Time (ms)
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
11 12 13 14 15 16 17 18 19 20
Case Number

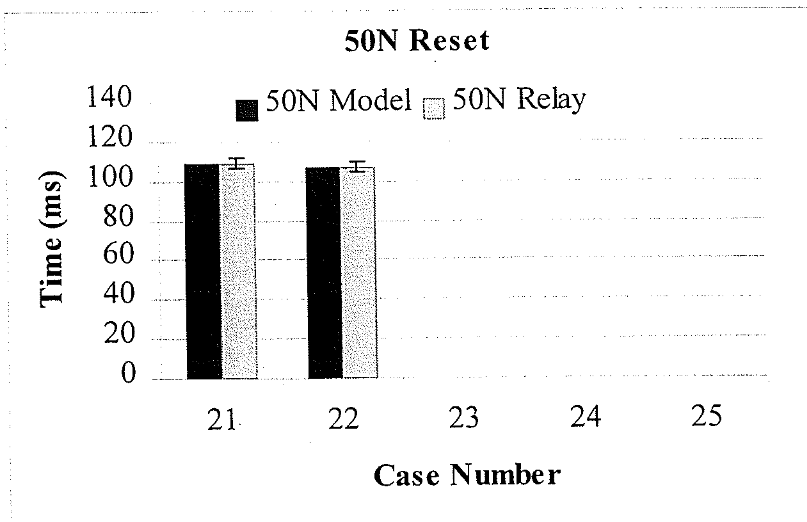
Cases 21-25

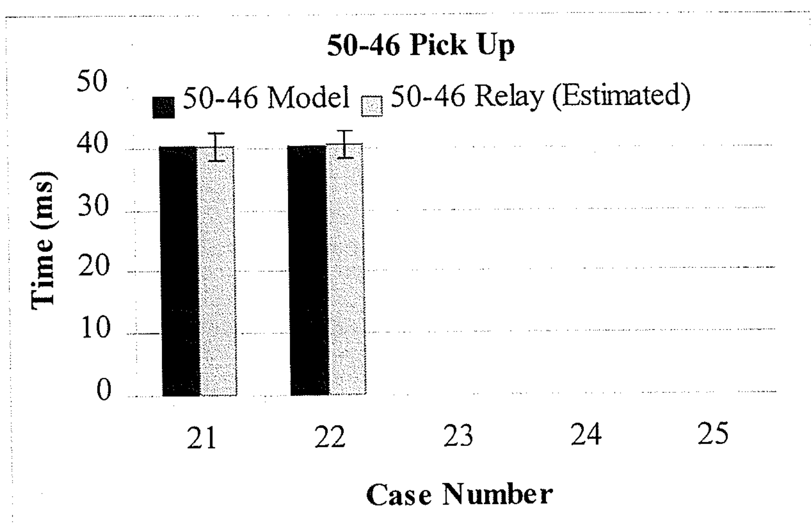
Single-line-to-ground faults at 10% of zone 1 reach with increasing CT saturation. Error bars indicate +/- 2.3 ms relative to the operation of the relay element.

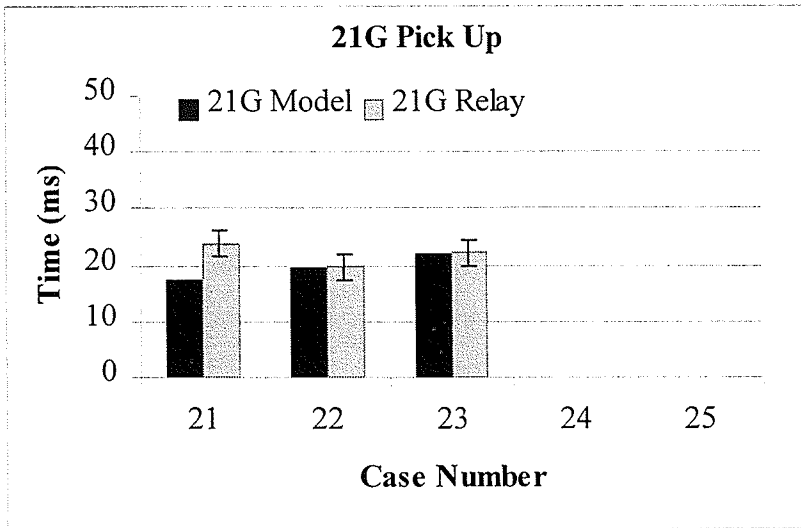


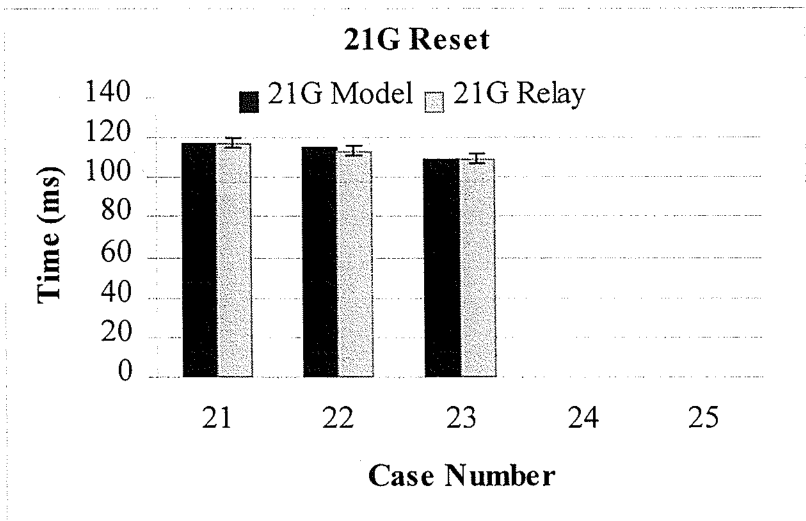



50N Pick Up
50N Model
50N Relay
Time (ms)
50
40
30
20
10
0
21
22
23
24
25
Case Number


50N Reset
50N Model
50N Relay
Time (ms)
140
120
100
80
60
40
20
0
21
22
23
24
25
Case Number


50-46 Pick Up
50-46 Model
50-46 Relay (Estimated)
Time (ms)
50
40
30
20
10
0
21
22
23
24
25
Case Number


21G Pick Up
21G Model
21G Relay
Time (ms)
50
40
30
20
10
0
21
22
23
24
25
Case Number


21G Reset
21G Model
21G Relay
Time (ms)
140
120
100
80
60
40
20
0
21
22
23
24
25
Case Number