MICROPROCESSOR PROTECTION
IN POWER DISTRIBUTION SYSTEMS

by

John Charles Zeigler
Electrical Engineering

Submitted in Partial Fulfillment of the Requirements of the
University Undergraduate Fellows Program

1978-1979

Approved by:

B Don Russell

B. Don Russell

April 1979

ABSTRACT

Microprocessor Protection
in Power Distribution Systems
(April 1979)

John Charles Zeigler, Texas A&M University
Advisor: Dr. B. Don Russell

The rapid growth in size and complexity of modern power distribution systems has created a corresponding need for advancements in the field of system protection. A device incorporating the inherent intelligence of microprocessor has been developed to replace the conventional electromechanical overcurrent relay in common use at the substation level. The intelligent relay exhibits the flexibility required to cope with frequently changing system configurations. In addition, secondary control and monitoring functions have been implemented to increase system efficiency at minimal cost. The ability to communicate directly with an intelligent relay of this type will also enhance the effectiveness of large scale control systems, whether operated manually or by a centralized computer system.

ACKNOWLEDGEMENTS

I would like to acknowledge my advisor, Dr. B. Don Russell, for his help and guidance throughout the year. Acknowledgements should also go to Ernie McWilliams and Page Heller, graduate students in charge of hardware and software development, respectively. In addition, Dow Chemical Company should be acknowledged for their support of the project.

TABLE OF CONTENTS

TABLE OF CONTENTS

(continued)

LIST OF FIGURES

INTRODUCTION

Protection Philosophy

The dramatic strides in industrial sophistication and complexity in recent years have placed increasingly stringent demands on the suppliers of electrical power. These requirements reflect the growing number and variety of industrial processes which can function only in the presence of a constant, uninterrupted source of energy. In an effort to cope with these escalating demands, the modern power system has evolved into an intricate network of distribution subsystems requiring exceedingly complex methods of protection.

There are a variety of approaches to power system protection, but each method must necessarily share several common considerations. In addition to the primary protection devices themselves, provisions must be made for adequate control and measurement functions at strategic points throughout the system. All three of these functions must then be supported by some form of communication network to allow proper operation of the system as a whole.

In the past, the equipment necessary to realize each of these functions was developed or redesigned only when a new requirement arose. Thus, the protection systems have evolved in a rather haphazard fashion, with devices based on radically different design philosophies operating side-by-side. In general, each device was totally dedicated to the performance of a single task, and was ill-suited or even useless for application in a different system or environment[1]. In recent years, however, a new design philosophy has emerged, demanding increased uniformity, capability, and flexibility from the myriad of protection devices currently in use[2]. Protection engineers have realized the need for a line of standard devices which can perform specific functions regardless of the actual system or environment in which they are to be placed. It has also become desirable to include more than one function in the operational capabilities of a single unit in order to reduce the total number of devices required. Additionally, these devices should be flexible enough to cope with the continually changing configuration of a dynamic distribution system.

Computer Applications

This recent change in protection philosophy has been prompted by rapid advancements in the areas of computer technology and applications. As computers began to gain speed and computational capabilities, various methods were proposed to allow a single, centralized computer facility to handle the protection, control, and monitoring functions for the entire system [3,4]. This approach would allow maximum accuracy since the central facility would have access to a tremendous amount of data collected from vital locations throughout the system. The complete reliance on software data manipulation and decision making would also allow maximum adaptability to changes in system configuration. Nevertheless, the volume of mathematical computations required and the speed with which they must be performed have prevented the implementation of a totally

The journal IEEE Transactions on Power Apparatus & Systems is used as a pattern for format and style.

integrated computer facility to handle all the basic functions for a large power system [5]. A further disadvantage would be the necessity for an extensive and complex communications network to allow interaction with each strategic point in the entire system. In addition, the inherent unreliability of a single, centralized facility would require the installation of multiple parallel computers to provide acceptable levels of availability. Thus, the cost of a totally integrated system would preclude its implementation in power networks small enough to fall within the computational capabilities of a single computer facility.

Advancements in the field of minicomputers, however, offered an alternative to the single, centralized mainframe computer facility. This approach involved the installation of a minicomputer in each substation to provide all the protection, control, and monitoring functions required by that portion of the system [6]. The data processed by each substation facility could then be communicated to a central computer providing supervisory control over the entire power system. This method would greatly increase the system's reliability and availability since each minicomputer would provide autonomous operation in the event of failure of the central computer. Although overall system flexibility would be diminished, the software orientation of individual substations would still permit adequate adaptation to changes within their respective realms of operation. The decreased volume of data to be evaluated by each facility would greatly improve the response time to system disturbances regardless of their location. The reduced capabilities and slower speed of a minicomputer, however, have prevented the development of a cost-effective segregated system which incorporates all phases of protection, control, and monitoring functions at a substation level [7].

In an effort to upgrade conventional practices, however, several power systems have incorporated either integrated or segregated computer facilities to provide two of the basic functions, control and monitoring, for the entire system [8,9]. In spite of their differing approaches, a feature common to each of these applications is the implementation of the protection function through the use of distributed electromechanical devices. An alternative to this hybrid approach has recently become available, however, with the advent of low cost microprocessors. This new technology would permit the replacement of several dedicated mechanical devices with a single software oriented device to provide all the protection, control, and monitoring functions required at that point [10]. In addition to a reduction in the total number of devices required, the system's flexibility would be greatly enhanced due to the microprocessors ability to adapt to a dynamic system configuration. A communications network linking individual microprocessors and substation minicomputers would further expand system flexibility and permit more efficient operation of the overall system. The required levels of reliability and availability could be achieved at minimal cost by incorporating redundancy at the individual chip level. Although the microprocessor's speed and computational capabilities cannot compare with the larger computers, the small amount of data to be processed would permit more rapid response to disturbances on that portion of the system. Nevertheless, the reliance on such a small data base cannot provide the accuracy inherent in the more integrated approaches. The necessity for more hardware also inhibits the flexibility of the distributed approach when compared to methods relying more fully on software manipulations. In spite of these disadvantages, however, the distributed approach seems to be the most cost-effective method of incorporating protection as well as control and monitoring functions into a single device.

Problem Definition

As previously mentioned, several power systems are presently utilizing some form of computer based network to provide two basic functions at strategic points throughout the system. Thus, any effort to incorporate all three basic functions of protection, control, and monitoring should focus its efforts on the one function excluded by these other systems: primary protection of power distribution systems.

The purpose of a protective device is to determine the nature and location of faults on the system as soon as possible so that corrective action can be taken to minimize the disturbance to consumers [11]. Thus, the device must interrupt fault currents rapidly in order to prevent damage to the line in the vicinity of the fault and to prevent the activation of protective devices in portions of the system otherwise unaffected by the fault. Although a wide variety of protective devices are presently in use, C. Russell Mason states that "Overcurrent relaying is the simplest and cheapest, the most difficult to apply, and the quickest to need readjustment or even replacement as a system changes" [11].

The conventional electromagnetic overcurrent relay, shown in Figure 1, utilizes an alternating current obtained from an instrumentation type current transformer to produce a time-varying electromagnet field [12]. This field induces currents in a printed disk (see Figure 2) which interact with the flux from the magnetic field to produce a torque. Thus, for any fault current above a minimum pickup value the disk will be forced to rotate against the damping effect of the restraining spring. If the fault current persists, the disk will rotate far enough to complete a

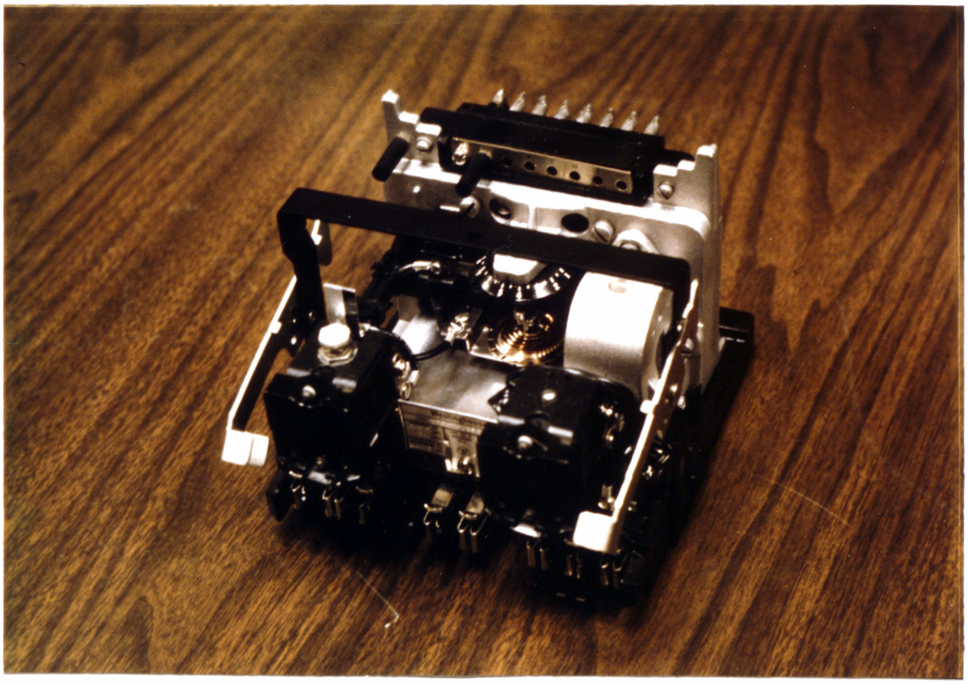


Figure 1. Electromagnetic Overcurrent Relay

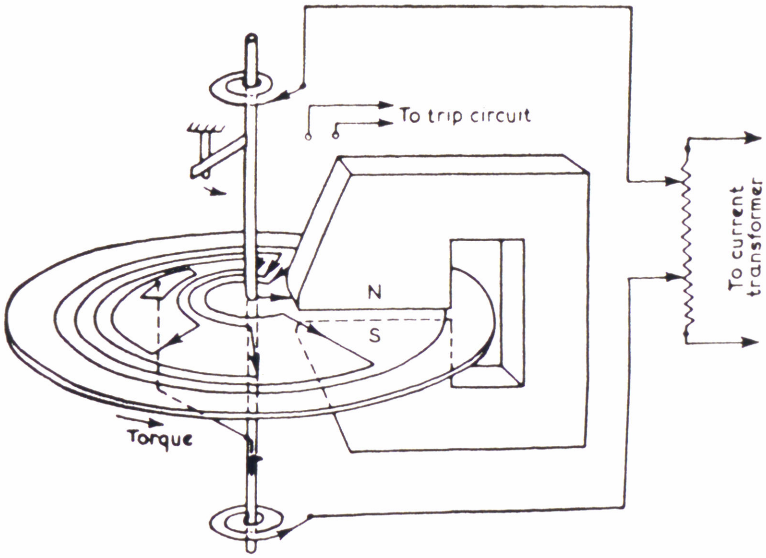


Figure 2. Schematic for Electromagnetic Relay

circuit which sends a trip signal to the corresponding breaker. This time-current characteristic for a typical overcurrent relay is shown in Figure 3. This distinctive shape, $I^n t$ = constant, is produced by the inverse characteristic of the restraining spring. A time lever is provided to allow the curve to be shifted vertically in time. In addition, the top setting of the relay is utilized to shift the curves horizontally in terms of current magnitude.

Although the electromechanical overcurrent relay has proven to be highly reliable in practice, several drawbacks have encouraged the search for an economical alternative. The basic principle of operation severely limits the type of time-current characteristics which can be realized. The extreme limits for these devices range from a fairly flat time-current curve (CO-6: definite minimum time) to a curve which is inversely proportional to the square of the current (CO-11: extreme inverse) [12]. This range is quite adequate in many applications, but increasing system complexities are requiring more flexible time-current characteristics. The problem, known as coordination, is exemplified by Figure 4. In (a), the CO-8 time-current characteristic is seen to fall between the fuse curves and the area corresponding to physical damage to the line. This is the desired situation, since the relay allows sufficient time for any fuses to blow which will sectionalize the system and correct the disturbance. If the fault is on a portion of the system unprotected by fuses, however, the relay will trip before any damage is done to the line. The contrasting situation in (b) is undesirable since the overcurrent relay will trip the breaker in certain cases before the line fuses have had time to blow. This results in deenergizing a larger portion of the system than is necessary to alleviate the fault condition.

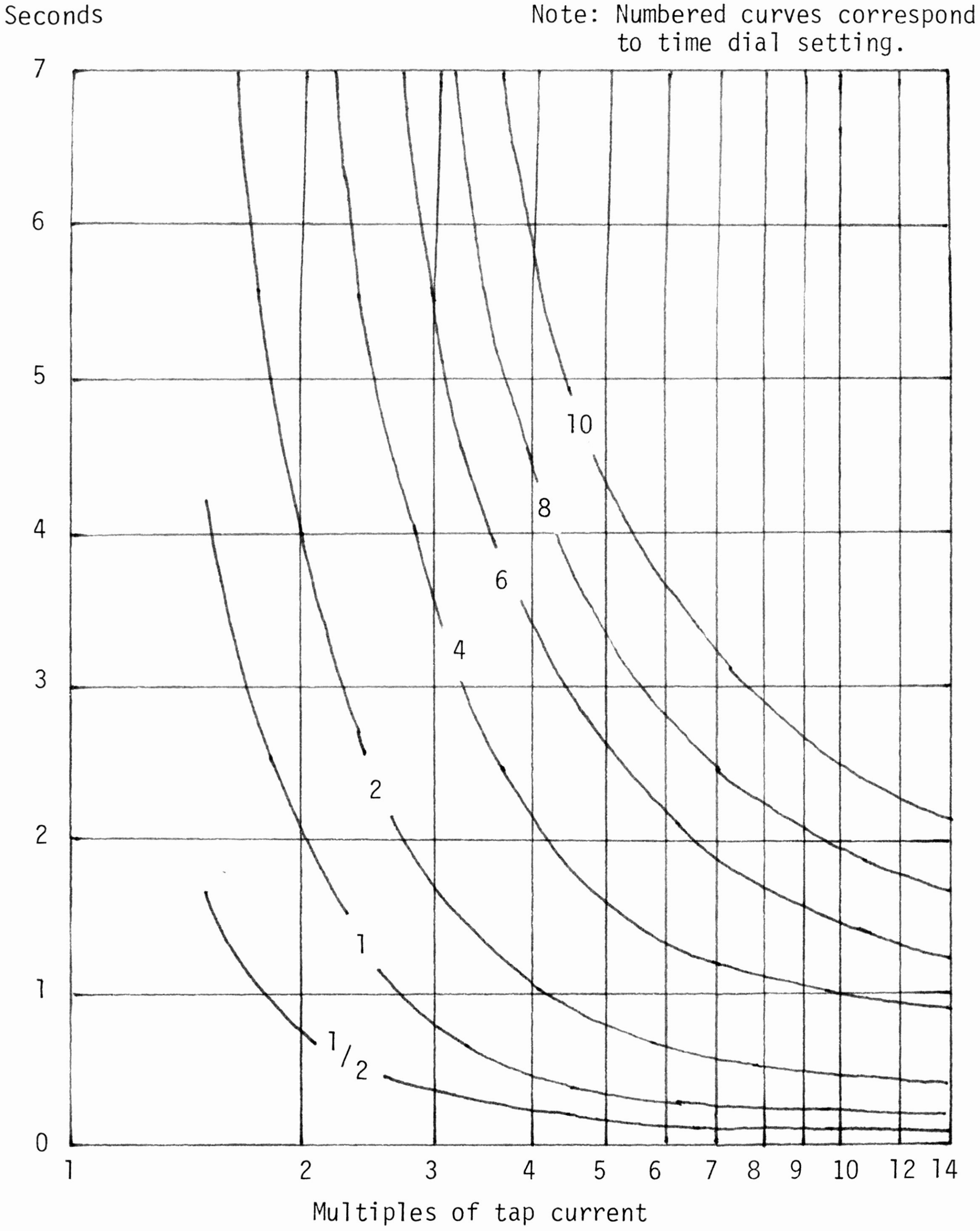
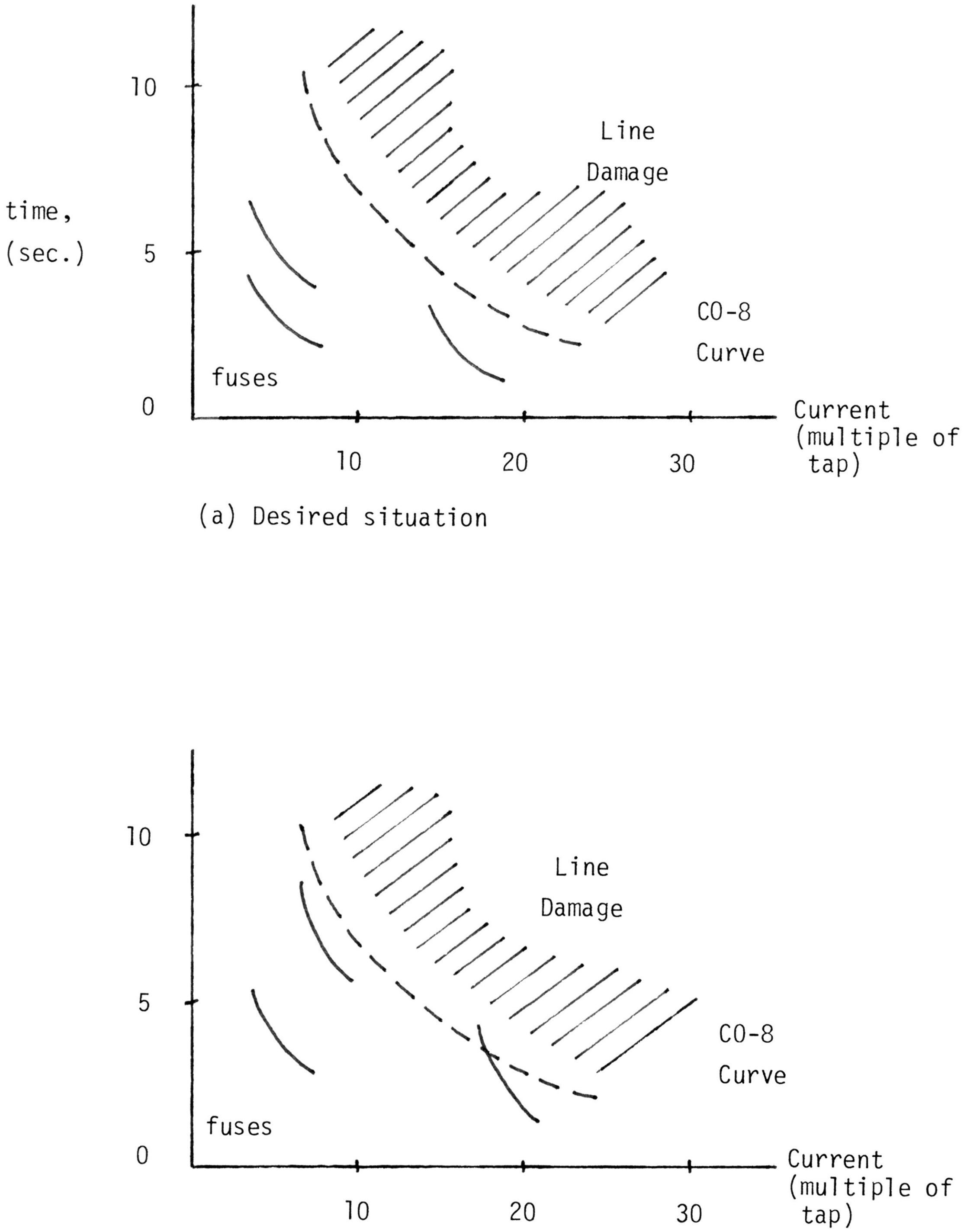


Figure 3. Time-Current Characteristic



(a) Desired situation

(b) Undesired situation

Figure 4. Coordination of Relays

A second drawback of electromechanical relays is their tendency to continue rotating after the fault current has been removed. This characteristic, known as overreach, is caused by the inertia of the disk and is of particular significance for inverse-time overcurrent relays since their operation is based on a time-delay characteristic. Thus, improper breaker action may result in situations where the fault current is interrupted several cycles prior to the desired trip time.

The reset time for the electromechanical relay is primarily dependent on the physical characteristics of the damping spring, since it supplies the force necessary to rotate the disc back to its original position after the disturbance has been corrected [13]. Since the damping spring also determines the time-current characteristics, the reset time is proportional to the "inverseness" of the relay curve. As this curve approaches the CO-11, or extreme inverse time characteristic, the reset time may become too long to allow proper timing of the breaker reclosing.

These drawbacks have prompted engineers to consider alternatives to the conventional electromechanical overcurrent relay. Various forms of solid-state devices have been proposed which achieve greater than I^2t = constant time-current characteristics [14,15]. These devices, however, lack the flexibility and intelligence inherent in a microprocessor based system. This approach would alleviate the problems of the electromechanical relay since it could provide any desired time-current characteristic, would be immune to overreach, and would provide almost instantaneous reset times [16]. The reliance on software analysis would facilitate adaptation to changing system configurations while providing optimum protection at all times. The capability for communication between devices would increase the speed and accuracy of responses to system disturbances. In addition, the microprocessor's primary role as a protection device would be augmented by its ability to incorporate any desired control and monitoring functions into a single device.

SELECTION OF ALGORITH

Overview

The initial application of microprocessors to provide overcurrent protection in power systems requires that they be compatible with the electromechanical relays performing various other functions. The electromechanical relays, however, base all their measurements on the rms or phasor value of the input signal, while the microprocessor is inherently responsive to the instantaneous value of the current or voltage. Thus, the microprocessor must rely on a new principle of fault detection and location or on some method of converting the instantaneous values to their corresponding rms magnitude [7]. The most significant work done in this area has been directed toward the latter approach, and has resulted in the development of the algorithms outlined below.

Fourier/Walsh Transforms

The principle underlying these two approaches is that any signal can be decomposed into a pair of orthogonal base functions and an infinite sum of the harmonics of these base functions. The major difference between the two methods is the form of the base functions: Fourier analysis utilizes sine and cosine waves, while Walsh analysis incorporates a pair of orthogonal square waves. The first step in either analysis is to correlate the base functions with the input signal to determine the magnitude of the fundamental 60 Hz signal. Then, for a Fourier analysis [17]:

$$g(t) = F_0 + \sqrt{2}\, F_1 \sin \frac{2\pi t}{T} + \sqrt{2}\, F_2 \cos \frac{2\pi t}{T} + \sqrt{2}\, F_3 \sin \frac{2\pi t}{T} + \ . \ . \ .$$

where

$$F_o = \frac{1}{T} \int_o^T g(t)\, dt$$

$$F_1 = \frac{1}{T} \int_o^T g(t) \sin \frac{2\pi t}{T}\, dt$$

$$F_2 = \frac{1}{T} \int_o^T g(t) \cos \frac{2\pi t}{T}\, dt.$$

For a Walsh analysis, the corresponding equation is [17]:

$$g(t) = \sum_{K=0}^{\infty} W_k \text{ wal } [K, \frac{t}{T}]$$

where

$$W_K = \frac{1}{T} \int_o^T g(t) \text{ wal } [K, \frac{t}{T}]\, dt$$

wal $[K, \frac{t}{T}]$ is a square wave having k transitions between +1 and -1 in a period T.

Since each method extracts the fundamental 60 Hz wave, they act as a very narrow bandpass filter and result in a slow, heavily damped but highly accurate representation of a distorted signal. Although these equations have been implemented in minicomputers for distance relaying at the transmission level [17], the volume of computations required preclude their implementation in a microprocessor, since it lacks the speed and capabilities of a minicomputer.

Derivatives

A method to determine the peak of a sinusoidal wave has been developed which relies on the measurement of a single sample and its derivative. This has the form [18]:

$$I_{peak} = i_K^2 + \left(\frac{diK}{dt} \right)^2$$

which may be approximated by three sample points in the form

$$I_{peak} = i_K^2 + [\frac{1}{hw} (i_{K+1} - i_{K-1})]^2$$

where

h = sampling interval in seconds

w = angular frequency of signal = 377 radians/second.

For a high sample rate, this equation allows a very fast decision due to the simplicity of the calculations involved. Nevertheless, the d.c. component must be eliminated either in hardware or software to prevent erroneous results. A similar equation which automatically eliminates the d.c. component can be defined in terms of the first and second derivatives. Thus, the peak value is given by [19]:

$$I_{peak} = (i'_K)^2 + (i''_K)^2$$

where

$$i'_K = \frac{1}{hw}(i_{K+1} - i_{K-1})$$

$$i''_K = (\frac{1}{hw})^2 (i_{K+1} - 2i_K + i_{K-1}).$$

The presence of high frequency components, however, have a more pronounced effect on the second derivative and would have to be eliminated by filtering to produce accurate results.

Curve-Fitting

All curve-fitting techniques assume a basic curve shape and attempt to evaluate a set of constants which will correspond to the input waveform. The least-squares approach assumes the general shape [20]:

$$I(t) = K_1 e^{-\lambda t} + \sum_{m=1}^{n} (K_{2m} \sin (mwt) + K_{2m+1} \cos (mwt))$$

where

$$K_r = \int_0^t I(t) f_r(t)\, dt \quad \text{for} \quad r = 1, 2, \ldots, 2n + 1$$

The number of computations required for the least-squares evaluation of the coefficients, however, would prohibit the implementation of this approach with a single microprocessor. But, if the waveform is assumed to be a sinusoid, a curve-fit may be employed of the form [21]:

$$I^2_{peak} = \frac{i^2_{n-1} - i_{n-2}\, i_n}{\sin^2\Delta}$$

where

Δ = angle in radians between samples.

If the sampling rate is held constant, this reduces to the form:

$$X = I^2_{peak} \sin^2\Delta = i^2_{n-1} - i_n i_{n-2}.$$

This form allows a rapid response at high sampling rates, but is adversely affected by d.c. and harmonic components. Nevertheless, this method requires only a minimal amount of computational ability, since it involves only two multiplications and one subtraction. In comparison, the sample and derivative method requires three multiplications, one subtraction, and one addition. Thus, the simplicity of the calculations render the sinusoidal curve-fit ideal for implementation in a microprocessor based system.

SOFTWARE PACKAGE

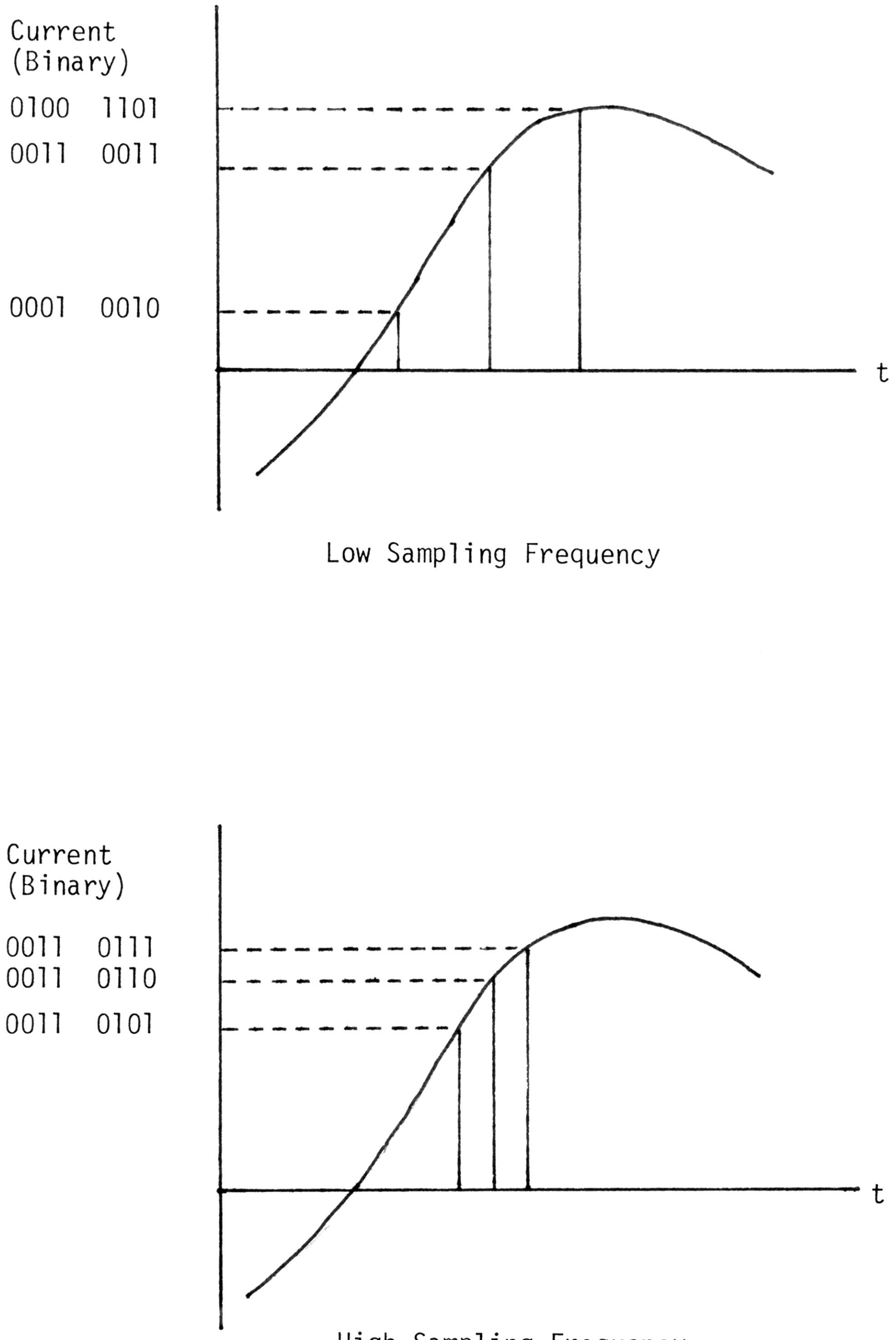
Overview

The selected curve-fitting equation must now be incorporated into a general routine which provides certain prescribed functions for each possible system condition. These functions may be implemented as four major routines: (1) no fault, (2) instantaneous trip, (3) timed trip, and (4) fault recovery. Obviously, the trip routines have the highest priority, but the no-fault loop is the normal mode of operation. Thus, this is the routine which must determine when a fault exists on the system and immediately transfer control to the appropriate trip routine. In addition to the primary function of fault detection and correction, the general routine should also include provisions for system diagnostics, warning flags, data logging, and any other desired secondary functions.

Sampling Rate & Error Analysis

The initial consideration for implementation of the general routine is the sampling frequency at which curve-fitting algorithm will operate. Analysis of the equation reveals that the error is highly dependent on the sampling frequency, since a high sampling rate would permit the algorithm to see more of the variations due to the harmonic content of the signal. If the sampling frequency is decreased below the Nyquist limit, however, the error begins to increase due to aliasing of the input signal. In addition to these errors introduced at these upper and lower limits, the error due to digitization must be included regardless of the sampling rate. Since the analog/digital signal is proportional to magnitude of the analog input, truncation error is introduced in almost all samples. Thus, the digitization error itself is relatively constant, but its error appears to be compounded at higher sampling rates. This effect is portrayed in Figure 5, which compares the digitized results for two sampling frequencies. The truncated digital representation of the three samples begin to approach a common value as the sampling rate increases. Since the algorithm squares the value of the center sample and subtracts the product of the adjacent samples, the comparison of these two numbers of approximately equal magnitudes will result in even higher percentage errors.

Obviously, there must be an optimum sampling rate at which the combined effect of all these errors is reduced to a minimum. This frequency was found by a statistical analysis of the curve-fitting equation and its response to varying types of fault currents. A series of faults were produced to test the algorithm at 190 discrete sampling rates in a range of 1.01 to 20.01 samples per cycle. The .01 sample per cycle offset was introduced to prevent synchronization of the sampling frequency and the waveform under test. Each sampling rate was analyzed for its response to 144 faults, since twelve fault current magnitudes ranging from one to twenty-four times tap current were used in conjuntion with twelve fault inception angles over the full 360^{Θ} range. Each estimate of the current peak was adjusted to provide twelve-bit accuracy and was compared to a double-precision thirty-two bit number representing the true current magnitude. The maximum error for each combination of fault current magnitude and inception angle was found, with these errors subsequently evaluated to determine the overall maximum error for the 144 fault waveforms. The first standard deviation for the run containing the maximum error was tabulated in addition to the average standard deviation of all the runs. Due to the volume and complexity of the calculations involved, the statistical analysis was programmed in a high level language, FORTRAN, and was



Low Sampling Frequency

High Sampling Frequency

Figure 5. Error in Digitization Process

executed on a DEC 11/34 minicomputer. Although the minicomputer has much greater speed and capability than a microprocessor, the program still required up to twenty hours of computational time to perform the four million peak current estimates in addition to the error analysis routines. The source listing for the analysis program is given in the Appendix.

The results of the statistical analysis were tabulated by the minicomputer, but the wealth of data required a more suitable means of interpretation. Thus, a routine was written to display the data on a DEC VT11 videographics terminal. Figures 6 through 10 are photographs of some of the resulting displays. As can be seen from Figure 6, the minimum errors for a purely sinusoidal fault current occur in a range from three to six samples per cycle. The greater errors in the negative direction, corresponding to underestimation, may be attributed to digitization and have a minimum of approximately 3%. The tremendous errors at lower frequencies are caused by aliasing, while the gradually increasing errors at higher frequencies are due to the comparision of values which are approaching the same magnitude. The graphs of Figure 7 offer qualitative support for these conclusions, since the minimum individual standard deviations as well as the minimum average deviations fall in a range corresponding to the minimum errors discussed above.

The current waveforms on an actual power system are rarely pure sinusoids, however, and often contain a relatively high percentage of harmonics. Figure 8 displays the effect of the addition of 0.3% second and 0.5% third harmonics to the original waveform. Although the actual system harmonics may be higher, the active filters discussed in the hardware section will severely attenuate these components. Thus, the graph shows that the addition of harmonics has little effect on the overall

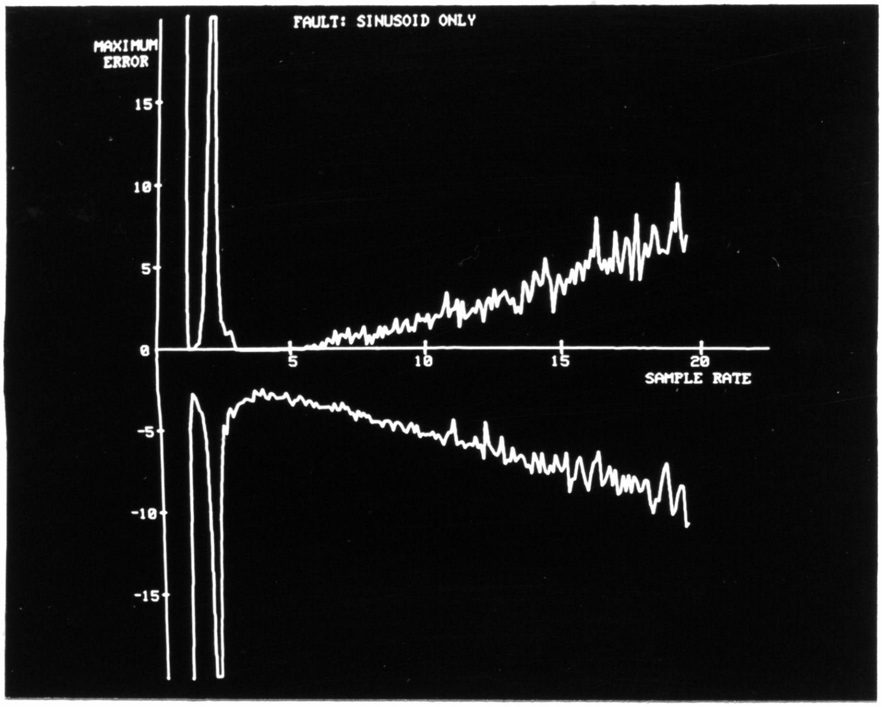


Figure 6. Algorithm Error: Sinusoid

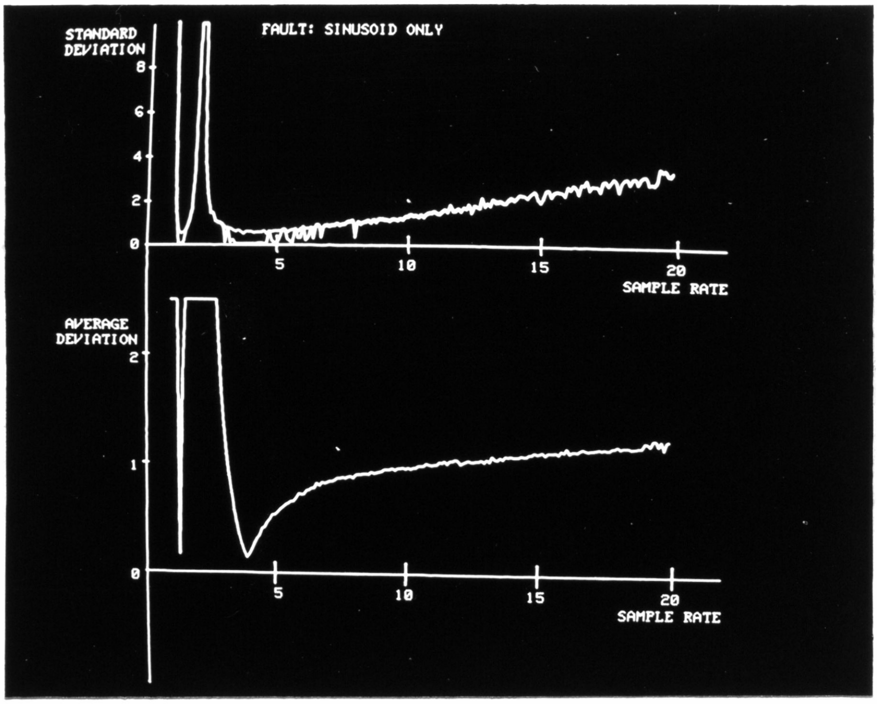


Figure 7. Deviation of Algorithm Error: Sinusoid

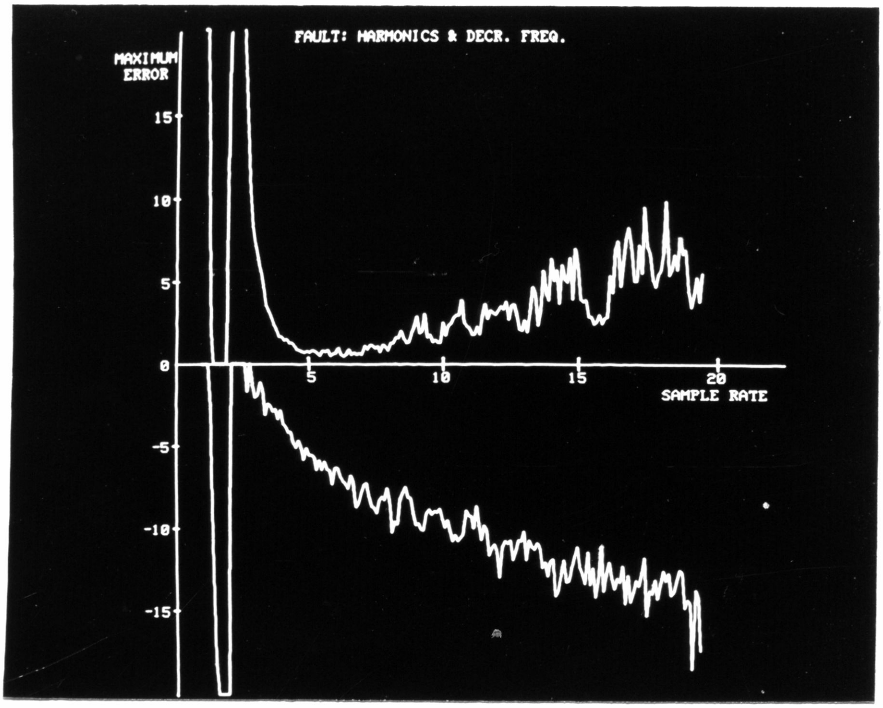


Figure 8. Algorithm Error: Harmonics

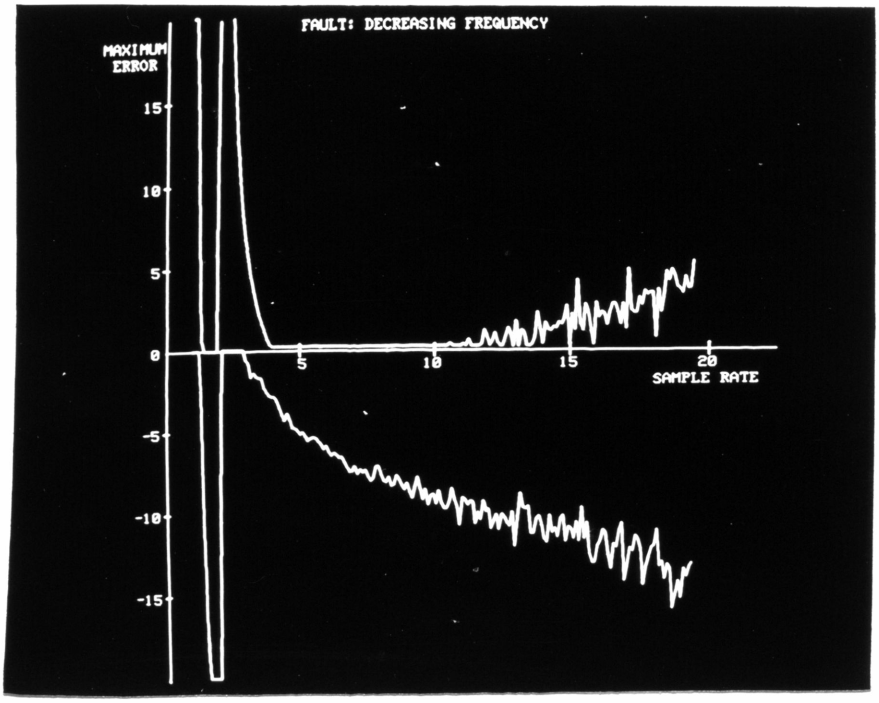


Figure 9. Algorithm Error: Decreasing Frequency

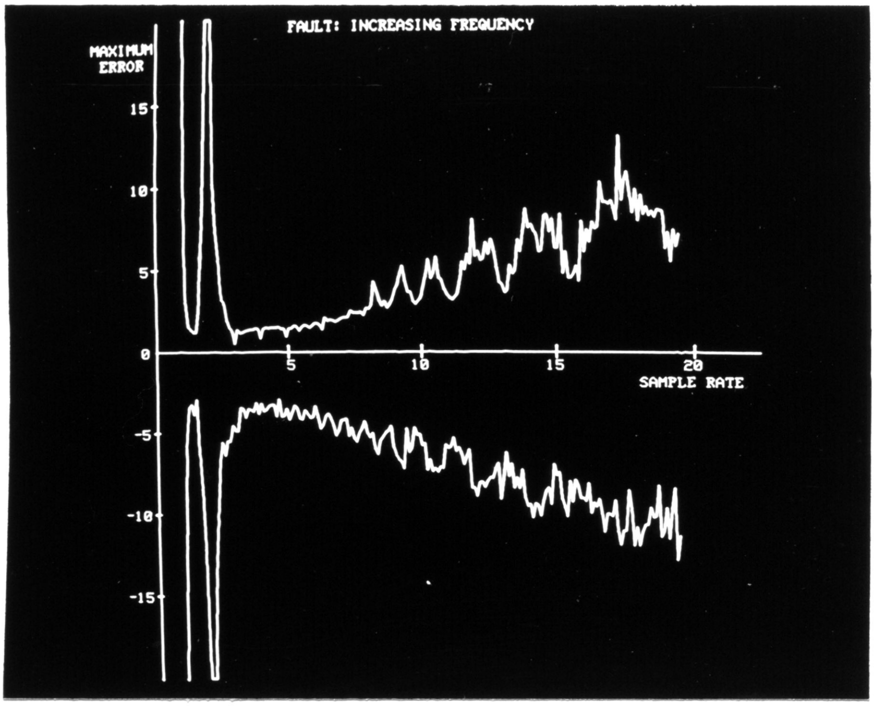


Figure 10. Algorithm Error: Increasing Frequency

shape of the error envelope, but does result in a slight shift upward. This is due to the number of points at which the fundamental and the harmonics have summed together to give a value greater than that of the fundamental alone.

Frequency changes also occur during system disturbances and must be taken into consideration in any error analysis. A fault waveform incorporating a 1.5% decrease in frequency over 10 cycles in addition to harmonics was utilized to test the algorithm, and the resulting data is shown in Figure 9. As before, the shape of the envelope is essentially unchanged, but a slight shift in the tendency toward underestimation is observed. For a corresponding increase in frequency, the envelope is seen to shift slightly upward once again (Figure 10). For these two cases, as well as for the addition of harmonics at a constant frequency, the standard deviation and average deviation plots remained relatively unchanged.

On the basis of the data presented by these plots, a sampling rate of five samples per cycle was chosen for implementation of the sinusoidal curve-fitting algorithm. Thus, the maximum error in estimation of the peak is approximately ±5%, although the typical value of the error is much less. A direct comparison with the accuracy of the electromagnetic relay is difficult, since the error for these devices is given in time to trip. Thus, a relay may be rated at ±5% error in time to trip for a time dial setting of six and a current of two times the tap value [29]. The error for other time dial settings and current magnitudes are undefined, and are highly dependent on the shape of the time-current characteristic. For an extremely inverse curve, the error would be much larger at small current magnitudes than at large magnitudes, since the curve is much steeper at lower multiples of tap. An analysis of the error in time to trip for the microprocessor based relay would yield similar results. Thus, the error is seen to be more dependent on the actual shape of the time-current characteristic than on the error in estimation of the current peak, as long as the estimation error is relatively small.

No-Fault Routine *

The no-fault routine, which is the relay's normal mode of operation, requires the microprocessor to constantly monitor the magnitude of the line current to determine if a fault does exist. This fault determination is based on one of two criteria: (1) the instantaneous current magnitude is above a preset limit corresponding to the maximum short term capacity of the line, or (2) the predicted peak current magnitude is above a second limit corresponding to the long term capacity of the line. In the first case, which involves a strict level comparison, a positive response will send the processor into the instantaneous trip routine. For the second case, however, the processor must utilize the sinusoidal curve-fitting equation to predict the peak magnitude of the current. If the predicted peak is above the preset level, but the sample magnitude is below the instantaneous limit, the processor is sent into the timed-trip routine. A further explanation of the function of these routines may be found in the corresponding subsections.

In addition to its primary role of fault detection, the no-fault routine is capable of performing various secondary functions which are not feasible for implementation in conventional electromechanical relays.

*The remainder of this discussion on the software package is based on a thesis submitted by R. Page Heller to the Graduate College of Texas A&M University in December, 1978.

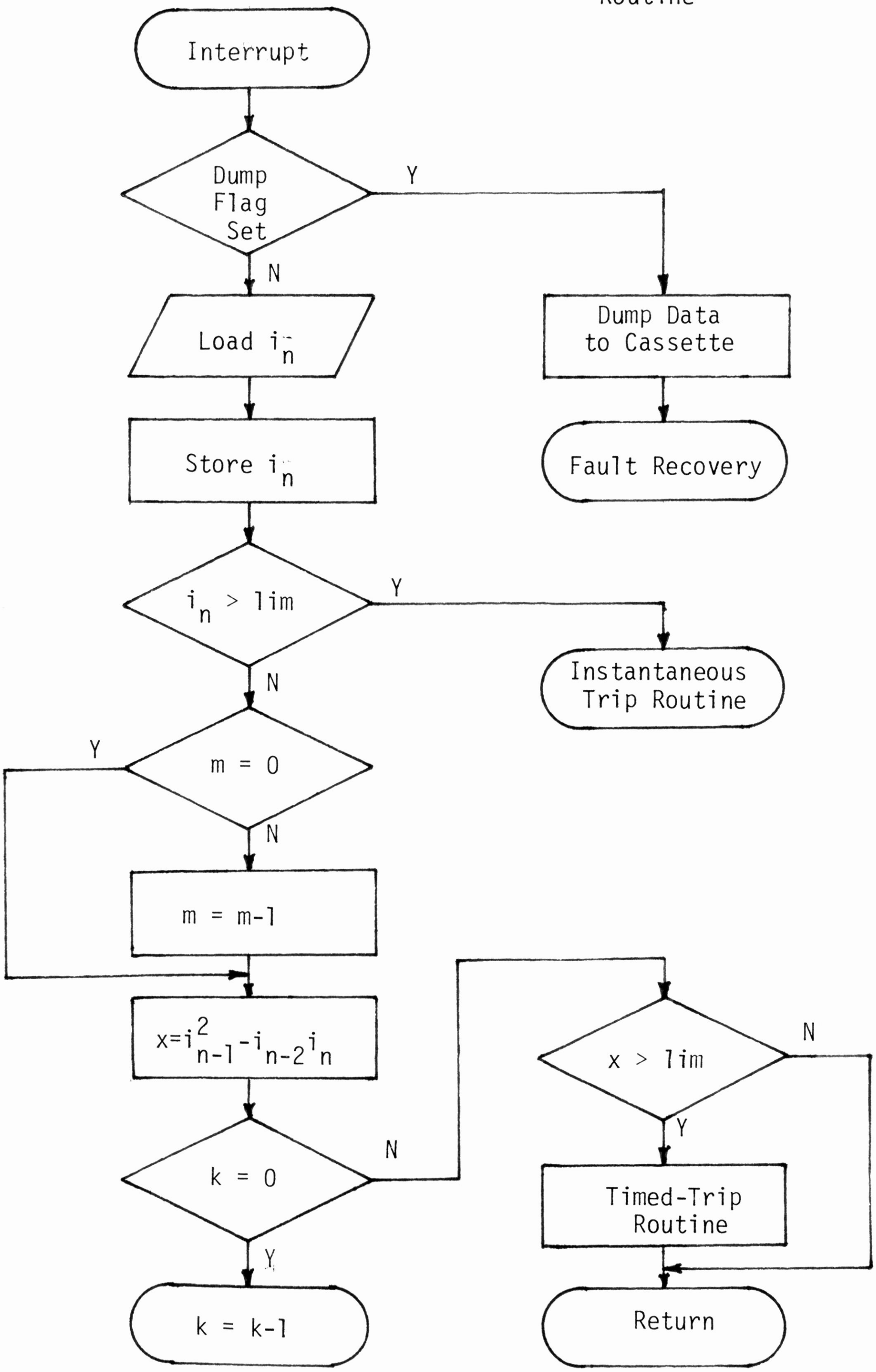
Sampled data values are stored in a continuous memory loop to allow convenient monitoring of the system at all times. In addition, the continuous loop storage method permits the reconstruction of a finite portion of the waveform prior to the determination of a fault on the line. This data is automatically transferred to cassette tape to allow storage of greater quantities of data and to permit analysis of the data at a later time. The output of a real-time clock is included in the stored data to provide a time reference for determination of the sequence of events in various portions of the system.

As shown in the flow chart of Figure 11, data logging is the only additional function which has been implemented in the prototype relay. Other functions may be included, however, and will be incorporated in future versions of the relay. A very desirable function would be a comlete self-diagnostic to allow the processor to determine the status of individual subsystem components. (A very simple diagnostic has been included in the prototype and is described in the section on testing). Thus, the relay could signal a need for total or partial replacement, and subsequently prevent unnecessary damage to the system in the event of a fault. Other functions would permit the processor to set warning flags or send messages to the operator to alert him to various undesirable situations within the system.

Instantaneous Trip Routine

The operation of the instantaneous trip routine in the microprocessor based relay is analogous to the level comparison performed by the electromechanical relay. Obviously, a sample rate of five cycles per sample is much too slow to allow adequate protection against currents large enough to begin damage to the line within one cycle of their inception. Thus,

Figure 11. Flow Chart: No-Fault Routine



this routine is operated at a sampling frequency of 7200Hz, or 120 samples per cycle. A comparison to the desired instantaneous trip setting requires no complicated calculations, and may be performed at very high speed. A fault waveform which causes twenty consecutive samples to be above the preset limit will initiate a signal to trip the breaker and disconnect the line. Although the required number of samples may be altered, twenty samples at 7200Hz corresponds to the portion of each half-cycle which has a magnitude greater than the rms value for the waveform. A comparison of the flow charts in Figures 11 and 12 reveals that the counter for the instantaneous trip routine is incremented each time a sample is above the limit and is decremented, if not already zero, each time the sample is below the limit. This measure was included to assure the integrity of the counter in case the magnitude of the fault current falls below the preset limit for one or two samples before returning to a value above the trip limit. Thus, this type of waveform would only delay the trip signal by a time equivalent to twice the number of samples which fall below the desired instantaneous limit. After a trip signal is initiated, a flag is set to notify the general routine that a fault has occurred and that the contents of the memory loop should be transferred to cassette tape, insuring the recovery of both pre-fault and post-fault data.

Timed Trip Routine

The timed-trip routine is utilized to evaluate fault currents which are below the instantaneous trip limit, but still pose a potential threat to the safety of the system. This routine utilizes the peak current magnitude calculated from the sinusoidal curve-fitting algorithm to determine the appropriate delay time before a trip signal should be sent. As

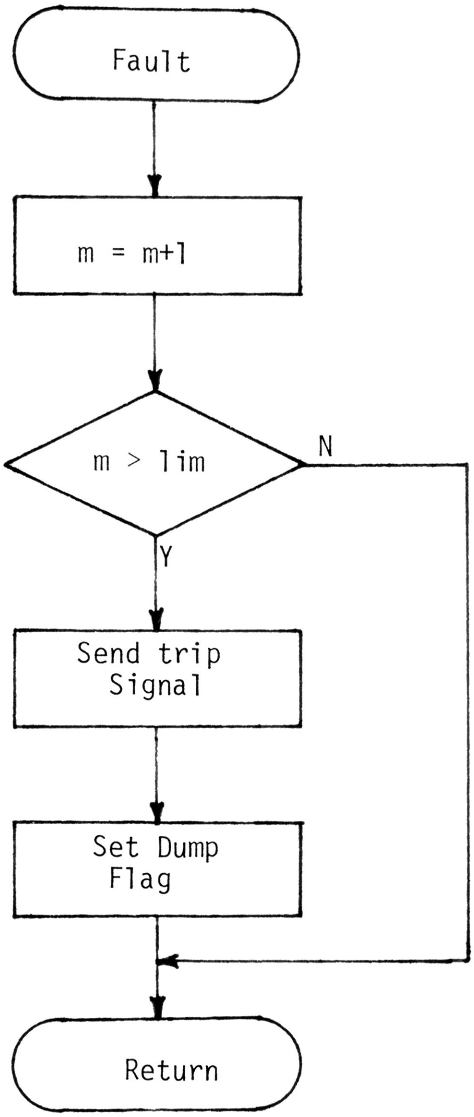


Figure 12. Flow Chart: Instantaneous Trip Routine

previously discussed, this delay allows time for fuses or other protective devices to alleviate the condition by sectionalizing a smaller portion of the system. The delay time must be computed from a knowledge of the calculated current magnitude and the time-current characteristic for the relay. Although various analytical methods have been developed to allow calculation of the time-delay from the known parameters of standard CO curves [28,29], a much simpler method has been implemented in the prototype relay. This approach utilizes a table stored in non-volatile memory to provide a base value to which the timed-trip counter is added, as shown in Figure 13. The sum is then compared to a preset limit to determine if a trip signal should be sent. As in the instantaneous trip routine, the initiation of a trip signal will also result in the setting of a flag to notify the general routine that the contents of the memory loop should be dumped to cassette. If the sum of the base value and the timed-trip counter is below the trip limit, however, the timed-trip counter is preserved and control is returned to the general routine to begin analysis of the next data sample.

Utilization of a table look-up to provide the base value for a timed-trip analysis removes any previous restrictions on the shape of the time-current characteristic. Thus, the relay is not required to duplicate the mechanical characteristics of conventional relays, nor must it implement a continuous mathematical function as do the analytical approaches. This allows the implementation of customized curves to provide adequate coordination regardless of system configuration. In addition, the relay's flexibility is increased tremendously, since the only requirement for changing the time-current characteristic is the relacement of a single memory chip.

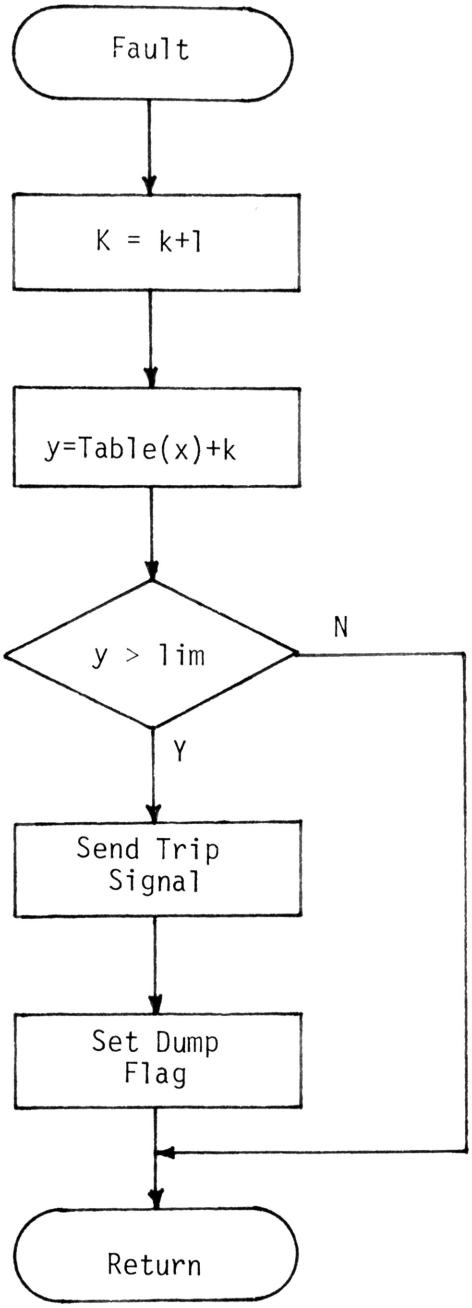


Figure 13. Flow Chart: Timed-Trip Routine

Fault Recovery

After a trip signal has been sent to the breaker, the relay enters a fault recovery routine which supplements its primary role of fault current interruption. For the electromechanical relay, the recovery routine consists of two functions: (1) dropping a flag to distinguish between an instantaneous trip and a time-trip, and (2) rotating the disc back to its original pre-fault position. The microprocessor based relay, however, allows an expansion of the recovery routine. As previously mentioned, the contents of the memory loop are dumped to cassette tape after each fault to allow reconstruction and analysis of the pre-fault as well as post-fault waveforms. The relay also initiates visible and audible alarms to signal the occurrence of a fault, and appropriate messages may be sent to the operator if a communications link is provided to the control room. If desired the microprocessor may also communicate with relays in other portions of the system to automatically control the reclosing of breakers in those areas which have been deenergized, but are otherwise unaffected by the fault.

HARDWARE PACKAGE

Overview

The primary advantage of the microprocessor based relay is the inherent intelligence provided by the central processing unit, or CPU. This unit utilizes non-volatile read-only memory to store time-invariant data, including the general software routines and a table for the desired time-current characteristics. In addition, random access memory is included to provide a scratchpad for calculations and to permit storage of sampled data. Peripheral interfaces allow the stored data to be transferred to cassette tape, and permit an operator to communicate directly with the central processing unit of the relay.

In spite of the intelligence of the CPU, however, this device cannot deal directly with the analog input signals. Thus, a signal conditioner must be inserted to convert the data from analog to digital form. In addition, various filtering and buffering techniques must be utilized to insure that the CPU receives accurate data in the proper format.

Signal Conditioning

The input signal is supplied to the relay from an instrumentation type current transformer which produces a current in its secondary coil proportional to the current in the primary coil. To allow proper processing of the signal, this current is converted to a voltage by a 0.01 ohm precision shunt, with a metal oxide varistor (MOV) placed across its terminals to aid in surge suppression. The MOV has a transfer characteristic similar to a back-to-back diode pair, and will attenuate any spikes due to normal system transients or due to abnormal conditions such as lightning strokes. A passive, low-pass RC-filter is utilized for additional surge suppression and to attenuate undesirable high frequency transients.

Since the signal produced by the current transformer has no ground reference, the resulting voltage at the output of the low-pass filter is floating with respect to the system ground. Consequently, the voltage is fed into a unity-gain differential amplifier to produce a single-ended output voltage referenced to ground. Although the amplifier does not affect the magnitude of the voltage, it does buffer the signal to provide a clean input waveform for the succeeding stages.

As can be seen from Figure 14, the next stage of the signal conditioner is an active 60-Hz bandpass filter. This stage removes any d.c. component which may be included in the waveform, since this component would have an adverse effect on the sinusoidal curve-fitting algorithm. Frequencies above the 60 Hz fundamental are also severely attenuated to minimize their influence on the analysis of the waveform.

The filtered signal must then be converted from its analog form to a corresponding digital representation for manipulation by the central processing unit. A sample-hold is utilized to sample the input signal at the desired frequency and provide a constant output until the analog/digital converter has had sufficient time to complete the digitization of the sample. For the prototype, an SHM-IC-1 sample-hold was used in conjunction with a .01μF holding capacitor to give an acquisition time of 10 μsec. The analog/digital converter, an ADC-EH12B3, provides a 12-bit digital representation of the analog input and requires 2.0 μsec to complete the conversion. Thus, this combination is capable of supplying digitized signals to the microprocessor at a rate well above the required 7200 Hz. Figure 15 is a photograph of the assembled signal conditioning portion of the system.

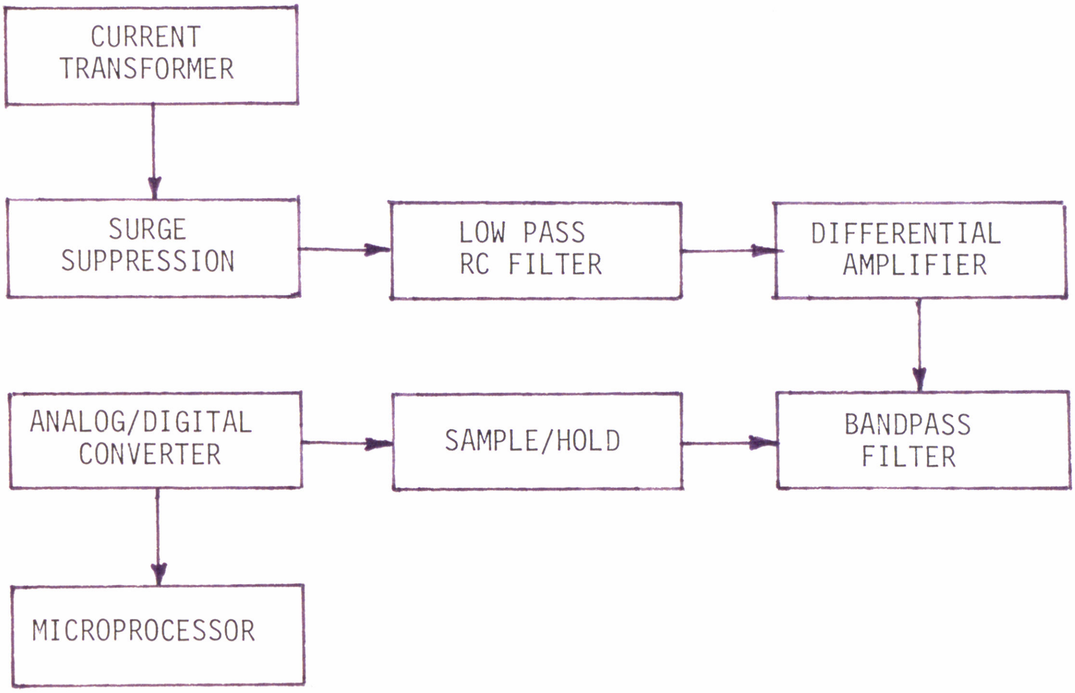


FIGURE 14 BLOCK DIAGRAM
SIGNAL CONDITIONER

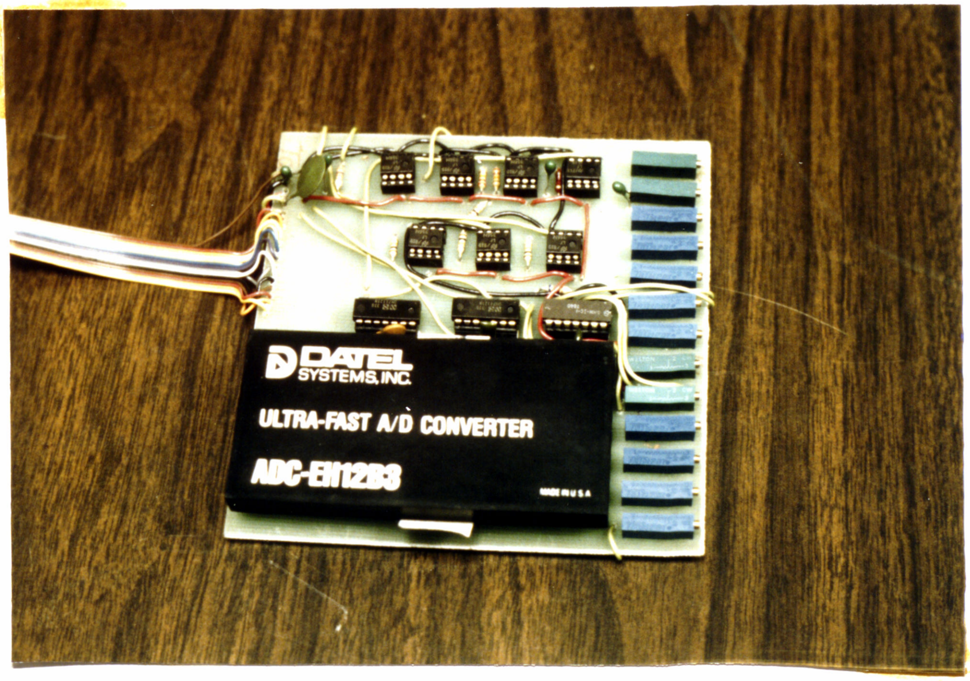


Figure 15. Assembled Signal Conditioner

Microprocessor System

The intelligence of the prototype relay is concentrated in the CPU, or central processing unit, which is a Motorola M6802 microprocessor chip. The operations of the CPU are generally controlled by programs written in assembly language, a convenient mnemonic representation of the actual machine language instructions. Each instruction represents a binary code which is utilized by the CPU to control all internal data manipulations. Although programming may be facilitated by the use of higher level languages, assembly language results in a more compact program requiring the least amount of execution time. Since both time and memory space are at a premium in real-time control operations, all software routines are implemented in assembly language.

The program to control the CPU is stored in erasable programmable read-only-memory (EPROM), since the non-volatile nature of this memory will preserve the instructions even if all power to the relay is lost. The prototype utilizes a 2716 EPROM, with the fault analysis program occupying approximately one-half of the 2K-bytes of available area. The remaining area is reserved for storage of the table for the customized time-current curve.

Additional memory elements are required for scratchpad calculations and for storage of the sampled data values. A 6810 RAM, or random access memory, is utilized as a scratchpad and furnishes 128 bytes of available area. Two 2114 RAMs were required to provide the desired 1K of data storage space, since each chip only provides four bits of the total eight-bit word. It was also necessary to utilize two eight-bit words of memory space for storage of each twelve-bit sample, thereby reducing the effective storage loop to only 512 words. Although the loop could be increased with the addition of more memory, this value is sufficient for the recording of slightly more than twenty-five cycles of data at a rate of twenty samples per cycle. This rate is higher than that used for calculations of the peak current magnitude, but it allows a better reconstruction of the data for analysis at a later time. The high sampling rate used for the instantaneous trip routine was also avoided since it would introduce unnecessary detail and would require excessive amounts of memory for adequate storage.

From the block diagram in Figure 16, it can be seen that the CPU utilizes a common data bus to communicate with the individual memory elements as well as the various peripheral devices. The lines used to address each of these devices are also grouped into a common bus. Since the CPU is not capable of driving this many devices directly, it was necessary to buffer both the address and data busses. This was accomplished through the use of two DS8833 quad transceivers to buffer the eight data lines. The address bus only requires one-way communication, which allowed the use of two SN74367 hex bus drivers.

The number of devices with which the CPU is required to communicate also dictated the use of memory mapping. This technique utilizes logic external to the CPU to decode the address lines and supply an enable signal to the selected device. Thus, only one external device is enabled at a time to prevent multiple sets of data from appearing on the common data bus at the same time. The required memory mapping was easily accomplished by the use of an SN74138 three-to-eight decoder to select the proper external device on the basis of the address supplied by the microprocessor.

Peripherals

The tremendous capabilities of the microprocessor cannot be realized,

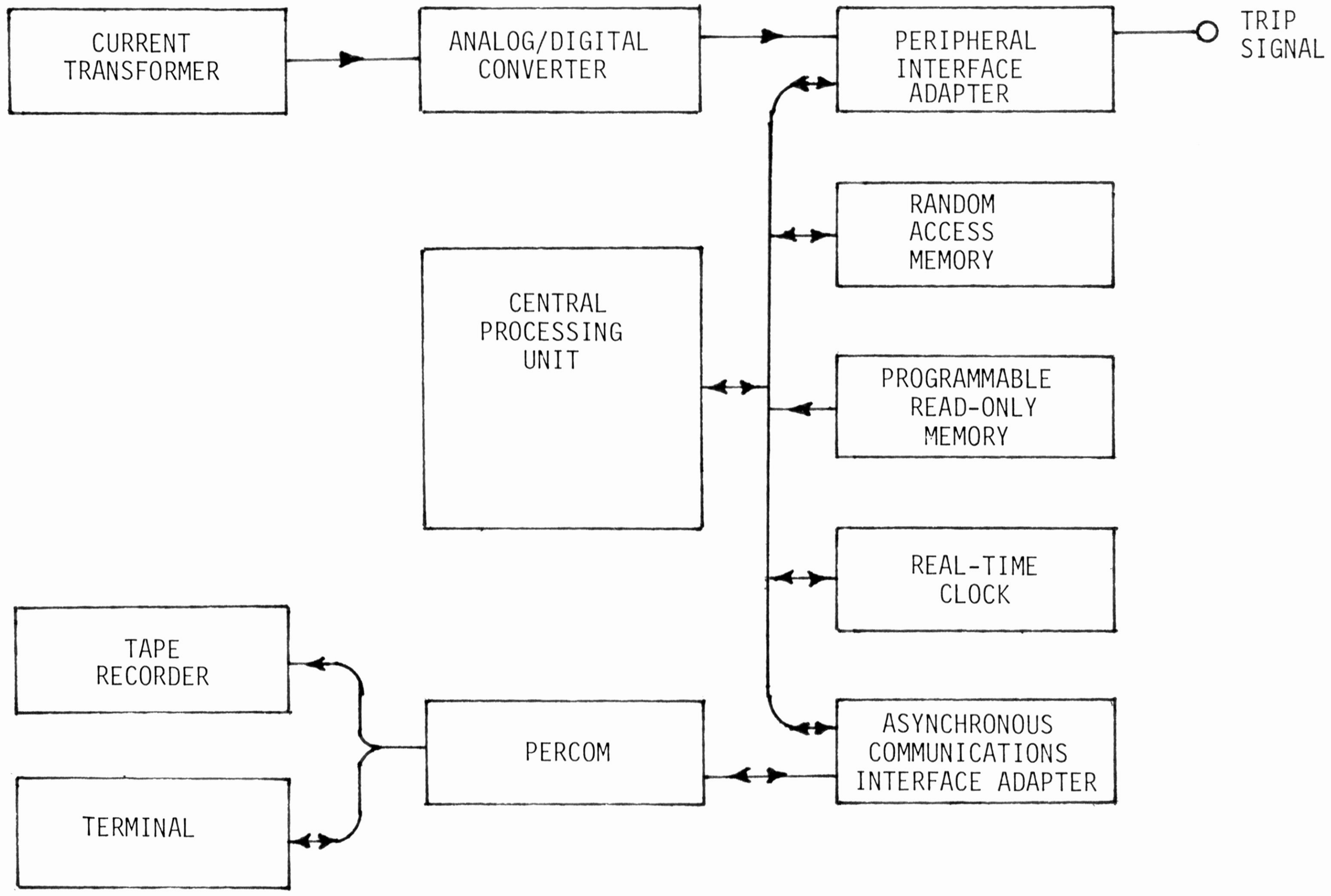


FIGURE 16 BLOCK DIAGRAM
MICROPROCESSOR SYSTEM

however, unless adequate means of communication with the outside world are provided. Obviously, the CPU must have access to the sampled data furnished by the signal conditioner before it can initiate execution of the general software routine. The required link is provided by a multifunction device known as a peripheral interface adapter, or PIA. This device accepts a digitized sample from the A/D and holds the data in a buffer until the CPU requests to have it placed on the common data bus. This prevents the loss of data in the event the CPU is not ready to accept the sample when the A/D signals that it has completed its conversion. The PIA also provides several control lines which may be used to activate devices external to the relay. The prototype utilizes one of these lines to control the gate of a thyristor which can trip a breaker to interrupt dangerous fault currents. Additional lines have been utilized to activate alarms in the control room if corrective action has been required.

Since the PIA only provides a parallel input/output port, an asynchronous communications interface adapter (ACIA) has been utilized as a serial input/output port. This device allows the microprocessor to dump sequential data to an external device, such as a cassette tape recorder or a remote terminal unit. The ACIA also provides a means of sending messages back to the microprocessor, since it will accept serial data from these same external devices. Although the ACIA does not provide user-defined control lines, its programmable control registers allow the microprocessor to reconfigure the port, thereby permitting two-way communication with virtually any device utilizing serial data.

A real-time clock has also been interfaced to the CPU to allow time-tagging of all faults and the subsequent breaker operations. This subsystem was designed around an MM5318 digital clock module, which provides multiplexed binary coded decimal outputs compatible with the common data bus. Provisions have been made to allow setting of the clock from an external keyboard.

A Percom cassette interface has been included in the prototype relay design to facilitate the recording of sampled data. The Percom provides the necessary control signals for automatic operation of the tape recorder motor. In addition, an internal baud rate generator supplies all required clocking signals to ensure dependable data exchange.

A photograph of the assembled microprocessor system is shown in Figure 17. This board contains the CPU, memory, and all peripherals with the exception of the real-time clock and the Percom. The real-time clock has been assembled on the mother-board, which allows the microprocessor and the signal conditioner to exchange information. In the prototype model, the Percom will be housed external to the relay box. Future models will not require the Percom, since they will communicate directly with a remote terminal unit, bypassing the need for individual cassette tape recorders at each relay. Thus, Figure 18 shows the completed prototype, with the exception of a copper board which will be inserted between the microprocessor and the signal conditioner to minize the possibility of electromagnetic interference. The prototype will be housed in a standard CO relay case and may be mounted directly in the racks presently used for all substation relays.

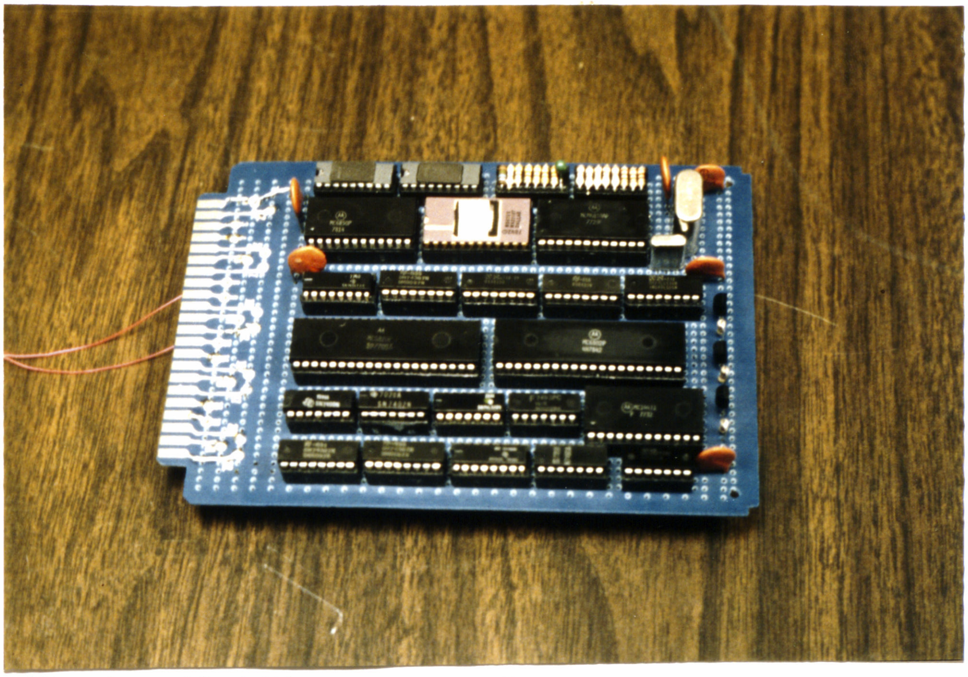


Figure 17. Assembled Microprocessor System

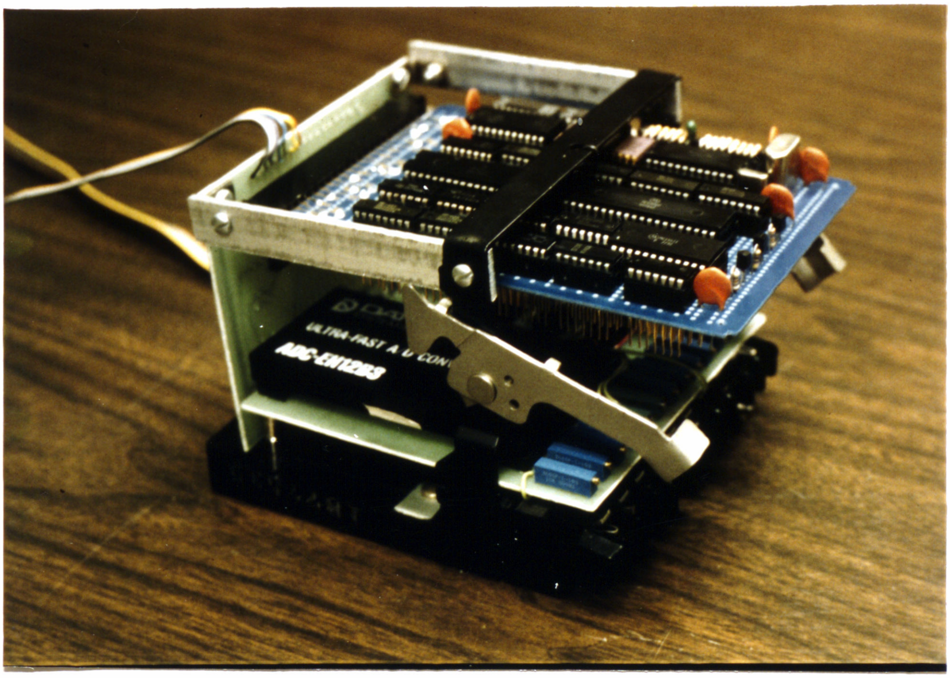


Figure 18. Assembled Prototype

TEST PROCEDURES

Overview

The design and construction of any new piece of equipment must necessarily be followed by appropriate laboratory tests to determine if the device actually meets the design specifications. Negative results at this stage require a careful review of the design methodology and construction practices utilized in the development of the device to determine the origin of the problem. Positive results, on the other hand, merely indicate the apparent soundness of the design approach. A final verification of the device's capabilities must be reserved until a prototype model has survived the rigors of extensive field testing.

Fault Simulator

Initial laboratory tests may be conducted with standard test equipment to verify that the major functions of the relay are operating properly. A careful simulation of actual fault data is required, however, to analyse the finer aspects of the relay's operating characteristics. Therefore, a special purpose distribution model will be employed to produce the necessary fault waveforms. The model utilizes a distributed network of resistive and inductive elements to simulate the characteristics of a ten-mile distribution feeder [22]. The model is constructed in four sections to allow the simulation of a phase-to-ground fault at five distinct locations: the relay tie-in, two miles, four miles, six miles, eight miles, and across the load at ten miles.

The flexibility of the model is greatly enhanced by incorporating controls to aid in the simulation of specific fault waveforms. In addition to the location of the fault, the inception angle may be monitored on an oscilloscope and adjusted to the precise angle desired. The power factor may also be selected from three discrete values: 0.80, 0.90., and 0.95.

Prototype Tests

The first stage of prototype testing will involve a thorough check of the operational status of all hardware components. This requires the calibration of the sample-hold and the analog/digital converter as well as the setting of the instantaneous trip limit. Switches have been provided to disconnect each device from the rest of the system, thus preventing any errors due to interaction with other components. After the calibration has been completed, a simple diagnostic test of the software routines will be performed. A light emitting diode has been provided to signal that the software is in operating condition. Thus, the microprocessor is programmed to send a signal to the appropriate port after completion of each loop through the general routine. This resets a counter which is clocked by the end-of-conversion signal from the A/D. If fifteen clocking signals are received before the counter is reset, the LED will be turned off, signalling a failure within the software routines.

The next phase of testing will utilize the distribution model to simulate actual fault conditions. The response of the relay will be carefully monitored to determine if the correct action was initiated. If a desired trip signal does occur, the time to trip will be recorded and compared to the proper time-current characteristic of a standard electromechanical relay. Any necessary modifications to the general software routine, the individual trip routines, or the tabulated time-current characteristic will be made at this time. This phase of the testing and modification process will continue until the microprocessor based relay can meet or exceed all specifications of the electromechanical relay in addition to proper performance of all desired secondary functions.

Although the fault simulator is a powerful tool for prototype testing, it lacks the capability to simulate non-sinusoidal waveforms. Figures 19 and 20 depict the idealized current waveforms for six- and twelve-pulse rectifiers utilized in many industrial applications. Although the actual system current will not contain all of the high frequency components inherent to these waveforms, it will differ significantly from a 60-Hz sinusoid. Thus, it would be desirable to subject the prototype relay to laboratory tests incorporating such non-sinusoidal waveforms. A simulator of this type is not currently available, however, so all testing must be performed with the sinusoidal fault simulator.

Field Tests

Once the relay has passed all laboratory tests, it will be installed in parallel with a conventional electromechanical relay. The prototype relay will not initially be allowed to trip a breaker, but the appropriate alarms will be sent to the control room to alert the operator that corrective action was necessary. The recording function will also be enabled to permit an analysis of reconstructed fault waveforms and the microprocessor's subsequent response. Thus, actual fault data may be collected without compromising the integrity of the system. After sufficient time has elapsed to verify the correct operation of the relay, it will finally be wired into the trip circuit and given control over the breaker operation. Careful monitoring of the relay will be continued in an effort to determine any problems which may occur as a result of errors in the software

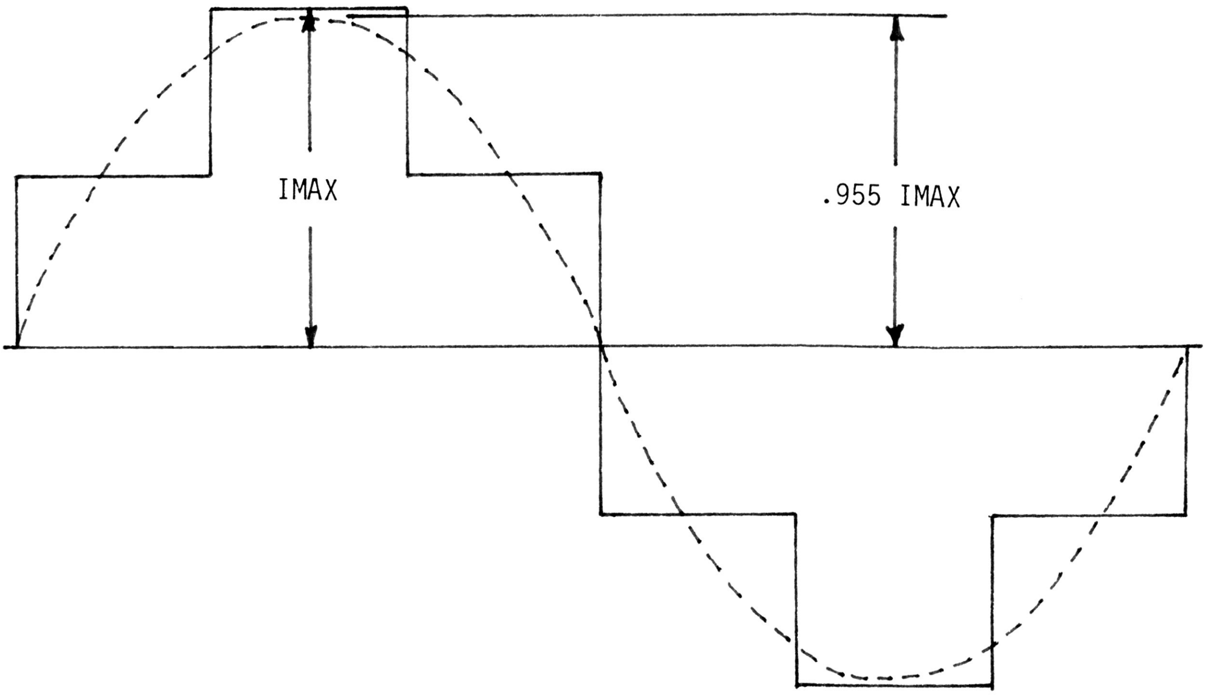


Figure 19. Waveform for 6-Pulse Rectifier

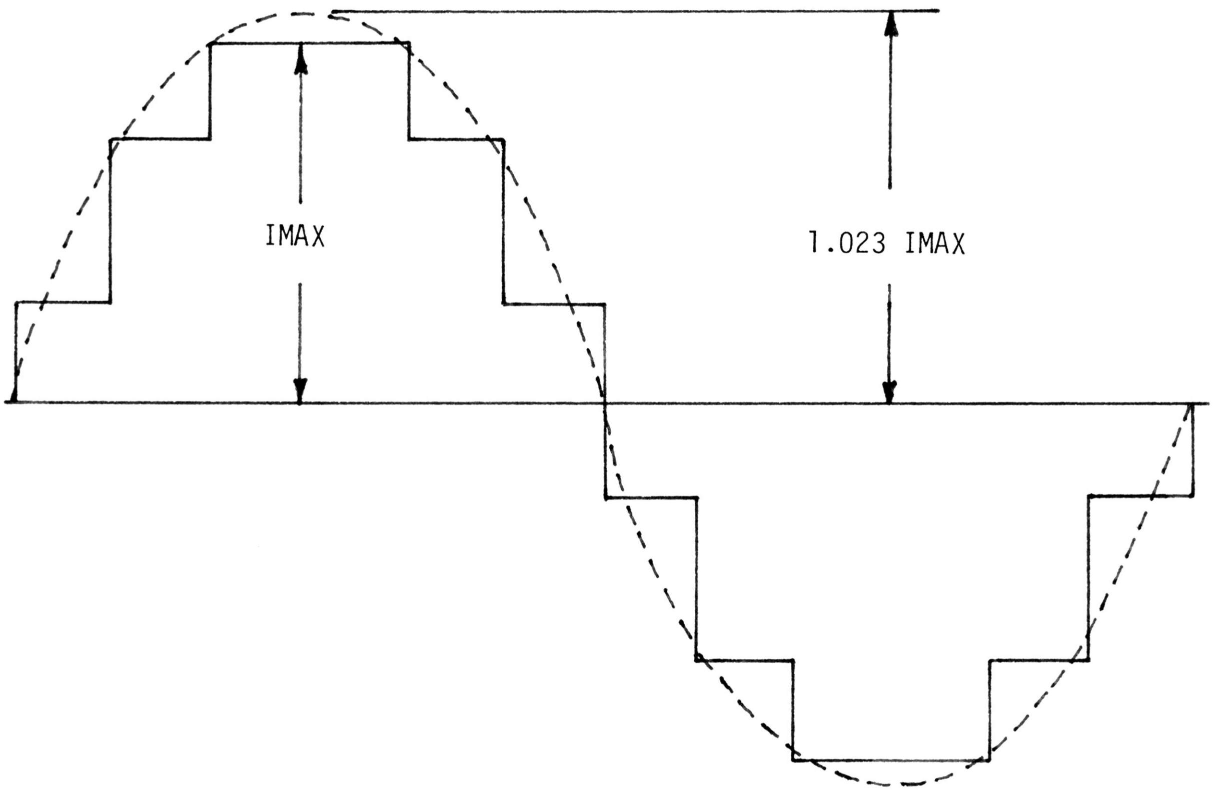


Figure 20. Waveform for 12-Pulse Rectifier

or hardware portions of the system.

A significant problem which will be encountered during field tests, however, is the relatively infrequent occurrence of faults. The project sponsors, Dow Chemical Company of Freeport, have agreed to install the relay on their most fault-prone distribution feeder, but this line only averages eight faults per year. The probability that the microprocessor based relay will actually see a fault is even further reduced since it will only monitor a single phase of the three-phase feeder. Thus, it will require months, or possibly years, before a valid statistical data base can be collected from the installation of a single prototype relay.

CONCLUSION

The increased speed and capability of microprocessors have been the key to their acceptance in a wide variety of technological areas previously immune to the influence of machine-based intelligence. Rapidly decreasing hardware costs have also encouraged the development of customized computer systems for applications in these new fields. Although power systems have followed this trend by integrating computers into large scale control and monitoring networks, no previous attempt has been made to include these functions as well as overcurrent protection into a single device. This new approach solves many of the problems associated with conventional electromechanical relays by providing a standardized relay flexible enough to cope with changing system configurations.

The conflicting claims of standardization and flexibility reflect the hardware/software division within the relay itself. Standardization is achieved through the use of multi-function components and subsystems to provide the basic hardware functions. The minimal effort required to replace a single chip containing all software routines and time-current characteristics permits the use of specialized programs and customized curves to facilitate coordination. Thus, additional functions or modification of existing functions may be implemented without the requirement of extensive hardware redesign.

At present levels, the projected cost of mass-produced microprocessor relays will exceed that of conventional electromechanical devices (approximately $350 compared to $250), but the additional functions provided by the intelligent relays far outweigh the cost differential. The ability to record pre-fault as well as post-fault waveforms will permit an in-depth analysis of each system disturbance to determine the probable cause and to point out possible preventive measures. If desired, normal system data may also be recorded to allow analysis of the harmonic content of the waveform or any other irregularities which may exist.

The communications capabilities inherent to microprocessors, however, seem to be the most significant advantage over conventional devices. This aspect allows the relay to nofify the operator of undesirable conditions on that portion of the system such as abnormal transients, current transformer failure, or even a failure within the relay itself. In addition, the relay may be allowed to communicate with other devices within the system, permitting more efficient control of functions normally performed by the operator or a central computer, such as breaker reclosing and re-sectionalizing, generation control, and load shedding.

Naturally, there are also disadvantages associated with the implementation of microprocessor based relays. In order to guarantee a constant power supply for the microprocessor, the substation battery must be utilized as an energy source. Although each relay will present a relatively small load, a large number of such devices might require an expansion of the substation's d.c. supply. In addition, this voltage must be stepped down from its normal 125 volt level to provide power at ±15 and +5 volts. Obviously, this requires the relay ground to float at +15 volts with respect to the substation ground.

The primary deterrent to full-scale implementation of microprocessor based relays, however, is their unknown reliability. In most instances, the relay will sit idle for months or years before a fault requires a rapid and accurate decision process. The reliability of the relay under these conditions cannot be accurately predicted, since the application of real-time computer controls is a relatively new field. Therefore, extensive field tests will be required to determine if the reliability of such devices can meet power system standards.

RECOMMENDATIONS

At the present time, a prototype overcurrent relay has been constructed, but laboratory test have not been conducted to determine the response of the system. Thus, the next step in the development process should be the utilization of the distribution model to stage carefully monitored fault scenarios. Thses tests should reveal any shortcomings of the relay, regardless of their hardware or software orientation. After sufficient laboratory tests have been conducted, the prototype relay should be installed in a distribution substation to permit monitoring under normal system conditions. This will reveal any problems which may occur as a result of insufficient shielding of electromagnetic interference, susceptibility to transients, or other factors which cannot be accurately modeled in a laboratory environment.

On a larger scale, however, the techniques utilized in the development of a microprocessor based overcurrent relay should not be restricted to a single application. It is highly probable that the same basic hardware configuration could be utilized to implement undervoltage and differential relaying functions as well as the original overcurrent function. Thus, a concentrated effort should be focused on the development of the software routines necessary for implementation of these functions. The realization of intelligent devices in these additional applications would enhance the power system's growing dependence on real-time computer controls. This trend promises to provide a much safer, more efficient, and more economical distribution system than previously thought possible.

REFERENCES

[1] R.P. Carter and L.P. Cavero, "Considerations on Future Power System Protection, Control and Measurements", IEEE Conference on Modern Developments in Protection, March 1975.

[2] W.A. Lewis, "Computer Applications of Overcurrent Devices - A Concept Whose Time Has Come", Twenty-ninth Conference for Protective Relay Engineers, April 1976.

[3] R. Poncelot, "The Use of Digital Computers in Network Protection", CIGRE, Paper No. 32-08, 1972.

[4] G. Dromey, et al, "Integrated Control Concepts in Power Systems", CIGRE, Paper No. 34-08, 1974.

[5] W.J. Cheetham, "Computerized Protection or Not? - Primary and Local Back-up Protection", GEC Measurements.

[6] J.E. Caldwell, "Distribution Protection for the 1980's", East Midlands Electricity Board, March 1975.

[7] W.J. Cheetham, "Computerized Protection or Not? - Remote Back-up Protection", GEC Measurements.

[8] F.L.Messec and M. Shea, "Experiences with a Modern Digital Supervisory Control System", Twenty-sixth Conference for Protective Relay Engineers, Texas A&M University, 1973.

[9] J.L. Bowen and L.M. Cox, "New System Control Center, Dallas Power & Light Co.", Twenty-seventh Ccnference for Protective Relay Engineers, Texas A&M University, 1974.

[10] E.A. Udren, "Outline of Computer Relaying Tutorial Session", Protective Relaying Congerence, Georgia Institute of Technology, May 1977.

[11] C. Russell Mason, The Art and Science of Protective Relaying, John Wiley & Sons, Inc., 1956.

[12] Electrical Transmission and Distribution Reference Book, Westinghouse Electric Corporation, East Pittsburgh, Pennsylvania, 1964.

[13] Applied Protective Relaying, Relay-Instrument Division, Westinghouse Electric Corporation, Newark, New Jersey.

[14] M. Ramamoorty, S.N. Lall, V. Saxena, "Digital Inverse Time Overcurrent Relay Using Counters", Dept. of Electrical Engineering, Indian Institute of Technology, Kanpur, India, Unpublished.

REFERENCES

(continued)

[15] Fred Wolf, "Digital Design Techniques Applied to Time Overcurrent Relays", Thirtieth Conference for Protective Relay Engineers, Texas A&M University, April 1977.

[16] R.P. Heller and B.D. Russell, "Microprocessor Algorithm for Overcurrent Protection of Distribution Systems", Control of Power Systems Conference and Exposition, University of Oklahoma, March 1978.

[17] J.W. Horton, "The Use of Walsh Functions for High-Speed Digital Relaying", IEEE PES Summer Meeting, San Francisco, California, July 1975.

[18] P.J. Mann and I.G. Morrison, "Digital Calculation of Impedance for Transmission Line Protection", TPAS Vol. 90, No. 1, Feb. 1971.

[19] G.B. Gilcrest, G.D. Rockefeller, E.A. Udren, "High-Speed Distance Relaying Using a Digital Computer; I-System Description and II-Test Results", TPAS Vol. 91, No. 3, May/June 1972.

[20] R.G. Luckett, P.J. Munday, B.E. Murray, "Substation Based Computer for Control and Protection", IEE (England) Conference Publication 125, Developments in Power System Protection, London, March 1975.

[21] Jun-ichi Makino, Yoshiteru Miki, "Study of Operating Principles and Digital Filters for Protective Relays with Digital Computer", IEEE Paper No. C 75 197-9, 1975 Winter Power Meeting.

[22] B.K. Colburn, R. Bistline, G.Fielder, "Line to Ground Transmission Line Fault Simulator", Texas A&M University, Unpublished.

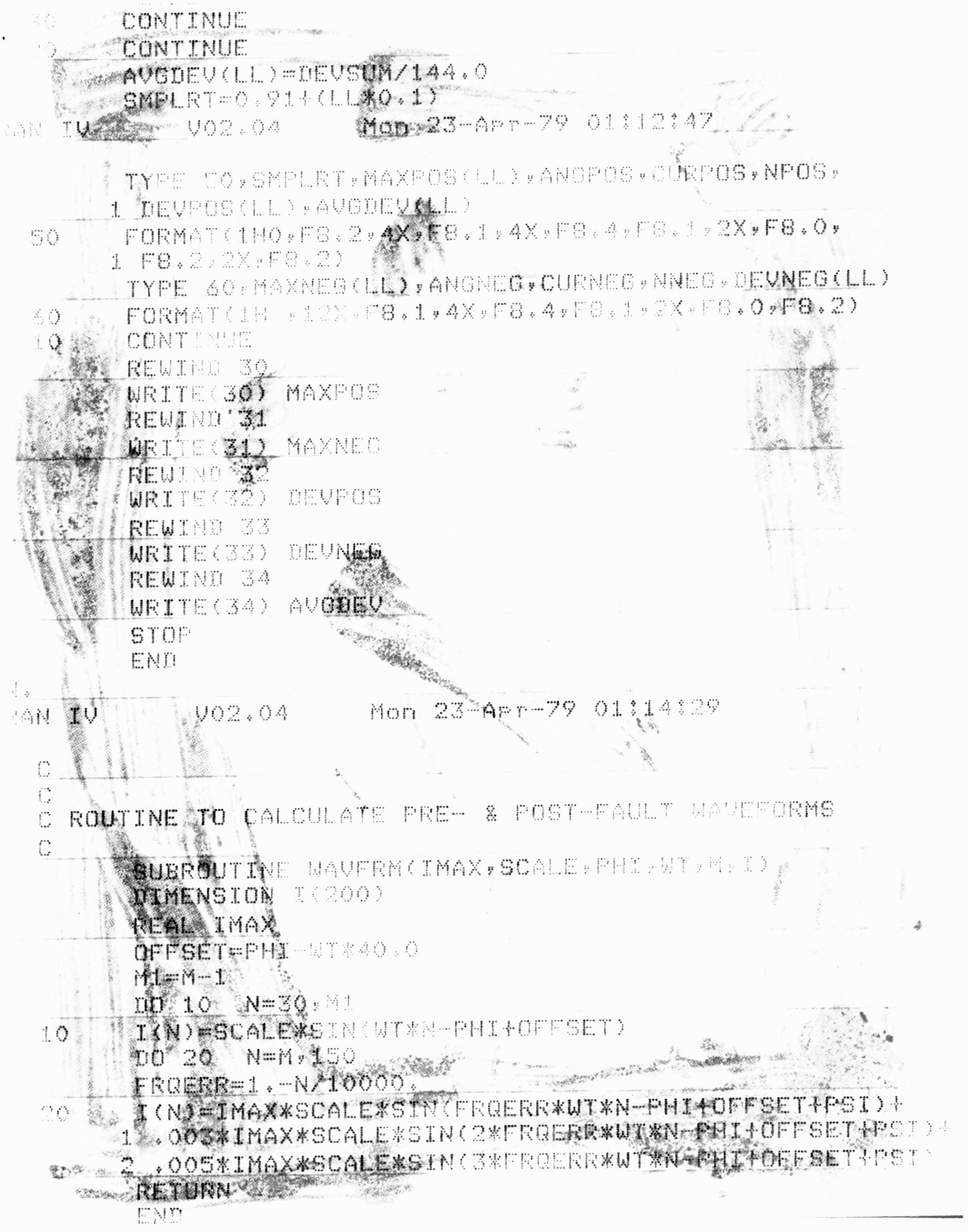
APPENDIX

Source Listing of Error Analysis Program

```
C*****************************************************************
C                                                                *
C     ROUTINE TO CALCULATE THE ERROR FUNCTIONS OF                *
C               OF OVERCURRENT ALGORITHM                         *
C                                                                *
C*****************************************************************
C
      DIMENSION I(200),IX(200),ERRVAL(200),DEVPOS(200)
     1 ,DEVNEG(200),AVGDEV(200)
      REAL IMAX,L,MAXPOS(200),MAXNEG(200),NPOS,NNEG
      SCALE=80.0
      PHI=0.4510
      L=0.471/(6.2832*60.0)
      R=0.3831
      SMPLRT=0.91
      DO 10  LL=1,190
      MAXPOS(LL)=0.0
      MAXNEG(LL)=0.0
      DEVSUM=0.0
      SMPLRT=SMPLRT+0.1
      WT=6.2832/SMPLRT
      PSI=PHI-0.5236
      DO 20  KK=1,12
      PSI=PSI+0.5236
      M=40+IFIX((PSI-PHI)/WT)
      IMAX=-1.0
      DO 30  JJ=1,12
      IMAX=IMAX+2.0
      CALL WAVFRM(IMAX,SCALE,PHI,WT,M,I)
      CALL ALGONE(IMAX,SCALE,WT,I,M+5,150,ERRVAL,DEV)
      DEVSUM=DEVSUM+DEV
      CALL EMAX(M+5,150,ERRVAL,ERRPOS,SMPPOS,ERRNEG,
     1 SMPNEG)
      IF(ERRPOS.GT.MAXPOS(LL))   GO TO 40
      IF(ERRNEG.GE.MAXNEG(LL))   GO TO 30
      MAXNEG(LL)=ERRNEG
      ANGNEG=PHI+(KK-1)*0.5236
      CURNEG=-1.0+JJ*2.0
      NNEG=SMPNEG
      DEVNEG(LL)=DEV
      GO TO 30
40    MAXPOS(LL)=ERRPOS
      ANGPOS=PHI+(KK-1)*0.5236
      CURPOS=-1.0+JJ*2.0
      NPOS=SMPPOS
      DEVPOS(LL)=DEV
```

APPENDIX

(continued)



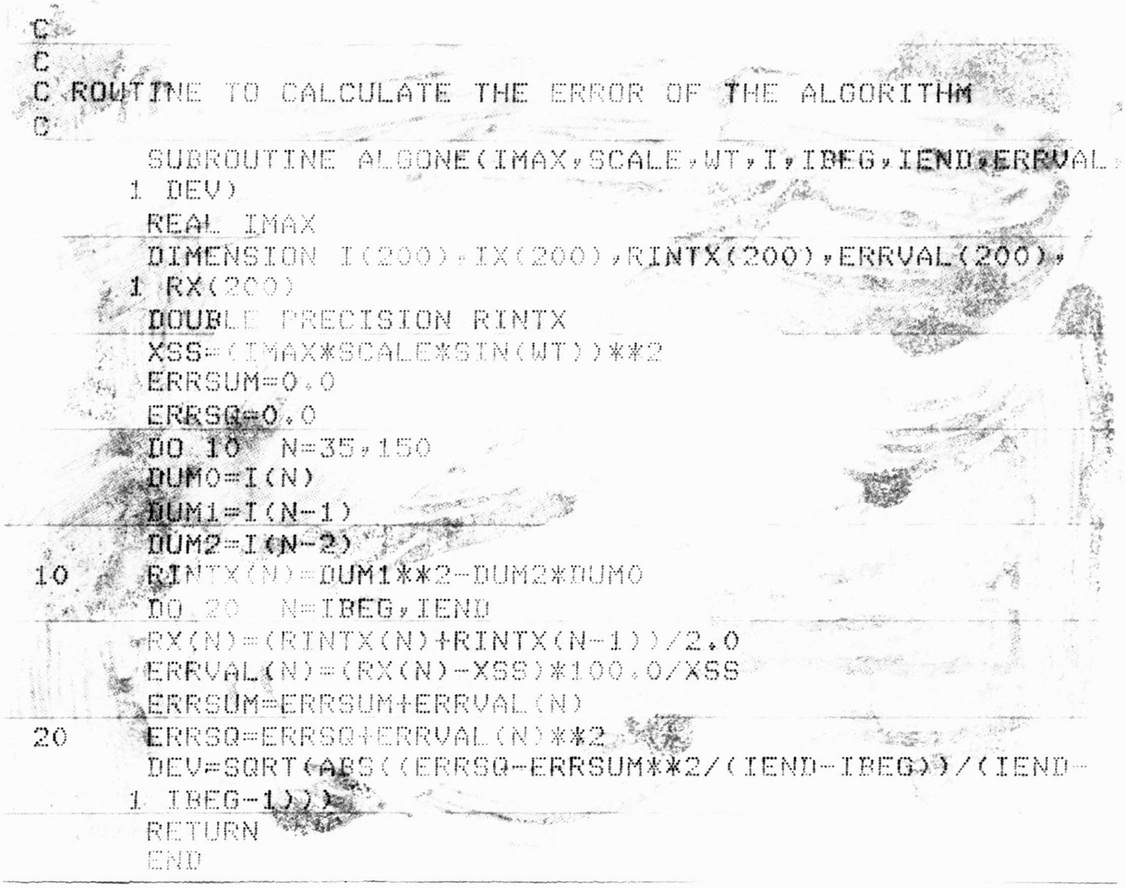
```
 30     CONTINUE
 20     CONTINUE
        AVGDEV(LL)=DEVSUM/144.0
        SMPLRT=0.91+(LL*0.1)
AN IV        V02.04      Mon 23-Apr-79 01:12:47

        TYPE 50,SMPLRT,MAXPOS(LL),ANGPOS,CURPOS,NPOS,
       1 DEVPOS(LL),AVGDEV(LL)
 50     FORMAT(1H0,F8.2,4X,F8.1,4X,F8.4,F8.1,2X,F8.0,
       1 F8.2,2X,F8.2)
        TYPE 60,MAXNEG(LL),ANGNEG,CURNEG,NNEG,DEVNEG(LL)
 60     FORMAT(1H ,12X,F8.1,4X,F8.4,F8.1,2X,F8.0,F8.2)
 10     CONTINUE
        REWIND 30
        WRITE(30) MAXPOS
        REWIND'31
        WRITE(31) MAXNEG
        REWIND 32
        WRITE(32) DEVPOS
        REWIND 33
        WRITE(33) DEVNEG
        REWIND 34
        WRITE(34) AVGDEV
        STOP
        END
1.
AN IV        V02.04      Mon 23-Apr-79 01:14:29

 C
 C
 C ROUTINE TO CALCULATE PRE- & POST-FAULT WAVEFORMS
 C
        SUBROUTINE WAVFRM(IMAX,SCALE,PHI,WT,M,I)
        DIMENSION I(200)
        REAL IMAX
        OFFSET=PHI-WT*40.0
        M1=M-1
        DO 10  N=30,M1
 10     I(N)=SCALE*SIN(WT*N-PHI+OFFSET)
        DO 20  N=M,150
        FRQERR=1.-N/10000.
 20     I(N)=IMAX*SCALE*SIN(FRQERR*WT*N-PHI+OFFSET+PSI)+
       1 .003*IMAX*SCALE*SIN(2*FRQERR*WT*N-PHI+OFFSET+PSI)+
       2 .005*IMAX*SCALE*SIN(3*FRQERR*WT*N-PHI+OFFSET+PSI)
        RETURN
        END
```

APPENDIX

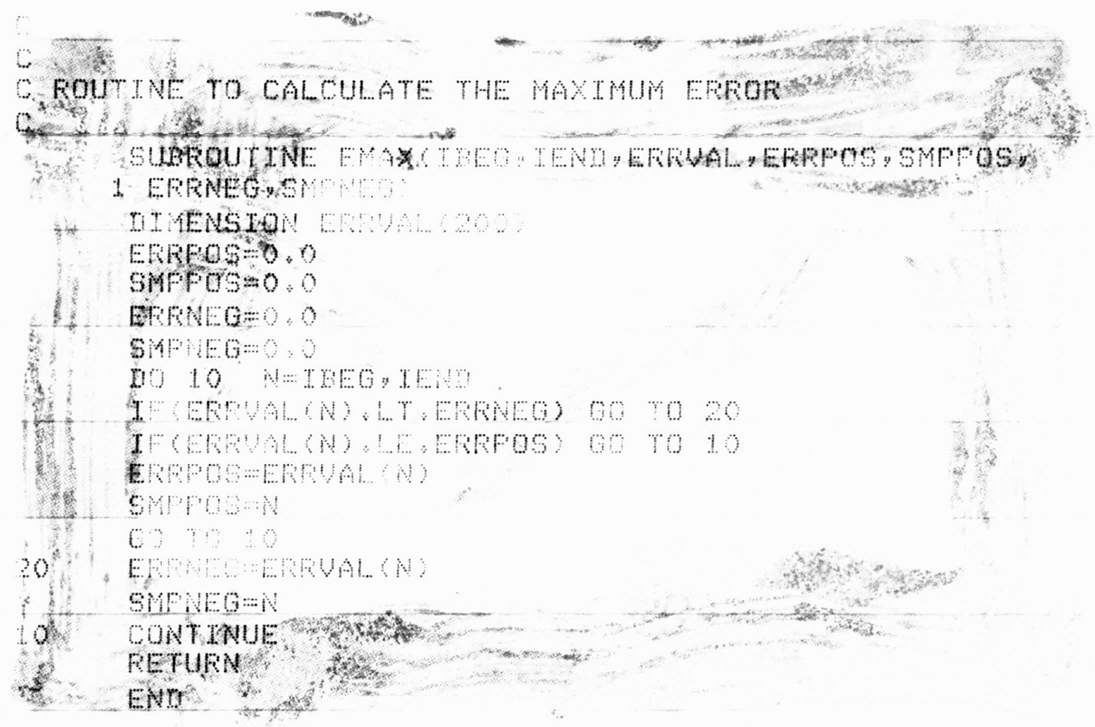
(continued)



```
C
C
C ROUTINE TO CALCULATE THE ERROR OF THE ALGORITHM
C
      SUBROUTINE ALGONE(IMAX,SCALE,WT,I,IBEG,IEND,ERRVAL,
     1 DEV)
      REAL IMAX
      DIMENSION I(200),IX(200),RINTX(200),ERRVAL(200),
     1 RX(200)
      DOUBLE PRECISION RINTX
      XSS=(IMAX*SCALE*SIN(WT))**2
      ERRSUM=0.0
      ERRSQ=0.0
      DO 10  N=35,150
      DUM0=I(N)
      DUM1=I(N-1)
      DUM2=I(N-2)
10    RINTX(N)=DUM1**2-DUM2*DUM0
      DO 20  N=IBEG,IEND
      RX(N)=(RINTX(N)+RINTX(N-1))/2.0
      ERRVAL(N)=(RX(N)-XSS)*100.0/XSS
      ERRSUM=ERRSUM+ERRVAL(N)
20    ERRSQ=ERRSQ+ERRVAL(N)**2
      DEV=SQRT(ABS((ERRSQ-ERRSUM**2/(IEND-IBEG))/(IEND-
     1 IBEG-1)))
      RETURN
      END
```

APPENDIX

(continued)



```
C
C
C ROUTINE TO CALCULATE THE MAXIMUM ERROR
C
      SUBROUTINE EMAX(IBEG,IEND,ERRVAL,ERRPOS,SMPPOS,
     1 ERRNEG,SMPNEG)
      DIMENSION ERRVAL(200)
      ERRPOS=0.0
      SMPPOS=0.0
      ERRNEG=0.0
      SMPNEG=0.0
      DO 10  N=IBEG,IEND
      IF(ERRVAL(N).LT.ERRNEG) GO TO 20
      IF(ERRVAL(N).LE.ERRPOS) GO TO 10
      ERRPOS=ERRVAL(N)
      SMPPOS=N
      GO TO 10
20    ERRNEG=ERRVAL(N)
      SMPNEG=N
10    CONTINUE
      RETURN
      END
```