MICROPROCESSOR-BASED PROTECTION RELAY FOR LARGE INDUCTION MOTORS

By

ALI MOHAMED KHIDER

Bachelor of Science
University of Baghdad
Baghdad, Iraq
1965

Master of Science
Oklahoma State University
Stillwater, Oklahoma
1982

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
May, 1989

MICROPROCESSOR-BASED PROTECTION RELAY
FOR LARGE INDUCTION MOTORS

Thesis Approved:

R. Ramakumar
Thesis Advisor

Wm. L. Hughes

[illegible]

J P Chandler

Norman N. Durham
Dean of the Graduate College

ACKNOWLEDGMENTS

I wish to express my sincere gratitude and appreciation to Professor R. Ramakumar, my major advisor, for his help and advisement during the course of this work.

I thank my doctoral committee other members, Dr. Wm. L. Hughes, Dr. D. D. Lingelbach, and Dr. J. P. Chandler for their support and assistance throughout this work. Also, I wish to express my gratitude to the faculty of Oklahoma State University School of Electrical and Computer Engineering.

Finally, I would like to thank my wife, daughter and my two sons, without their support and sacrifices this work would not have been possible. I wish to dedicate this work to my family and to the memory of my father and father in-law.

TABLE OF CONTENTS

LIST OF FIGURES

Figure		Page

NOMENCLATURE

A	reactance ratio = $\dfrac{X_r + X_c}{X_c}$
C_c	thermal capacitance of motor core
C_r	thermal capacitance of motor rotor
C_s	thermal capacitance of stator winding
I_a, I_b, I_c	phase currents
I_p	+ ve sequence motor current
I_n	- ve sequence motor current
I_{rp}	+ ve sequence rotor current
I_{rn}	- ve sequence rotor current
P_c	power dissipated in motor core
P_s	power dissipated in stator winding
P_r	power dissipated in rotor
R	+ ve sequence equivalent total resistance of the motor
R_c	motor core equivalent resistance
R_r	rotor resistance
$_lR_r$	locked rotor resistance
$_rR_r$	running rotor resistance
R_{rp}	+ ve sequence equivalent rotor resistance
R_{rn}	- ve sequence equivalent rotor resistance

R_s	stator resistance
R_{ce}	thermal resistance between motor core and the environment
R_i	thermal resistance between stator winding and core
R_{re}	thermal resistance between rotor and the environments
S	slip
T_c	core temperature
T_s	stator winding temperature
T_r	rotor temperature
V_p	phase to nutral voltage (+ ve sequence)
V_n	phase to nutral voltage (- ve sequence)
X_c	core equivalent reactance
X_s	stator reactance
X_r	rotor reactance
θ_p	angle between +ve sequence current and voltage

CHAPTER I

INTRODUCTION

Since the early days of the electric industry, induction motors have been used to convert electrical energy into mechanical torque to drive a variety of loads. With the proliferation of these drives in a wide variety of applications, it became increasingly clear that there is a need to protect these motors against abnormal operating conditions and modes of operations which could subject them and their drives to unacceptable safety hazards and dangers.

In the case of an induction motor, the protection engineer is faced with a dilemma in the process of designing a protection scheme, because while on one hand a closely protected drive will enhance the safe operation of the induction motor, it, on the other hand, could cause unwarranted trips and interruptions which could disrupt the overall process in which the motor is part of. This could happen when the motor is very closely protected and a small disturbance (in the supply side or in the load side) occurs, this turbulence could have passed without damage to the motor, but, due to the close protection scheme, causes unnecessary trip of the motor breaker.

A microprocessor-based protection scheme could be designed to overcome most of the shortcomings of electromagnetic or electronic protection schemes due to the enhanced accuracy and flexibility of the number of functions adapted in the protection scheme.

Background

Microprocessor-based digital computers have been applied to, or considered for, a variety of off-line and on-line tasks in the area of power system control, analysis, and operation. Off-line tasks include such things as load flows, fault studies, stability studies and relay coordination and settings. On-line applications include generation scheduling and power dispatching, supervisory control and data acquisition (SCADA) systems, sequence of events monitoring, sectionalizing, and load distribution and management.

The use of digital computers solely for the protection of power system equipment, however, is of relatively recent origin. The first serious proposal appeared in the late sixties (1). A great deal of research has been done since then, and presently considerable effort is being undertaken in this field, and already some positive indications have appeared on what the future of these systems will be. Protective devices which employ a microprocessor as the protective element have already reached the commercial market (2).

A microprocessor-based computer, for any application, have several inherent advantages that makes it attractive and provides incentives for further integration and assimilation in any system. It is always active and this is particularly desirable for relay application since it permits constant monitoring and self checking. It also has the ability to consolidate logical functions of many devices in one processor unit, thus possibly avoid dublication in situations where many separate pieces of equipment use identical inputs or perform similar functions. Finally, in an integrated station concept, there is potential for significant economic returns.

Computer relaying, on the other hand, is not inherently free of many of the problems that beset electromechanical or electronic solid-state relaying. Errors in the input signals caused by transients, dc offsets, and current transformer saturation must still be recognized and dealt with. The station environment, on which a lot has been done with regard to conventional devices, need to be evaluated in terms of computer technology requirements.

It is worthwhile to note that a majority of the papers written on this subject have mainly come from research efforts at universities and research centers. These tend to emphasize algorithms, software, and on models that can be tested on multipurpose minicomputers and on special purpose circuits and hardware that are laboratory oriented. However, the real world test installations covered in some of the publications have been the result of cooperation between utilities and manufacturers, and were primarily concerned with line protection using impedance sensing due to the amount of calculations required and the ability of the computer relay to carry out these calculations efficiently.

For the time being, computer relaying in the context of using a multitude of microprocessors tied together seems essential for acceptance of the technology by system protection engineers who, generally, oppose the concept of "putting all their eggs in one basket". Hence microprocessor protection of motors appears to be a logical starting point for the long path leading to a fully dedicated computer for all substation protection and control.

The application of computer relaying to line protection is well covered and documented in several publications (3,4). However, application to motor protection is not yet that well developed. This is mainly due to the cost of the hardware and software compared to the cost of the motor, and the complexity of the functions required for this type of protection.

In recent years very powerful microprocessors have been introduced commercially. Their costs are rapidly decreasing, leading to their economic viability for motor protection applications, especially in the case of large (> 1500 HP) motors.

Advantages of a microprocessor-based relay

The advantages of using a microprocessor for motor protection are enumerated below:

1. Can generate any protective function that a conventional relay can generate.
2. Can generate multiple protective functions simultaneously.
3. Improved dynamic characteristics of the relay.
4. Increased sensitivity.
5. Fits any load and motor characteristics.
6. Curve testing, plotting and graphics capabilities could be incorporated.
7. Continuous thermal memory of the motor is possible.
8. Data acquisition and display capability.
9. Easily extendable to include other functions that may become a necessity in future applications.
10. Potentially faster than a comparable electromagnetic relay.

Judging by the cost of present-day protective relays, the cost of a microprocessor-based relay could be tolerated only for large motor drives and drives which are, by the nature of their application, are essential for the operation of a plant and as such their down time is very expensive. The main element in the cost of a microprocessor-based relay is the software development cost, and this can be reduced by making the software general enough to be applicable to different types of motors and drives.

The problem to be solved

Provision of adequate protection for induction motors in any application and drive configuration is a problem solved by compromise. As the motor becomes larger in capacity, more costly and more agile relaying becomes necessary to prevent major equipment damage. Conversely, the larger units assume a greater proportion of the capacity of the system and any load reduction or trip-out due to incorrect operation of the motor protection relay becomes less tolerable.

The problem, in general, is to provide relaying that is sensitive enough to prevent damage yet not so sensitive as to cause false tripping.

With the increase in the size and mechanical output required of induction motors, and the requirement for them to operate at maximum efficiencies, the tolerance between their maximum thermal operating limits and safe operation are narrowing, thus requiring more accurate protection.

Algorithms for generating most of the functions of the type used in the protection of induction motors using digital computers have been developed and documented in several publications (5,6). However, these algorithms have not been developed specifically with the induction motor in mind, and they have not been tested for this particular application. As such, an evaluation and screening procedure is needed to test which of these function algorithms can be used for induction motor protection.

One protection scheme for induction motor rotor was suggested by Eliasen (7) using an impedance relay of the type used in line protection. A computer relay can perform the task of an impedance relay more accurately and efficiently. This concept of motor protection deserves further development and testing.

Another possible solution is the use of multifunction characteristics of the induction motor thermal limit curve pre-programmed into the computer relay memory. This entails using several functions to represent an exact replica of the thermal limit curve of an induction motor.

In this dissertation a microprocessor-based protection scheme will be developed for an induction motor based on the solution of the electrical, thermal, and mechanical models presented by Zocholl, Schweitzer, and Zegarra (8).

The testing and evaluation of the protection system is to be done by simulation using the mathematical models of the motor and drive.

Method of study

The induction motor is an electrical machine that converts its input electrical energy into rotary mechanical output that can be used to drive mechanical loads. The electrical energy is normally supplied to the stator of the induction motor which has the stator winding, which in turn transfers most of that energy to the rotating part of the motor through the air gap. This energy is in turn converted into mechanical torque, which is used to drive the mechanical load.

The digital relay will monitor the three phase-currents flowing into the motor and the phase voltage on one of the motor feeders. Figure 1 shows a typical arrangement of the digital relay. In order to enable the relay to incorporate an accurate thermal model of the motor, the ambient temperature of the motor environment is also monitored.

To properly protect an induction motor, each element in the power train shown in Figure 2 need to be protected against overheating during all phases of the operating cycle. Therefore, each and every one of these elements should be studied to determine the amount of energy dissipated and the temperature rise of that part of the

A microprocessor could greatly enhance the ability to represent the earlier mentioned functions used to protect induction motors and to solve the model equations to arrive at the temperatures of each part of the motor. The speed and accuracy of the relay could be controlled much more readily with a digital relay than with an electromechanical or an electronic relay. Hence, any desired level of safety factor or tolerance could being incorporated into the software design to achieve a close enough, yet trip-free operation.

The currents and voltage are preprocessed to eliminate any high frequency components present due to transients or pickups. The signals are sampled at 720 samples per second, which is becoming almost a standard sampling rate for digital protection. The computer will process the sampled data to extract the values of the variables used to protect the motor. This is performed by checking their sampled values against predetermined limits and set points, and then by using electrical and thermal models to estimate the temperatures of the various components of the motor. If any of these calculated values is found to exceed the thermal limit temperature of that component of the motor, a trip signal is initiated, which will then trip the motor circuit breaker and protect the motor from thermal damage.

Organization of the thesis

This thesis is organized into six chapters as follows:

1. Chapter 1 covers the introduction to this thesis, describes the problem on hand, together with the advantages of tackling this problem in the manner outlined.
2. Chapter 2 explains the functions currently used to protect induction motors together with explanations of each of these functions. Also the advantages and disadvantages of each method are outlined.
3. Chapter 3 outlines the mathematical models that the relay algorithm will solve to develop the protection scheme recommended.
4. Chapter 4 explains the design methodology for the digital protection relay.
5. Chapter 5 is an explanation of the performance considerations and the test and simulation results of the proposed relay.
6. Chapter 6 has a summary of the thesis and suggestions for further work.
7. The appendix will have the data of the motor used to simulate the relay operation and a listing of the digital relay programs

CHAPTER II

INDUCTION MOTOR PROTECTION

The following functions are recommended for induction motor protection and the actual functions used depend on the size of the motor, type of drive, and the economics of the plant or process this particular motor is a part of:

PROTECTION	FUNCTIONS
Overload	Time Overcurrent
Short Circuit	Instantaneous
Ground Fault	Ground Sensor
	Residual Overcurrent
	Zero Sequence
Stator Temperature	Temperature
Unbalanced Currents	Negative Sequence
Abnormal Voltage	Undervoltage
	Overvoltage
	Phase Sequence
	Negative Sequence
	Negative Sequence Overvoltage
	Residual Voltage

PROTECTION	FUNCTIONS
Differential	Self Balancing
	Percent Differential
Starting	Rotor Temperature
	Speed Sensor
	Impedance

Figure (3) shows a typical arrangement of relays in a large motor circuit (9).

Typical thermal limit curves commonly used to set the relay limits for an induction motor are shown in Figure 4 for 90% and 100% of rated line voltage.

While a typical overcurrent time relay will be able to protect the induction motor against overload, it is not always possible to protect the motor if it is required to make frequent starts or if it is to run on temporary phase unbalance and intermittent high inertia loading.

By using a protection scheme based on electrical and thermal models of the motor, it is possible to maintain a thermal memory of each of the motor parts, thus enabling the protection of the motor during all the phases of its operation cycle, including variable loads or supply conditions.

Time overcurrent function

Overload or overcurrent relays are employed to protect the stator of an induction motor since it is the most vulnerable motor element while the motor is running. This is in contrast to locked rotor or accelerating conditions during which the rotor is most vulnerable.

Three basic protection schemes may be used for overload protection. The first is a thermal replica relay (supplied as an element of the stator) which responds to motor input current. The second method employs direct measurement of stator winding temperature, typically by resistor temperature detection. Thermal replica relays employ a heating element and a temperature resistance element; the heating element responds to motor current, and the thermal time delay is designed to be long enough to avoid unnecessary shut down on short time overload peaks but short enough to avoid motor insulation damage. One disadvantage of thermal replica (current sensitive) relays is the lack of ability to sense motor overtemperature caused by loss of motor cooling such as blocked ventilation passages, high ambient temperature, etc. It is also impractical to match the thermal time constants of motors and relays for all the different motor ratings, characteristics, and operating conditions.

For large motors, direct winding measurement is a superior method of motor overload protection. On large motors with form wound coils, temperature detectors are embedded between the upper and lower coil sides in the stator slot. Today's virtually solid, void free insulation has good thermal conductivity and so the embedded temperature detectors respond quickly to actual motor winding temperature. This class of protection should more properly be called overtemperature relays rather than simply overload relays since they respond to overload plus ventilation loss and any other condition which could cause motor insulation overtemperature. Typically several winding temperature detectors are provided in large motors and the detector registering the highest temperature is connected to a relay arranged to shut down the motor when a potentially damaging temperature is reached. Controls could be arranged to set off an alarm at a lower temperature, or indicate on a temperature indicating instrument or recorder. Temperature trends are sometimes useful, particularly if there is a possibility of dirt build-up on the winding or in the ventilation passages.

To avoid possible unnecessary tripping that can occur with bimetal element based relays, and to take actual motor temperature into account, a third type of relay can be used. Its operation requires both high motor temperature and overcurrent before the breaker is tripped. Using a relay which responds to an 'and' operation, it can be seen that motor thermal capacity and ambient cooling air conditions are utilized to obtain full motor protection. This protection is particularly applicable for cyclic loads with high peak torques when there is little chance of losing ventilation through dirt build up, etc. In this case, if cyclic loading is involved, as long as the current is low when the resistance temperature detector becomes hot from heat transferred to it from the winding, the relay will not cause a trip and the motor will continue to operate producing maximum ventilation and rapid cooling.

Protective schemes employing various overload relay arrangements are shown in figure 5. Figure 5a illustrates the connection of a thermal element connected in the secondary circuit of a suitable current transformer. This is known as a replica type protective relay and is normally used on motors under 1500 Horsepower. Larger motors which normally have Resistance Temperature Detectors (RTD) embedded in the machine windings are protected by relays which apply the RTD's in a bridge circuit. Figure 5b operates when both high temperature and overcurrent are present while figure 5c operates on high temperature only.

Short circuit protection

Short circuit protection of the motor protects the power system more than the motor. A short in the motor or in the circuit feeding it can cause severe voltage dips in the power system which, in turn, could cause other loads to drop off the line or damage other equipment unless the fault is cleared promptly. Rapid clearing of motor or motor circuit indicates that some damage has already occurred, but the faster the fault is removed, the less costly in time and money will the repair be. This is particularly important for large motors which are often critical to plant production and where spares are frequently not available. Motor starting inrush currents usually set the lower limit for short circuit relays. Instantaneous relay elements can be used with switchgear type circuit breaker control since coordination is not required with any downstream devices. Further, setting the relay for instantaneous operation at as low a level as possible (just above starting inrush current) makes coordination with upstream relays very simple.

Ground fault protection

If the power system is grounded, the motor should be always protected with ground fault relays. Even in ungrounded systems, a sensitive ground fault relay will detect multiple ground faults (one on each of two phases on separate feeders) long before most such faults are detected by phase-overcurrent relays.

Several schemes are available for ground fault detection. The most sensitive one uses a large diameter current transformer (CT) that encircles the three motor leads. Any ground shield on the cable is carried around the outside of the CT. Short-circuit ground current flows through the CT to ground, returning external to the CT. All power currents, regardless of transients, starting inrush, etc., balance out at the ground detector CT. Even phase-to-phase short circuit currents balance out and are not detected. But small ground currents, even down to 5 or 10 A are readily picked up by the ground fault element which typically has a 0.5 A pickup. Instantaneous relaying is used since the motor is last in the line of relay coordination and it does have to override any other protective devices.

If phase current balance or negative sequence voltage relays are used to detect ground faults on all three phases, then only two overload relays are required. Ground fault relaying is therefore very economical for large motors with current or voltage unbalance protection. A total of 3 CT's and 3 overcurrent relays are required.

Figure 6 shows a typical arrangement for a zero sequence ground protection while Figure 7 shows a residual ground protection arrangement.

Stator temperature sensing

The most effective protection against thermal runaway is the use of temperature sensors on any part of the motor where temperature damage is expected. This is more easily done on the fixed parts of the motor i.e. stator and bearing. However, it is almost impossible to plant sensors in the rotor unless elaborate systems, with transmitters, remote power supplies, and pickups (which are inherently impractical on large machine due to their inaccessibility and vulnerability) are used.

Unbalanced currents

Once running, a three-phase ac motor will usually continue to operate if the power supply goes single-phase. Single-phase operation or even modest unbalance in currents are serious for large motors because of the severe rotor heating these events causes. Any unbalance causes a negative sequence air-gap flux, which, in turn, induces 120 Hz currents in the rotor. Just as under locked-rotor conditions, this high frequency current results in heating much greater than the larger, slip frequency (approximately 1 Hz or less), rotor power current. Even single-phasing is not detectable by most voltage relays since running motors act as generators so that even under single-phasing conditions, motor terminal voltages are nearly normal. Current balance relays give a positive indication of unbalance on single-phase operation.

The pickup on the relay is normally set between 5% and 20% of full load current and adjustable time delays ranging from 0.5 to 4 seconds are employed.

Typically, these relays are set to trip for unbalanced currents of 15% of motor full load. The time delay is selected to avoid tripping the motor feeder on an external fault.

An interesting bonus to the use of negative sequence unbalanced protection is the fact that it may also be used for ground-fault protection of the motor feeder circuit. This is because the negative and zero sequence networks are in series for a ground fault ($I_- = I_o$).

Superficially, it might seem that a residual relay is more sensitive since it receives $3I_o$. However, an analysis of the magnetizing currents and burdens reveals that negative-sequence and residual ground fault protection produce similar results. Economics favor the use of negative sequence protection for both unbalanced current and ground fault protection.

Undervoltage relay

If a voltage signal is available from a potential transformer either on the motor feeder or on the station bus, reduced voltage can be detected.

Reduced voltage jeopardizes starting since torque is proportional to voltage squared and if it occurs while the motor is running, it reduces pull-out or breakdown torque. It also causes increased stator current because a running motor acts like a constant KVA load (as a first approximation) near full speed. The undervoltage relay is set to prevent starting and running under reduced voltage conditions.

Differential relay

Except in motor circuits in which high speed reclosing or rapid transfer schemes are employed, the existence of a current in the circuit larger than the locked rotor current implies a fault in the motor or in its associated circuit. Since there is no reason to delay tripping for this value of current, an instantaneous element is applied to isolate the fault.

Instantaneous phase relays operate fast enough to respond to the current asymmetry that occurs during the first few cycles of inrush. For this reason, they must be set above the peak-asymmetrical inrush current. A setting of 1.4 to 2 times the symmetrical inrush value is usually employed. If high speed reclosing or transfer is applied, then a factor of 2 should be used. Faults can start at a current value lower than the setting of the instantaneous-phase relays and burn until the current increases sufficiently to be recognized.

Equipment damage for such faults can be minimized by the application of differential relays.

Figure 8 illustrates the application of conventional differential relays. The scheme requires a six-terminal motor, six current transformers, and three differential elements. The conventional relay uses CT ratios based on the rating of the motor and allows for discrepancies in CT characteristics. The relay will operate in about 1 1/2 to 6 cycles depending on whether a standard or high speed relay is applied and the magnitude of the fault.

A better differential scheme for motors is shown in figure 9. This is called self-balancing primary current differential scheme. It requires only three CT's and one three-element instantaneous overcurrent relay. The three CT's are normally mounted at the motor with the relay mounted in the switchgear. This scheme is sensitive and fast. It will detect all faults over 5 amperes and trip in approximately one cycle. The advantages of this scheme over the conventional scheme are that it is less expensive, faster, more sensitive, less complicated, has fewer CT's and is a much simpler relay.

The single disadvantage of this scheme is that it does not provide differential protection for the circuit between the motor and the switchgear.

Starting relays

Protecting the rotor of the induction motor during starting is a major issue in any protection scheme employed since the rotor is the most vulnerable part of the motor during starting. Several methods have been have been suggested and a brief summary of these methods is presented below.

Rotor temperature

This is the most direct and effective method for protecting the rotor and it is done by measuring the rotor temperature during the starting cycle, and a decision is made as to whether or not it exceeds a predetermined level. However, implanting sensors in the rotor and providing power supplies and pickups would severely weaken the inherent ruggedness of the rotor and hence of the motor.

Speed sensor

To alleviate some of the drawbacks of the protection scheme using temperature sensors, a shaft speed sensor can be installed to detect the speed of the motor. The output of his sensor is compared with predetermined benchmarks on the speed-time curve of the motor during starting.

While this method is very successful in detecting possible rotor damage due to locked or stalled rotor, it again requires an access to the motor shaft and its environment. Moreover, the speed-time curve will depend on the load being driven.

Impedance relay

An impedance relay can be used as a form of speed sensor since the impedance of the motor increases in a predetermined fashion during starting and this can be used to decide whether the motor starting is normal or not.

The relay can have a MHO characteristic, or a variation thereof. Current and potential can be supplied from current and potential transformers. Impedance relays are often accepted as being more reliable than mechanical speed switches or electronic speed sensing equipment, when evaluated on the basis of long term availability of the complete installation.

CHAPTER III

INDUCTION MOTOR MATHEMATICAL MODELS

The performance of an induction motor can be represented by a collection of electrical, thermal, and mechanical models.

Electrical model

The electrical model of an induction motor consists of two equivalent circuits, one for positive and the other for negative sequence components. These are shown in Figure 10. By solving these models and by making certain logical assumptions, the power dissipated in various motor components can be found. It is this loss that causes the temperature of that particular component to rise if the rate of generated energy is higher than the rate at which this energy is dissipated.

The model used for this application is based on the following assumptions:

1. The power supply is unbalanced, and therefore the negative sequence current needs to be considered as a major contributor to the rotor heating.

2. From the electrical equivalent circuits the

2. From the electrical equivalent circuits the total effective rotor resistance is represented by two resistances in series, one is the load-equivalent resistance, with a value that depends on the mechanical power delivered directly to the load, and the second is the rotor losses equivalent resistance. The resistance that represent the mechanical power output is slip dependent as shown in Figure 10. The dependence of the actual rotor resistance on slip is also included in the calculations.
3. In the electrical equivalent circuit ,the rotor reactance is shown as constant and not slip dependent; while this is not completely true, experiments have shown that the change is small enough as to not to cause a major error in the calculations (8).
4. Supply voltage variations need to be included in the model.

__Electrical model calculation steps__

From the phase voltage and the positive sequence components of the phase currents and the phase angle between them the resistive component of the motor circuit impedance can be found at any time as:

$$R = \frac{V_p}{I_p} \cos\theta_p \qquad (3.1)$$

The instantaneous value of the slip can be closely estimated using this resistance value as:

$$S = \frac{{}_rR_r}{A\,(\,R - {}_1R_r\,) - (\,{}_1R_r - {}_rR_r\,)} \qquad (3.2)$$

Where

$$A = \frac{X_r + X_c}{X_c} \qquad (3.3)$$

This slip formula assumes that the rotor resistance is linearly dependent on the slip, and that the rotor leakage reactance is not slip dependent as is evident from the motor electrical equivalent circuits.

Using the value of the slip calculated above, the value of the slip dependent positive and negative rotor resistance can be found as:

$$R_{rp} = (\,{}_1R_r - {}_rR_r\,)\,S + {}_rR_r \qquad (3.4)$$

$$R_{rn} = (\,{}_1R_r - {}_rR_r\,)\,(\,2 - S\,) + {}_rR_r \qquad (3.5)$$

By using the calculated values of the resistances in equations (3.4) and (3.5) and the values of the currents and voltage,the instantaneous power dissipated in each of the motor components can be evaluated using the standard power formulas:

$$P_s = 3\,(\,I_p^2 + I_n^2\,)\,R_s \qquad (3.6)$$

$$P_c = 3\,\frac{V_p^2}{R_c} \qquad (3.7)$$

$$P_r = 3 \left(I_{rp}^2 R_{rp} + I_{rn}^2 R_{rn} \right) \qquad (3.8)$$

These values of the dissipated power in the stator winding, motor core, and the rotor will be used as inputs to the thermal model.

Thermal model

Estimation of the temperature of each motor component as a function of time requires a thermal model for each one of these components, the solution of which provides the necessary answers.

Implanting thermal sensors at every point where the temperature need to be monitored (to make sure the temperature at these points do not exceed their thermal damage limits) is virtually impossible.

A distributed thermal model can be constructed, but the amount of calculations required could overwhelm the processor capability, but only with a small advantage over a lumped thermal model.

The lumped model for the stator and rotor of the motor are shown in Figure 11. The nodes in the thermal model represent thermally critical points in the motor.

The models assume a certain thermal capacitance for the winding, core, and rotor of the motor, and for all practical purposes the rotor is considered thermally insulated from the stator due to the very high thermal resistance of the air gap.

The models also assume that the thermal resistance representing the motor fan is linearly variable with the speed, hence slip, of the fan which has to be taken into consideration when calculating the heat dissipated to the motor environment.

<u>Thermal model calculation steps</u>

The lumped parameter thermal model is used to construct the model state equations.

The stator thermal model equations are:

$$\begin{bmatrix} \dfrac{dT_s}{dt} \\ \\ \dfrac{dT_c}{dt} \end{bmatrix} = \begin{bmatrix} -\dfrac{1}{R_i\ C_s} & \dfrac{1}{R_i\ C_s} \\ \\ \dfrac{1}{R_i\ C_c} & -\dfrac{R_i + R_{os}}{R_i\ R_{os}\ C_o} \end{bmatrix} \begin{bmatrix} T_s \\ \\ T_o \end{bmatrix} + \begin{bmatrix} \dfrac{P_s}{C_s} \\ \\ \dfrac{P_c}{C_c} \end{bmatrix} \qquad (3.9)$$

The thermal model state equation for the rotor is:

$$\frac{dT_r}{dt} = -\frac{1}{R_{rs}\ C_r} T_r + \frac{P_r}{C_r} \qquad (3.10)$$

Runge-Kutta 4th order method is used to solve the above equations, let

$$A_t = -\frac{1}{R_i\ C_s} \qquad (3.11)$$

$$B = \frac{1}{R_i\ C_s} \qquad (3.12)$$

$$C = \frac{1}{C_s} \qquad (3.13)$$

$$D = -\frac{R_i + R_{cs}}{R_i \, R_{cs} \, C_c} \qquad (3.14)$$

$$E = \frac{1}{C_c} \qquad (3.15)$$

The state equations become:

$$\frac{dT_s}{dt} = A_t \, T_s + B \, T_c + C \, P_s \qquad (3.16)$$

$$\frac{dT_c}{dt} = B \, T_s + D \, T_c + E \, P_c \qquad (3.17)$$

The Runge-Kutta 4th order discrete time form with a step size of H is:

$$T_s\,(k+1) = T_s\,(k) + 1/6 \,(\, K_{1s} + 2\,K_{2s} + 2\,K_{3s} + K_{4s} \,) \qquad (3.18)$$

$$T_c\,(k+1) = T_c\,(k) + 1/6 \,(\, K_{1c} + 2\,K_{2c} + 2\,K_{3c} + K_{4c} \,) \qquad (3.19)$$

Where

$$K_{1s} = H \,\{\, A_t \, T_s\,(k) + B \, T_c\,(k) + C \, P_s \,\} \qquad (3.20)$$

$$K_{1c} = H \,\{\, B \, T_s\,(k) + D \, T_c\,(k) + E \, P_c \,\} \qquad (3.21)$$

$$K_{2s} = H \,\{\, A_t \,(\, T_s\,(k) + 1/2\, K_{1s}) + B \,(\, T_c\,(k) + 1/2\, K_{1c}) + C \, P_s \,\} \qquad (3.22)$$

$$K_{2c} = H \,\{\, B \,(\, T_s\,(k) + 1/2\, K_{1s} \,) + D \,(\, T_c\,(k) + 1/2\, K_{1c}) + E \, P_c \,\} \qquad (3.23)$$

$$K_{3s} = H \,\{\, A_t \,(\, T_s\,(k) + 1/2\, K_{2s}) + B \,(\, T_c\,(k) + 1/2\, K_{2c}) + C \, P_s \,\} \qquad (3.24)$$

$$K_{3c} = H \,\{\, B \,(\, T_s\,(k) + 1/2\, K_{2s} \,) + D \,(\, T_c\,(k) + 1/2\, K_{2c}) + E \, P_c \,\} \qquad (3.25)$$

$$K_{4s} = H \,\{\, A_t \,(\, T_s\,(k) + K_{3s} \,) + B \,(\, T_c\,(k) + K_{3c} \,) + C \, P_s \,\} \qquad (3.26)$$

$$K_{4c} = H \,\{\, B \,(\, T_s\,(k) + K_{3s} \,) + D \,(\, T_c\,(k) + K_{3c} \,) + E \, P_c \,\} \qquad (3.27)$$

Similarly for the rotor state equation, let

$$F = -\frac{1}{R_{re}\ C_r} \qquad (3.28)$$

$$G = \frac{1}{C_r} \qquad (3.29)$$

then;

$$\frac{dT_r}{dT} = F\ T_r + G\ P_r \qquad (3.30)$$

and the Runge-Kutta 4th order time discrete form:

$$T_r\ (k+1) = T_r\ (k) + 1/6\ (\ K_{1r} + 2\ K_{2r} + 2\ K_{3r} + K_{4r}\ ) \qquad (3.31)$$

where;

$$K_{1r} = H\ \{\ T_r\ (k) + G\ P_r\ \} \qquad (3.32)$$

$$K_{2r} = H\ \{\ T_r\ (k) + 1/2\ K_{1r} + G\ P_r\ \} \qquad (3.33)$$

$$K_{3r} = H\ \{\ T_r\ (k) + 1/2\ K_{2r} + G\ P_r\ \} \qquad (3.34)$$

$$K_{4r} = H\ \{\ T_r\ (k) + K_{3r} + G\ P_r\ \} \qquad (3.35)$$

Mechanical model

A mechanical model for the motor is not required for most normal drive applications. However, it could become necessary if the running speed, hence the slip, of the motor is subjected to sudden variations due to the torque requirements of the load. For this reason, and in the context of the problem considered in this thesis, the mechanical model will not be covered.

To be able to mechanically model the motor, the load speed-torque characteristics should be available. This is not always the case, especially with certain types of loads and applications, for example metal rolling or rock crushing loads.

The mechanical model is based on knowing the difference between the torque developed by the motor and the load torque at all points over the speed range, this value being the accelerating torque of the motor. If the moments of inertia of the rotating component (including the load) are known the speed of the motor as a function of time can be estimated by solving the appropriate differential equations.

CHAPTER IV

DESIGN APPROACH

Study approach

After reviewing the computer relaying literature on methods used to protect power system components and the algorithms used in the modeling of various functions and calculation procedures, the following design approach was selected for microprocessor-based induction motor protection. The main elements of this design approach are enumerated below.

1. Since there are many commercially available analog to digital (A/D) interface cards (10) which are quite adequate for this type of application, an off-the-shelf interface can be used to sample the currents, voltages, and temperatures required by the protection algorithm. While this choice was made due to its availability and convenience, it is by no means the only interface that could be used in this application.
2. The signal processing algorithm that will extract the values of currents and voltages necessary from the sampled data is well covered in several publications and papers; the one most suitable for this application

due to its rapid convergence uses Kalman's filter algorithm (11). This filter will be assumed for the extraction of the currents and voltage signals. The ambient temperature signal need not use a filter algorithm due to the slow rate at which the temperature signal needs to be entered into the thermal model solution.

3. Using the values of currents and voltage, the positive and negative sequence components of the currents are calculated using well-known formulas. Next by using the method discussed in Chapter III, the power dissipated in the stator winding, stator core and the rotor are estimated.
4. Using the calculated powers as the driving vector in the thermal model and by solving that model using a Runge-Kutta fourth-order method, the temperature rises of the motor components are found.
5. By comparing the temperatures of the various components against their safe operating limits, a decision is made on whether or not a trip signal should be sent to the motor breaker to isolate the motor from the power supply.
6. If none of the stator winding or the rotor temperature was found to have exceeded its limits, the algorithm will repeat the loop and fetch the next phase voltage and line currents.

Relay hardware design

The main concerns in the selection of a microprocessor for an online application such as a digital protective relay are the processor word length and the average instruction processing speed of the processor. The two most common word lengths in commercially available microcomputers are 8 bits and 16 bits. The word length may be selected based upon the requirements of the relaying algorithm. The relaying algorithm requires the execution of digital filter equations on currents and voltage samples followed by a solution of the positive and negative sequence equivalent circuits and the dynamic equations of the thermal model. The value of the least significant bit (LSB) B of an 8 bit processor is 2E-8 = 0.004 while for a 16 bit processor it is 2E-16 = 0.000015. A linear filter transformation on N samples of data is subject to a maximum round-off error of (1/2) NB. If the digital filter equations require multiplications, some of which can be executed exactly in binary arithmetic, then the round-off error would be less than the estimate given above. If the computations of the filter equations is followed by a total number N_x of multiplications and divisions, the total maximum round-off error in the result would be $(1/2)(N + N_x) B$. Consider, for example, the computation of an impedance from six samples of current and voltage signals. The output of the filter equations is used in a complex division, which is equivalent to eight multiplications/divisions. This would lead to a round-off error of (1/2) (8 + 6) B = 7B. For an eight-bit processor, this error would be about 0.01 %. Considering the dynamic range of a current signal, a ratio of 20 may exist between near-fault current and the current for a fault at the end of the protected zone. Thus the B in current signal processing is likely to be greater by a factor of 20. In motor protection the maximum current which needs to be sampled does not exceed 7 times full load current; anything above that is considered a short circuit and hence tripped by the instantaneous function. Based on these considerations, it is clear that to maintain the algorithm within reasonable bounds, a sixteen-bit microprocessor would be essential. Similar considerations would lead to the selection of a suitable word length for any computational algorithm.

The second important factor in choosing a computer for any online application is the time taken to execute an instruction or its reciprocal, the number of instructions per second, expressed in million of instructions per second (MIPS). This factor mainly depends on how often the samples are taken, and how much delay is acceptable in the computer in making a decision on these samples.

In recent years the development of advanced microprocessors with more than 4 MIPS is very common, hence it is no longer necessary to worry about processor speed in relaying applications, especially in motor protection due to the relatively slow sampling rate and the simplicity of the calculations as compared to line impedance relaying applications.

Protective relay flow chart

Figure 12 shows a flow chart of the proposed protective relay. It consists of the following:

1. Three line currents and phase-to-neutral voltage are sampled at a rate of 720 samples per second. This sampling rate is based on multiples of the supply frequency, in this case 12 times 60 Hz.
2. Using a Kalman filter algorithm (11), the values of phase currents I_a, I_b, I_c, and V_p are extracted from the sampled data.
3. Check if any of these values is beyond acceptable range for these variables. At this stage the program will check for short circuits, overvoltages, and undervoltages. If any of these is detected, a trip signal is initiated and sent to the motor breaker.
4. Calculate the positive and negative sequence components of current from the three phase currents.

5. Using the positive and negative sequence equivalent circuits of the motor, calculate slip, using the formulas in the electrical model, and find the power dissipated in the stator winding, core and rotor of the induction motor.
6. The powers calculated in step 5 constitute the driving vectors to the thermal model of the motor. By using this model, and knowing the ambient temperature of the motor environment as a base, the temperature rise in each part of the motor is calculated using Runge-Kutta fourth-order method of solving the thermal model differential equations.
7. If any of the temperatures calculated in step 6 are above the recommended operating temperatures for that component of the motor, a trip signal is initiated to trip the motor breaker.
8. If the calculated temperatures are below the recommended temperatures, go to step 1 to sample new values of currents and voltage.

Computer software design

Two computer programs were written in Turbo Pascal, Revision 2.00B, the first program (Appendix C) generate motor currents and voltage data using the following inputs:

1. Motor parameters are read from a data file (Appendix A).

2. The percentage of the negative sequence current to the positive sequence current (entered manually).
3. The load, as percent of full load, the motor is driving (entered manually).

When these parameters are read and entered, the program will generate a continuous stream of data written to a file, this data consist of the following:

1. A binary 1 to indicate the data is ready, 0 if it is not ready.
2. The line to neutral voltage value in per unit and its phase angle in radians.
3. The three phase currents values in per unit and their phase angle in radians.

A sample of the data generated by the first program is attached to Appendix C.

The second program (Appendix B) is the motor protection program and it does the following.

1. Ask for the motor parameters data file, and once that is entered, it is read.
2. Read the voltage and currents entries generated by the first program.
3. The procedure OverVoltage determines if the phase voltage has exceeded a predetermined value, and trips the motor on overvoltage if that happens.
4. Checks for undervoltage in a procedure called UnderVoltage and trips the motor if that occurs.

5. Checks for overcurrent in all three phases using a procedure called OverCurrent and trips the motor if any of the three phase currents exceed six times the full load current.
6. Calculate the positive and negative sequence components of the currents using a procedure called SeqCalc.
7. Using the electrical model, the program calculates the power losses in the stator winding, stator core and the rotor.
8. The program will read the ambient temperature from a temperature file.
9. Using the thermal model in a procedure called ThemalModel, the temperatures of the stator winding, stator core and the rotor are calculated.
10. If the stator winding or the rotor temperature exceed their predetermined limits, the program will trip the motor.
11. If none of the above abnormalities was detected, the algorithm will go back and read the next voltage and currents values.

A typical sample of the data generated by the second program to draw the graphs in Figure 13 is included in Appendix B.

CHAPTER V

PERFORMANCE REVIEW

General performance considerations

In general any protective scheme for a power system will divide that system into several zones, each requiring its own group of relays. In motor protection , the motor is usually at the end of the power system, hence no performance penalty need to be paid if the motor breaker was made to trip as fast as required. This contrasts with other types of protection, for example transformer, transmission lines, or cables, in which faults and fault zones need to be coordinated so that only the faulty link is tripped.

In most relay applications, the three performance criteria listed below are commonly used to assess the quality of the protective system segment:

a. Reliability: It is defined as the ability of the relay to perform correctly when needed (dependability), and to avoid unnecessary operation (security).

Dependability is the certainty of correct operation in response to a system fault, while security is the ability of the system to avoid misoperation between faults.

Unfortunately, these two aspects of reliability tend to counter one another: increasing security tends to decrease dependability and vice versa. In general, however, modern relaying systems are highly reliable and provide a practical compromise between security and dependability.

In the case of a digital relay, reliability could be markedly improved due to the increasing reliability of digital components in general and due to the ability to increase or decrease sensitivity in close tolerances to match the motor or load characteristics.

Protective relay systems of any type must perform correctly under adverse system and environmental conditions.

The relays must perform accurately and dependably regardless of whether other systems have detected the failure or not; they must either operate in response to trouble in their assigned area or block correctly if the trouble is outside their designed area.

b. Speed: this is an important factor in relay performance because the faster the relay operates the more limited is the damage to the protected equipment and processes.

With the advent of some very fast microprocessors, it is no longer necessary to be concerned about the speed of the relay since it is very small compared to the time taken by the switching media (circuit breaker), which is typically of the order of two to four cycles (33 to 66 milliseconds), while the speed of the calculations required by the relay is in tens to hundreds of microseconds.

c. Selectivity: This is another important factor in relay performance. It is defined as the relay's ability to provide maximum service continuity with minimum system disconnection. In the case of induction motor protection, since the motor is typically at the end of the electrical power system, this is not a serious concern for the relay designer.

Test and simulation results

The digital relay algorithm developed was tested by simulating several hazardous operating conditions to find how and when the motor breaker tripped after the initiation of the fault.

The motor used for this simulation was rated at 18,000 Kilowatts with a full load starting time of 21 seconds and no load starting time of 12.6 seconds. The other motor parameters used in this simulation are tabulated in the motor1.dat file in Appendix B.

The microcomputer used was IBM PC compatible with

80286 microprocessor operating at clock frequency of 10 MHz with no mathematical coprocessor.

The speed of the computations could not be measured accurately since the time measurement algorithm can only have a maximum accuracy of 0.01 second, and since the trip signals are given much faster than the time it takes for the algorithm to write this information on the screen of the terminal.

Test # 1. Overvoltage: the overvoltage trip was set at 15% above the nominal operating voltage. The trip signal was initiated as soon as the voltage reached that trip level.

Test # 2. Undervoltage: the undervoltage trip was set at 80% of the nominal operating voltage. As expected the relay tripped accordingly.

Test # 3. Overcurrent protection was designed to trip the motor whenever the line current on any of the three phases reached or exceeded six times the full load current; this is based on the assumption that the motor power supply has a short circuit capability of much higher than six times the full load current of the motor. This is actually a power system design requirement for any reliable motor circuit. Otherwise the motor will not be able to start within its specified starting time and consequently within the rotor temperature limits, since the starting current value at full load for the motor being considered is 5.9 times full load current. Again, the motor trip signal was initiated when any one of the phase currents exceeded this value.

Overcurrent protection, in its various forms, i.e phase-to-ground, phase-to-phase, phase-to-phase-to-ground and three-phase shorts, is a major concern for the line protection system designer and not the motor protection designer. Therefore, if the line feeding the motor has an appropriately designed backup protection, protection against these failures is not of concern in the design of the motor protection.

The digital relay overcurrent or short circuit protection will provide protection against the following short circuit conditions:

a- Single-phase-to-ground, assuming the shorted line current value will exceed six times the full-load line current, which is usually the case in all large motor circuits.

b- Phase-to-phase short circuit currents are higher than phase-to-ground fault current, so the overcurrent protection will provide this type of protection.

c- Three-phase short is the most unlikely short to occur in motor circuits. However, the relay will protect the motor and the power circuit against this condition, as long as the fault current exceed six times the full-load current.

Test # 4. Thermal protection was tested in the following manner. The ambient temperature was set at 30 degrees Celsius, and the motor was simulated for normal starting with full mechanical load. The temperature of the stator reached 53° C, the rotor temperature reached 64° C, and when the motor was kept running with full load, the stator temperature remained almost steady at 53° C and the rotor temperature started to drop slightly till it settled down at around 61° C, as shown in Figure 13.

Test # 5. The next test was to find out at what level of unbalance the motor would trip during the starting cycle. After several trials with different unsymmetrical values of full load currents at starting, the rotor temperature did reach the trip level at 80° C when the percentage of negative sequence current component reached about 7.8 % of the phase current. In the meantime, the stator temperature remained well below the 100° C trip level at about 69° C as shown in Figure 14.

Test # 6. To test for locked rotor protection the motor input current was maintained at the locked rotor current value and the 80° C rotor trip level was reached after 14.1 seconds, while the stator temperature only reached about 68° C as shown in Figure 15.

Test # 7. Protection under repeated starting conditions:

Every time the motor is started the rotor temperature will be increased by an amount dependent on the energy losses in the rotor and the rate at which this energy is dissipated to the environment. When the motor is started from rest while its components are at 30° C, its rotor temperature rises to 64° C. If the motor was tripped for any reason, whether manually or automatically, and then restarted, the rotor temperature will reach about 84° C, and the motor will trip due to the rotor temperature limit of only 80° C. The minimum time between starts according to this model was found to be 11.6 minutes so that the rotor temperature would not exceed the 80° C limit for two consecutive starts as shown in Figure 16.

Test # 8. Another simulation test was made to find the limit on number of starts per hour. By employing the same model the number of starts per hour was found to be 2.78 starts per hour, which means that the motor can be started for the third time only after about 65 minutes from the first start, otherwise, and if the motor was started in less time than 65 minutes after the first start, the rotor temperature will reach the 80° C limit as shown in figure 17. The figure of 2.78 was obtained from the fact that the time between the first start and the third should be at least 65 minutes.

Test # 9. Another abnormality that could increase the rotor temperature beyond the recommended limit is the phase unbalance that could happen while the motor is running. The most severe manifestation of this phenomena is the loss of one phase in what is commonly called Single Phasing Operation. This could occur due to a single line voltage loss for any reason. In this case, and depending on the load on the motor, the rotor temperature will reach its critical limit in 44 seconds if the motor was running with no load and in about 12 seconds if the motor was fully loaded, as shown in Figure 18.

Test# 10. If an unbalance happens while the motor is driving its full mechanical load, the motor will trip if that unbalance reaches 10.8% and is sustained for 25.2 seconds, while the same unbalance at no load will not trip the motor. The motor will trip if the unbalance is sustained at 24.6% for 35 seconds under no load, as shown in figure 19.A severe case of unbalance could occur if any of the rotor cage conductors became loose and did not make good connection to the cage ring. The effect of this failure is to cause an unbalance in the three-phase currents. The relay will respond in a similar way no matter what the unbalance source is.

Test # 11. As the final test for overload protection, the was modeled to be running and had reached its stable stator temperature of 42° C when the load was then the load was increased to 1.2 times the full load. The stator temperature started to gradually increase till the trip level of 100° C was reached after running for almost 3.4 minutes at this load level. During this simulation test the rotor temperature level reached only 65° C which is well below the rotor trip level.

In conclusion, although only single fault or abnormality was considered at any one time, it is clear that this relay will protect the motor against a confluence of these as well.

CHAPTER VI

RESULTS AND DISCUSSION

Relay Testing Critique

It is clear from the testing results and the temperature profile that have been discussed in chapter 5 that the digital relay algorithm does work exceptionally well in protecting the motor against all the abnormal operating conditions as does a conventional relay. In particular, the digital relay will protect the motor much more closely against a combination of these abnormal conditions by closely modeling the motor thermally.

Another advantage is that any tolerances that is required to prevent unwarranted trips can be incorporated in this relay much more readily than in a conventional electromechanical or electronic relay.

The disadvantage of the proposed relay is that it relies for its operation on signals derived from current and voltage transformers that are inherently, susceptible to saturation and other types of errors. Even considering this the digital relay is superior to the conventional relay since the compensation for these errors requires only software modifications rather than hardware changes and mechanical adjustments.

As far as reliability and security of the digital relay as compared to those of conventional electromechanical or electronic elements, this depend largely on the quality and reliability of the digital components utilized in the relay circuits. It is well recognized that the quality and reliability of these components are far superior to other comparative system components such as mechanical devices.

Another very important consideration is the economics of the digital relay compared to the other types. Digital relays cost less than, or at least equal to, the cost of their electromechanical or electronic counterparts. Taking into consideration that the costs of electromechanical and electronic relays have been increasing, while at the same time and the costs of digital devices have been rapidly decreasing in the last decade, it appears that the break even level has already been reached. However, the one time cost of developing the software should be included in the case of the digital relay.

Scope for further work

1. A major enhancement of the relay accuracy and selectivity can be incorporated in the relay algorithm if the exact nature of the load and motor mechanical characteristics can be evaluated and modeled. This is

especially true when the motor is required to drive loads which are nonlinear in nature such as rock crushers or pumps for with changing mixtures of liquids and solids at different temperatures.

2. An improved fan model which takes into consideration the cooling properties of the fan as a function of the motor speed could also enhance the accuracy of the thermal models of the motor components. This becomes especially important when the motor is driving nonlinear loads at varying speeds.
3. Another possible improvement in the thermal model is the inclusion of air flow measuring devices which could estimate more accurately the values of the thermal resistances. This is especially important if the motor is required to operate in environments with heavy airborne particles that could block the cooling ducts of the motor, thus reducing their effectiveness. Inclusion of these devices will not only improve the accuracy of the thermal model but they could also indicate when these cooling ducts are to be cleaned.
4. The algorithm could be made to monitor the Resistance Temperature Detectors normally embedded in the stator winding and a trip signal is generated when an abnormal temperature is detected.
5. Several other functions can be added to this relay to enhance its operation, these include:

a. Zero sequence detector to protect against small current ground faults.

b. Voltage unbalance function, this protection function requires as input three voltage signals and hence, three voltage transformers, one on each phase.

c. Differential protection could also be added to the proposed protection scheme.

Summary and discussions

The purpose of all protective relay designs is to protect the system or certain components against a variety of hazards and abnormal operating conditions and to limit the financial and operational liabilities in case of motor or system failure.

In the case of a large induction motor, digital protection is far superior to other electromagnetic or electronic systems currently used since the digital relay is much more flexible than the other types, the only limitation being the availability of the hardware and software specifically designed for this application.

With the increased sophistication of digital systems, this approach is readily flexible and expandable to take advantage of any future developments in system integration and speed.

The proposed digital relay can protect the motor under all operating conditions the conventional relay can protect it from. The relay will use the same input signals used to drive an electromechanical or electronic relay. The algorithm will calculate the operating conditions of the motor and will trip it only if it detects an abnormality that could cause damage to the motor or its drive.

The models used are more accurate than conventional current traces used in other motor protection relays. If a tolerance need to be included in the protection scheme.The proposed protection relay can incorporate it in a very accurate and controlled manner that will not inadvertently jeopardize the other protection functions.

In conclusion a microprocessor-based digital protection relay for induction motor protection represents a major improvement over conventional electromechanical and electronic schemes. However, at the present time, it can be justified only for large motors (> 1500 HP) and for motors in critical installations.

REFERENCES

(1) G. D. Rockefeller, "Fault Protection with Digital Computers", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-88, No. 4, April 1969, pp. 438-461.

(2) "Type MCGG, Overcurrent Relay for Phase and Earth Faults", GEC Pub. R-6002A, GEC Measurements, Stafford, England.

(3) S. H. Horowitz, "Protective Relaying of Power Systems", IEEE Press, Selected Reprints Service, 1980.

(4) IEEE Tutorial Course on Computer Relaying, Pub. no. 79 EH0148-7-PWR, 1979.

(5) B. J. Mann and I. F. Morrison, "Digital Calculations of Impedance for Transmission Line Protection", Trans. IEEE Power Apparatus and Systems, vol. PAS-90, No. 1, Jan./Feb. 1979, pp. 270-279.

(6) M. A. Rahman and P. K. Dash, "Fast Algorithm for Digital Protection of Power Transformer", IEEE proc. vol. 129, March 1982.

(7) A. N. Eliasen, "The Protection of High Inertia Drive Motors During Abnormal Starting Conditions", Trans. IEEE. Power Apparatus and Systems, Vol. PAS-99, No. 4, July/Aug. 1980, pp. 1483-1492.

(8) S. E. Zocholl, E. O. Schweitzer III and A. Aliaga-Zegarra, "Thermal Protection of Induction Motors Enhanced by Interactive Electrical and Thermal Models", Trans. IEEE.Power Apparatus and Systems, Vol. PAS-103, No. 7, July 1984, pp. 1749-1755.

(9) C. R. Olson and R. D. Valentine, "Protective Relaying for Large AC Motors", IEEE. Petroleum and Chemical Industry Conf. PCIC, 1976.

(10) DASH-8, Data Acquisition and Control Interface Board, Metra-Byte, 440 Myles Standish Blvd., Taunton, MA 02780, 1987.

(11) M. S. Sachdev, H. C. Wood and N. G. Johnson, "Kalman Filtering applied to Power System Measurements for Relaying", IEEE Transactions on Power Apparatus and Systems, Vol. PAS 104, No. 12, Dec. 1985.

(12) Martiny, W. J., McCoy, R. M., Margolis, H. B. "Thermal Relationships in an Induction Motor Under Normal and Ubnormal Operation.", AIEE Transactions, PAS, Presented at AIEE Winter General Meeting, New York, NY, Jan. 31- Feb. 5, 1960.

(13) Y. V. V. S. Murty and W. J. Smolinski, "Design and Implementation of a Digital Differential Relay for a 3-Phase Power Transformer Based on Kalman Filtering Theory", IEEE Transactions on Power Delivery, pp. 525-533, April 1988.

(14) D. J. Fink, H. W. Beaty, Standard Handbook for Electrical Engineers. Eleventh Edition, McGraw-Hill, New York, NY, 1978.

(15) John R. Rice, Numerical Methods Software and Analysis. McGraw-Hill, New York, NY, 1983.

(16) A. R. Bergen, Power System Analysis. Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

(17) D. S. Stivison, Introduction to Turbo Pascal, Sybex Inc., Berkeley, CA, 1985.

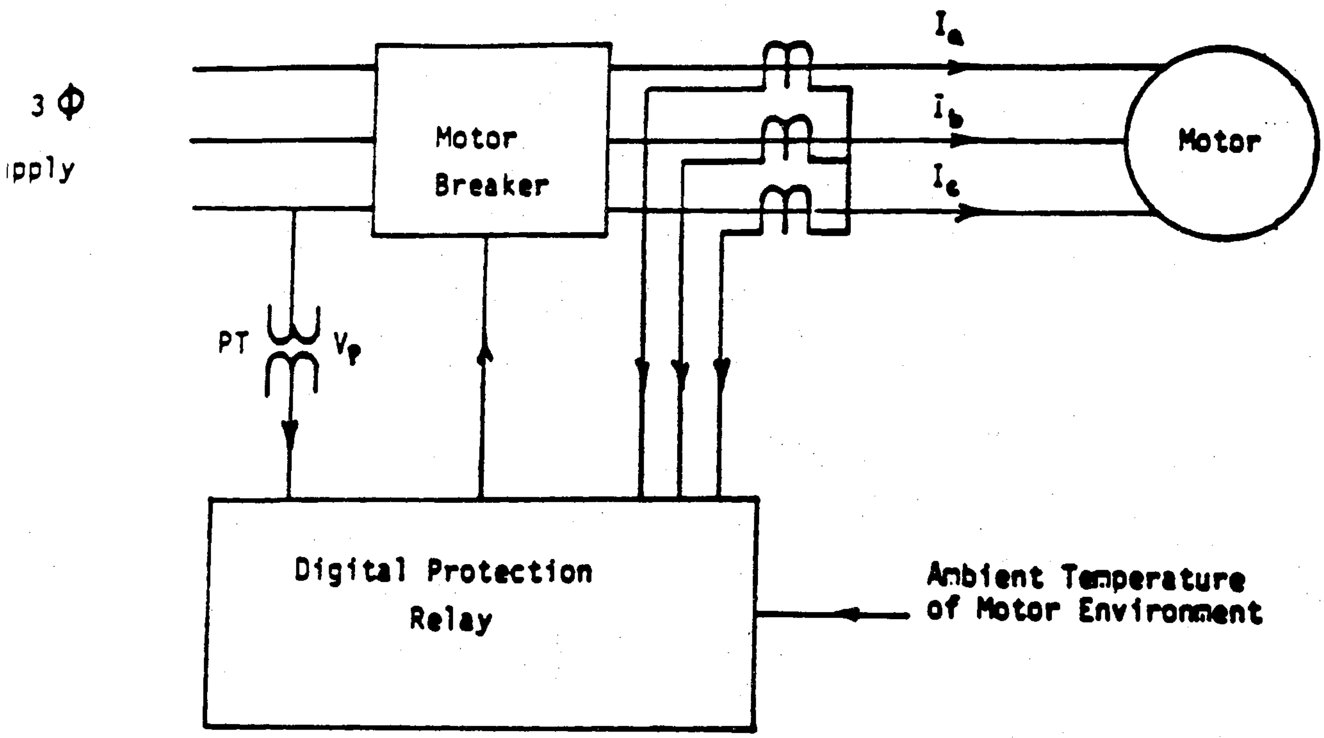


Figure 1 Simple block diagram of the digital protection scheme for an induction motor.

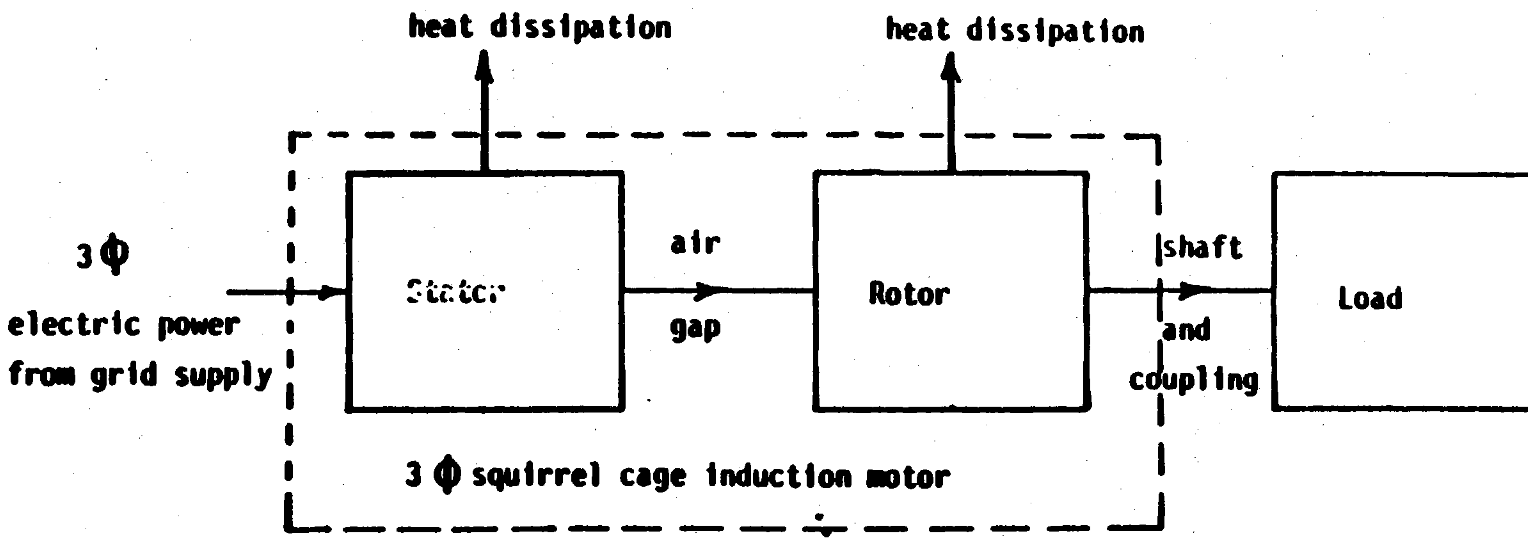
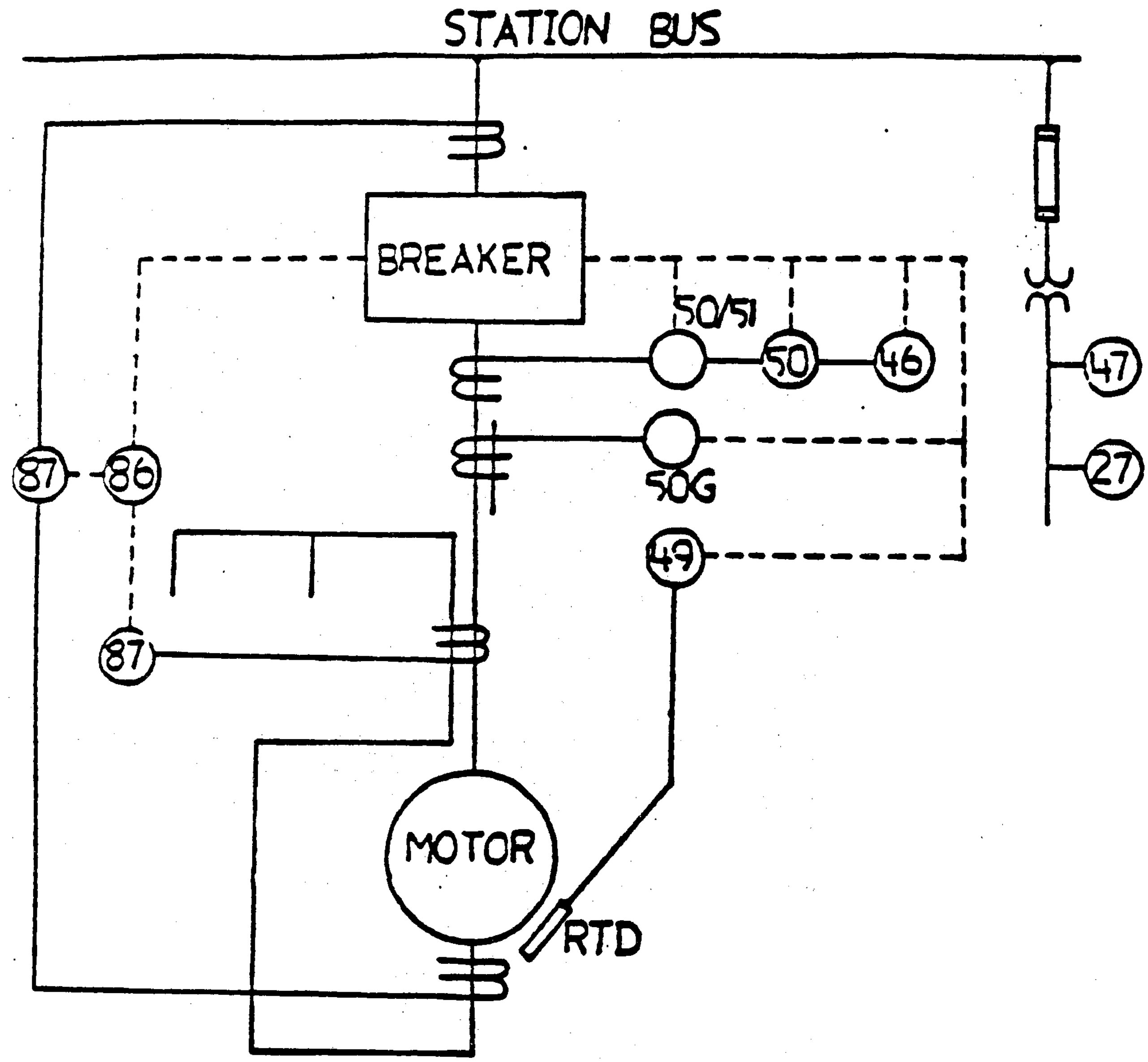


Figure 2 Power train of a typical induction motor.



NOTE:

* The above numbers are based on a system adopted as standard for automatic switchgear by IEEE, and incorporated in American Standard C37.2-1970.

Figure 3

Recommended Induction Motor Protection

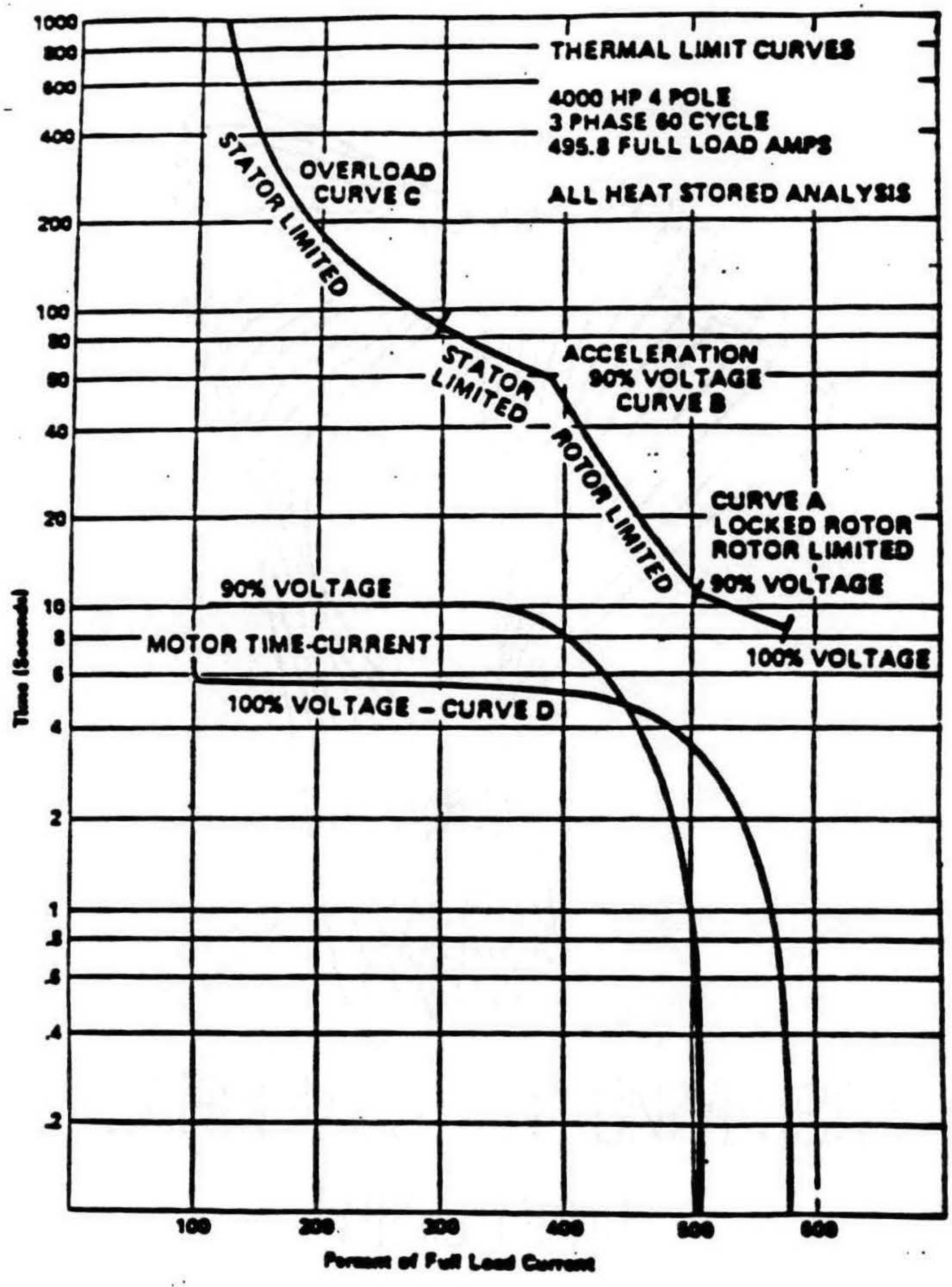


Figure 4

Motor Thermal and Time-Current Starting Curves

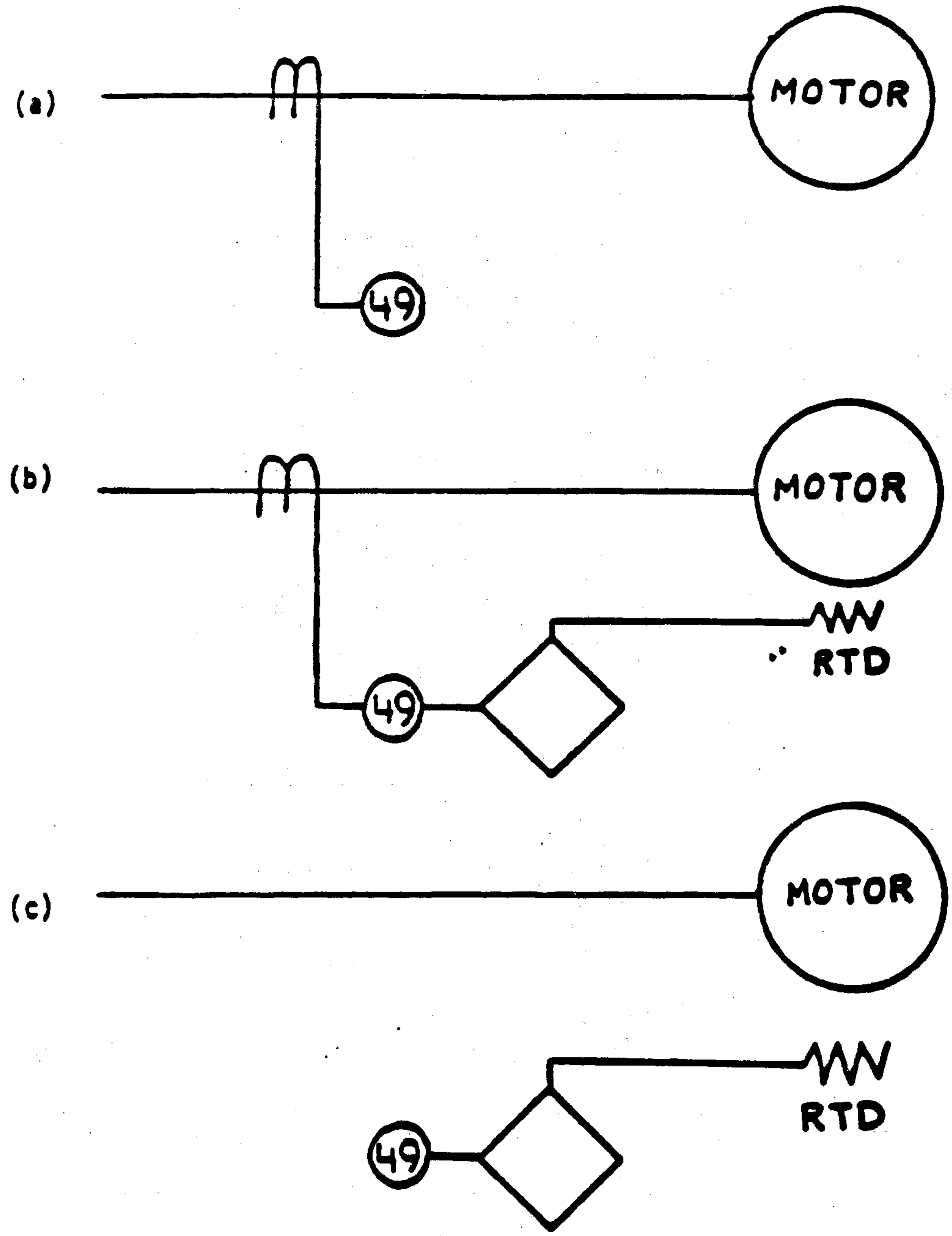


Figure 5

Thermal Overload Protection

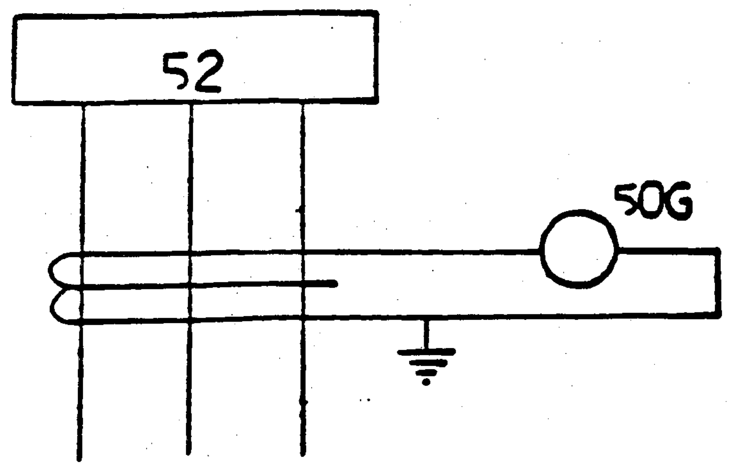


Figure 6

Zero Sequence Ground Relaying

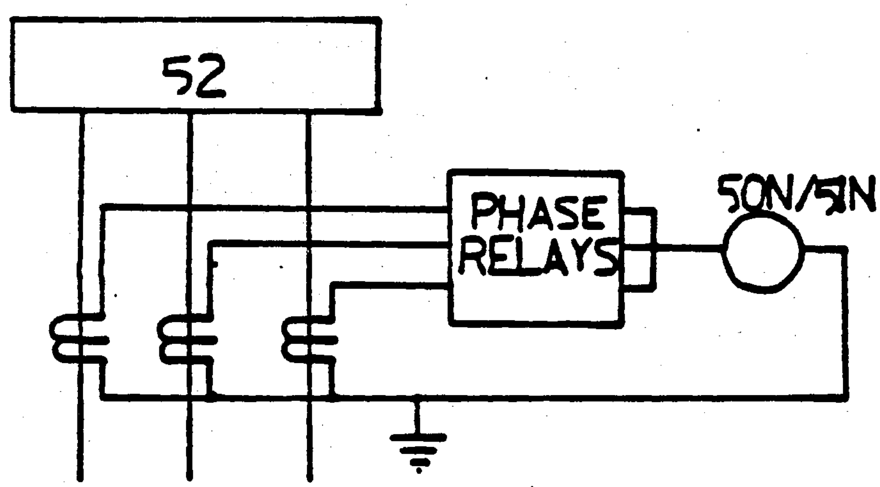


Figure 7

Residual Ground Relaying

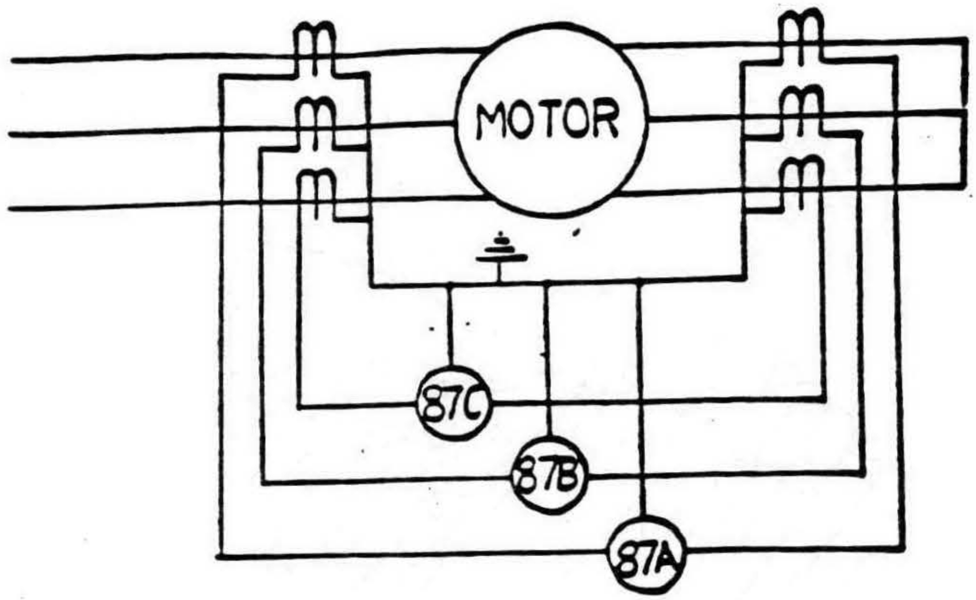


Figure 8

Percentage Differential Relaying

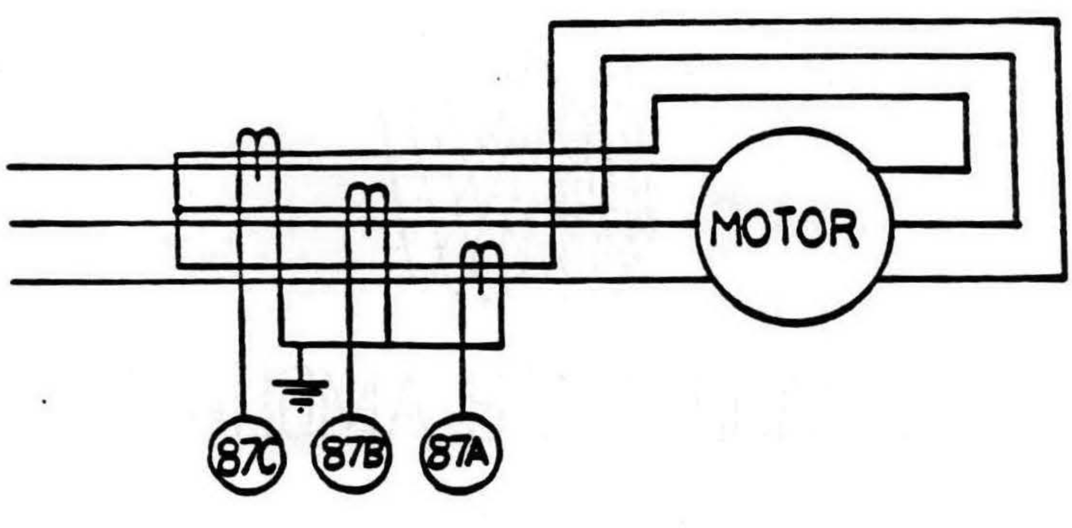
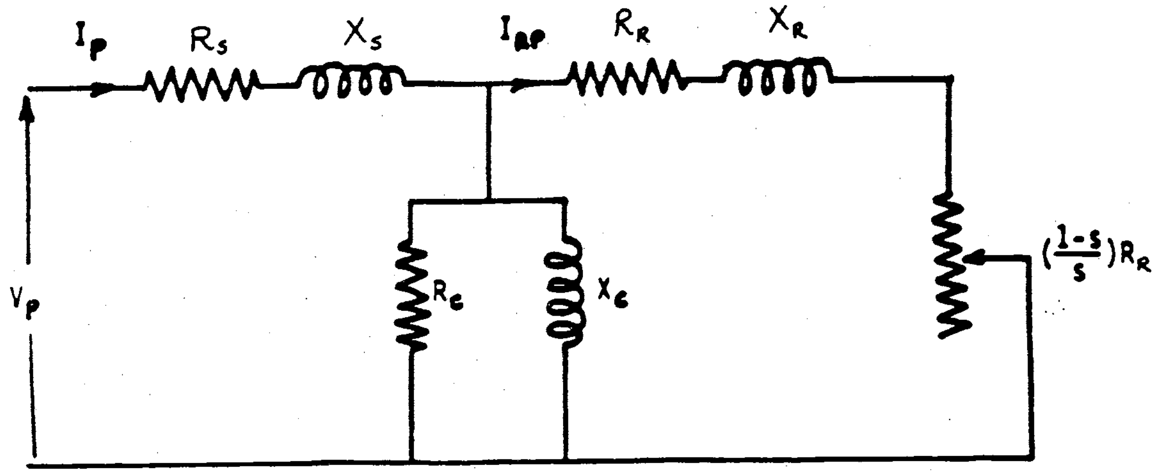
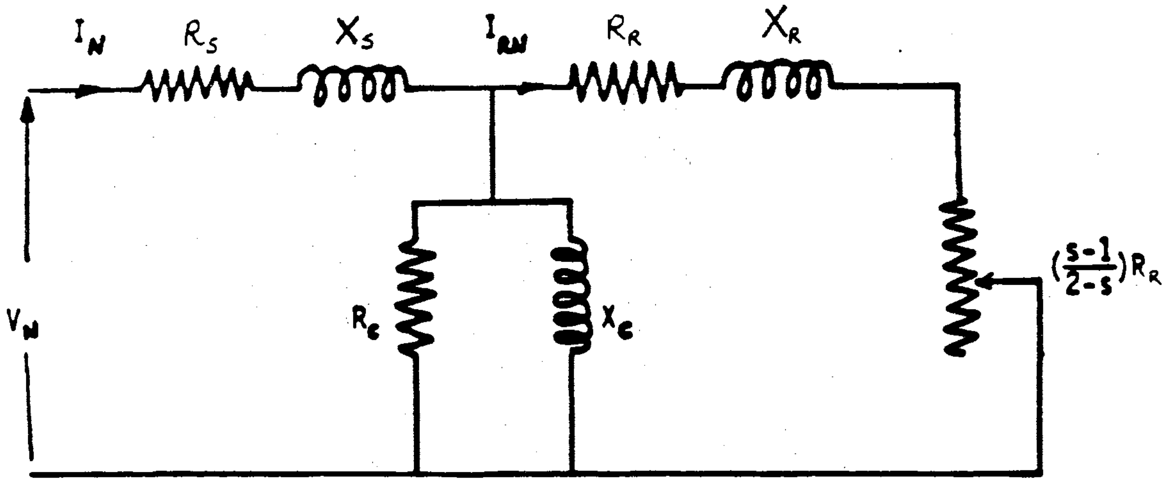


Figure 9

Self Balancing Primary Differential Relaying



(a)

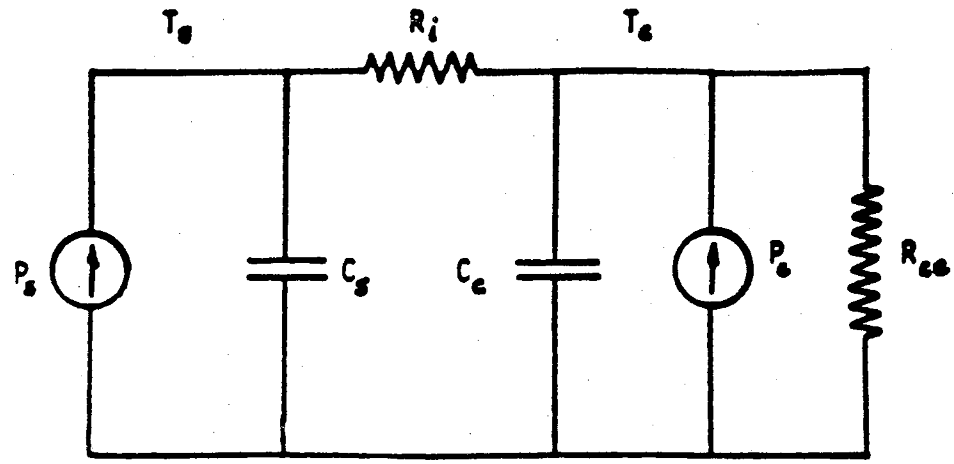


(b)

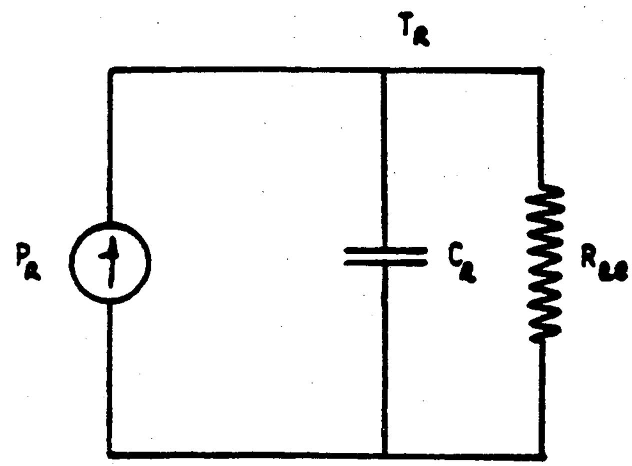
Figure 10 Induction Motor Equivalent Circuits

(a) Positive Sequence

(b) Negative Sequence



(a) Stator



(b) Rotor

Figure 11 Induction Motor Thermal Model

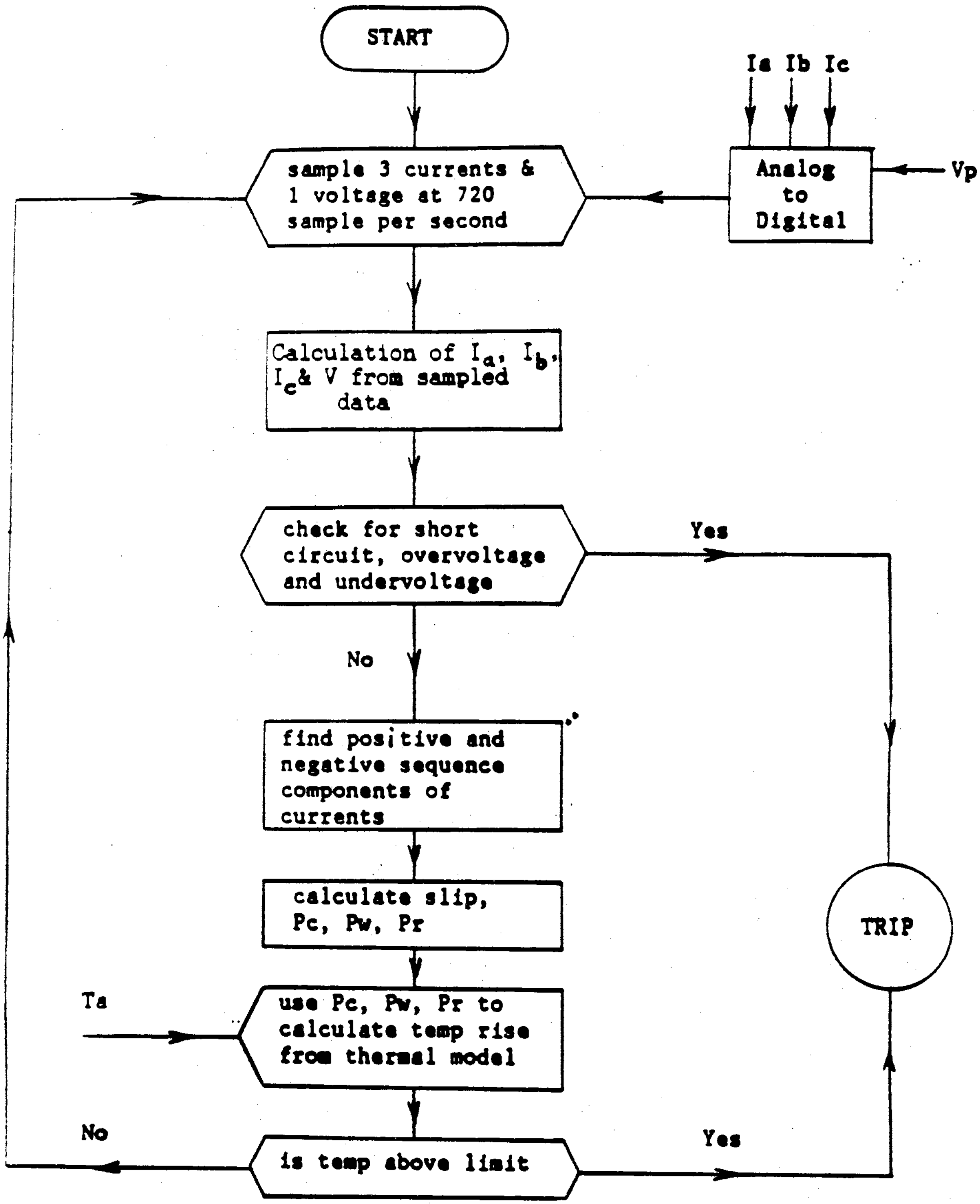


Figure (12)

Flow chart of induction motor digital protection relay

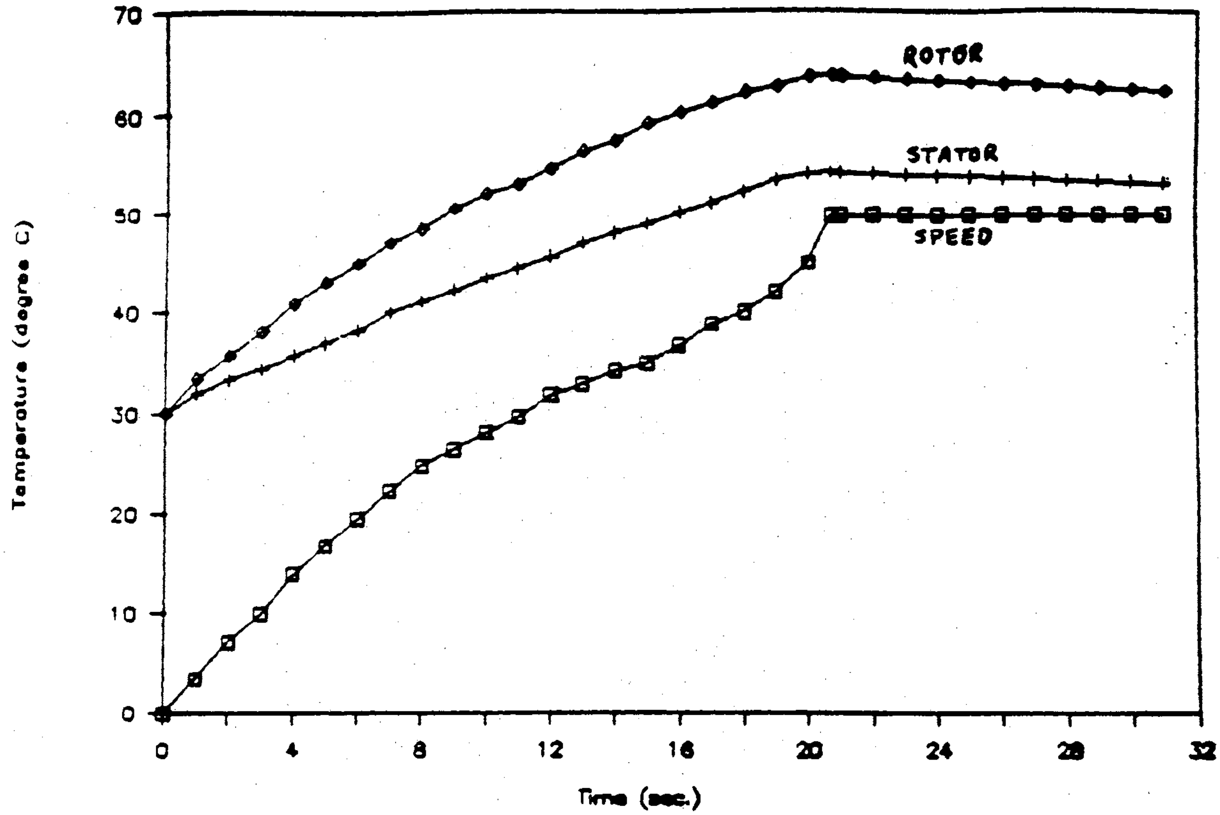


Figure 13

Temperature rise with full load

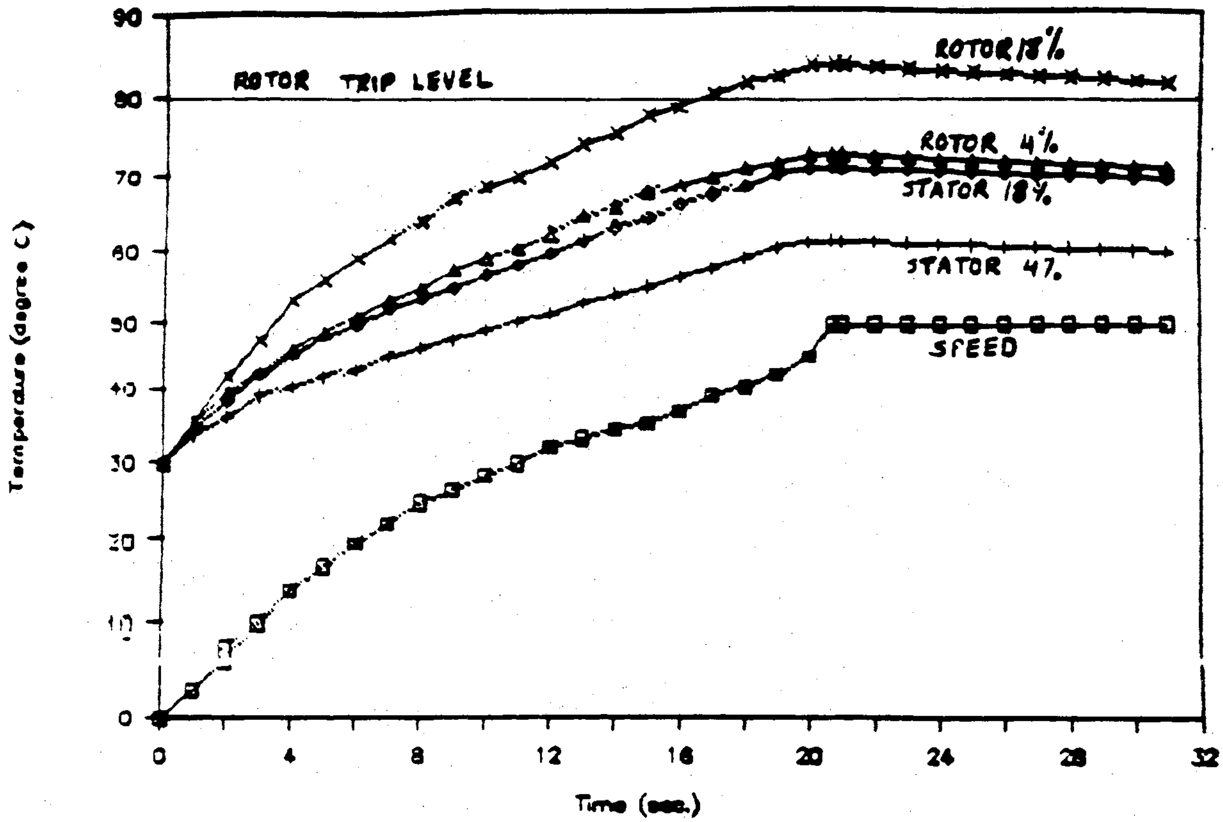


Figure 14

Temperature rise while starting with unbalanced currents and with full load

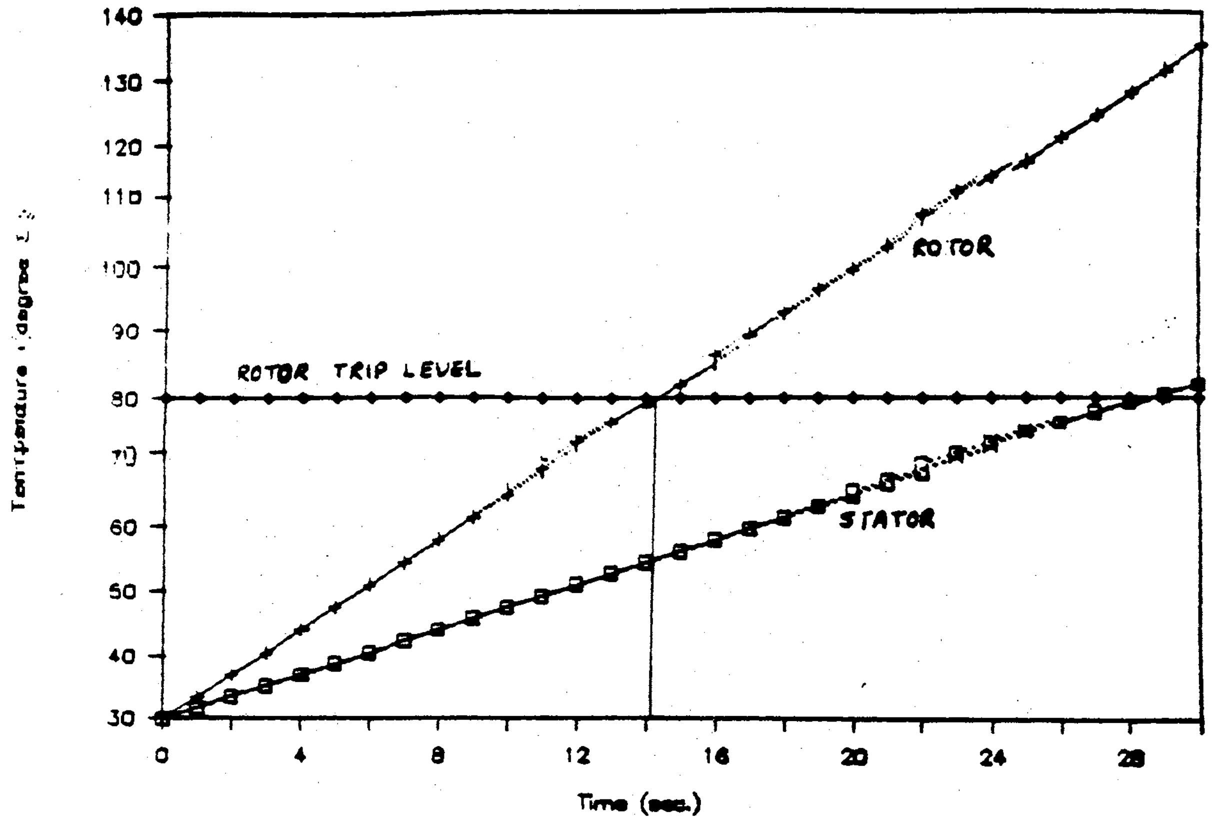


Figure 15

Locked rotor temperature rise

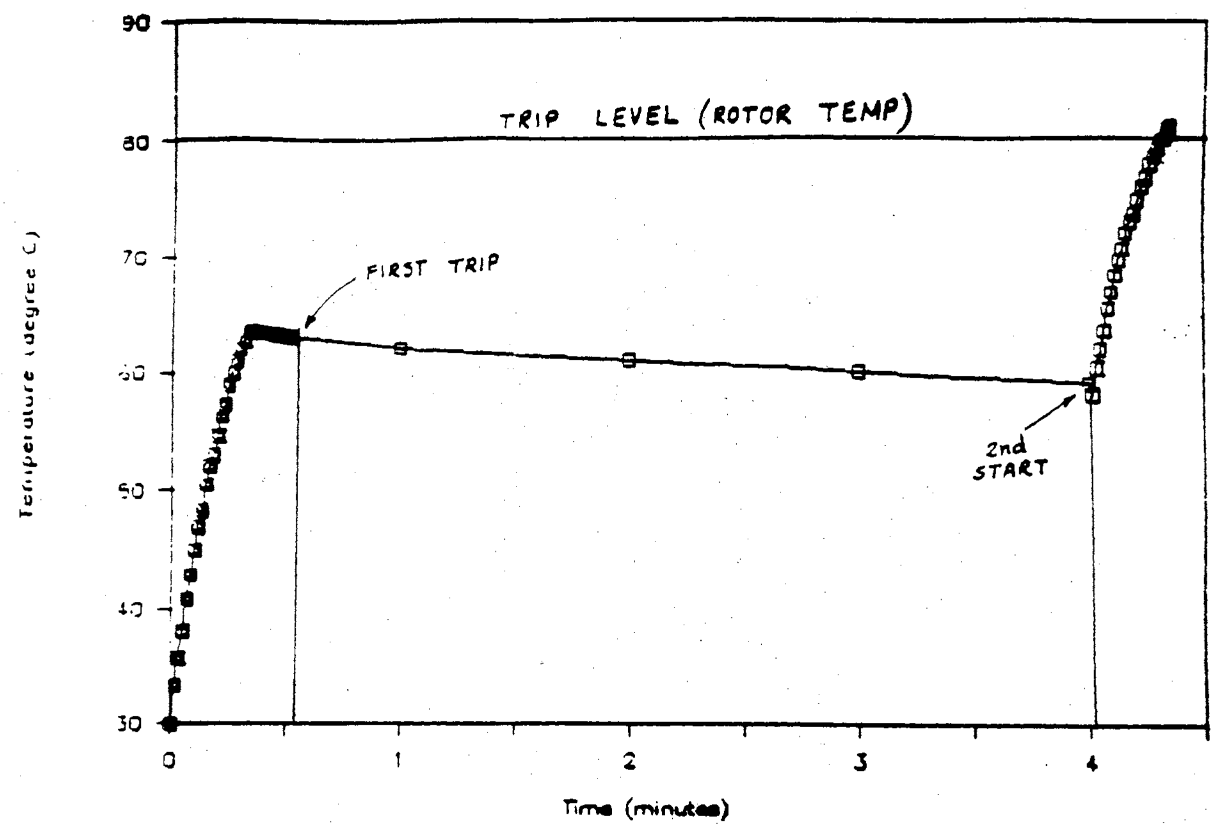


Figure 16

Rotor temperature profile during two starts 4 minutes apart

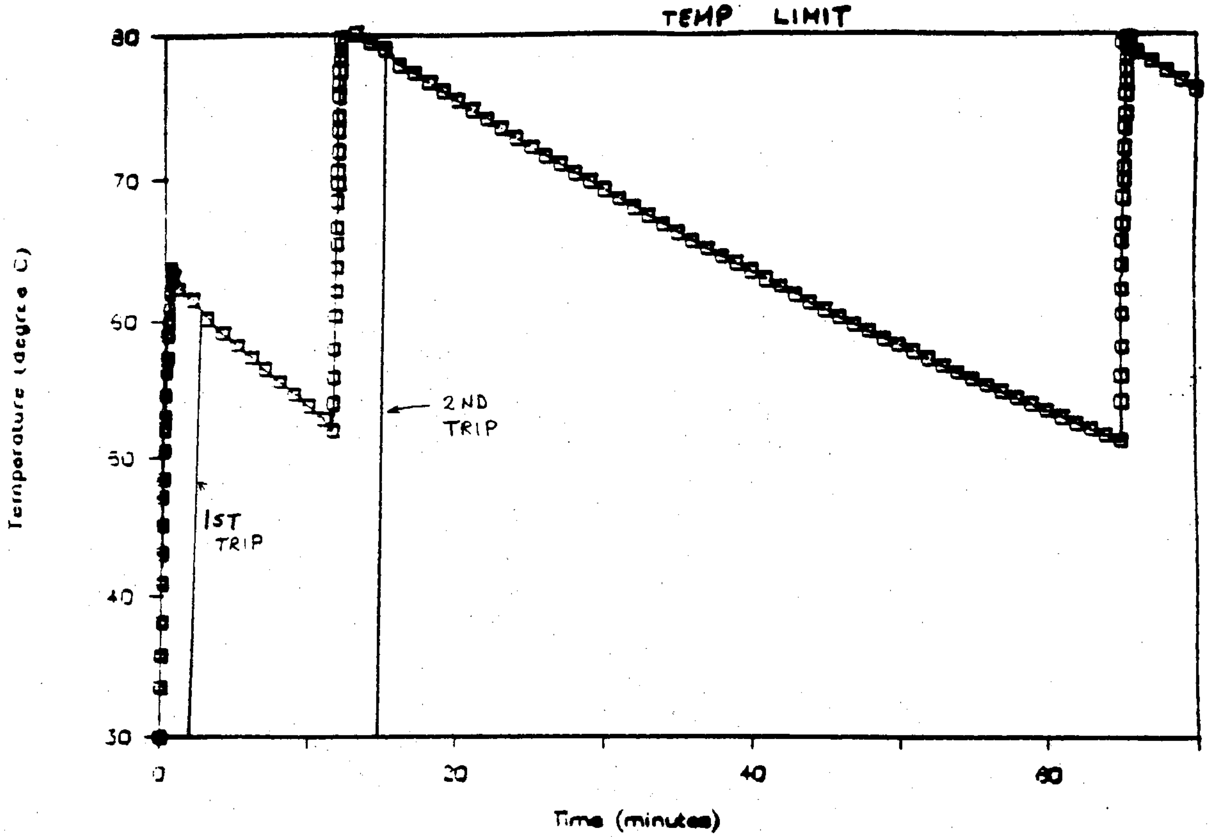


Figure 17

Rotor Temperature Profile
during Three starts in 65 minutes

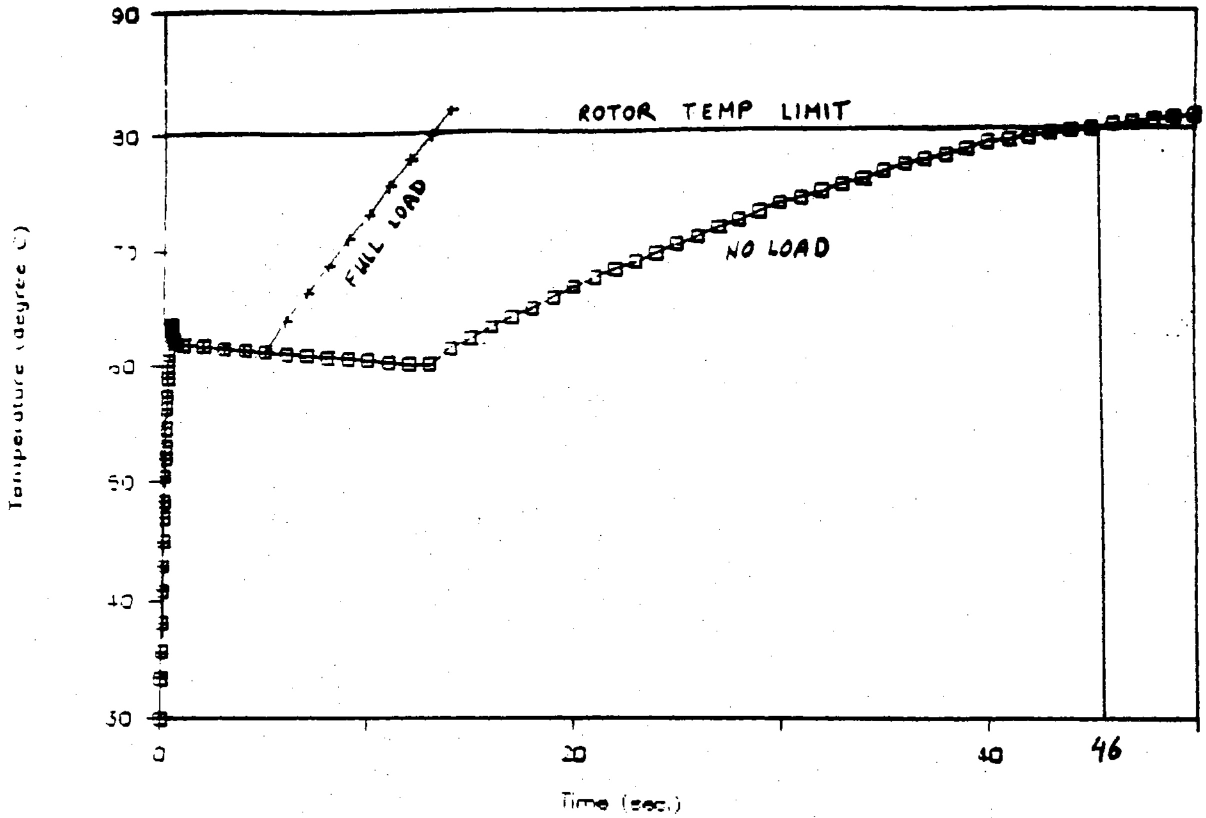


Figure 18

Single Phasing effect on Rotor Temperature

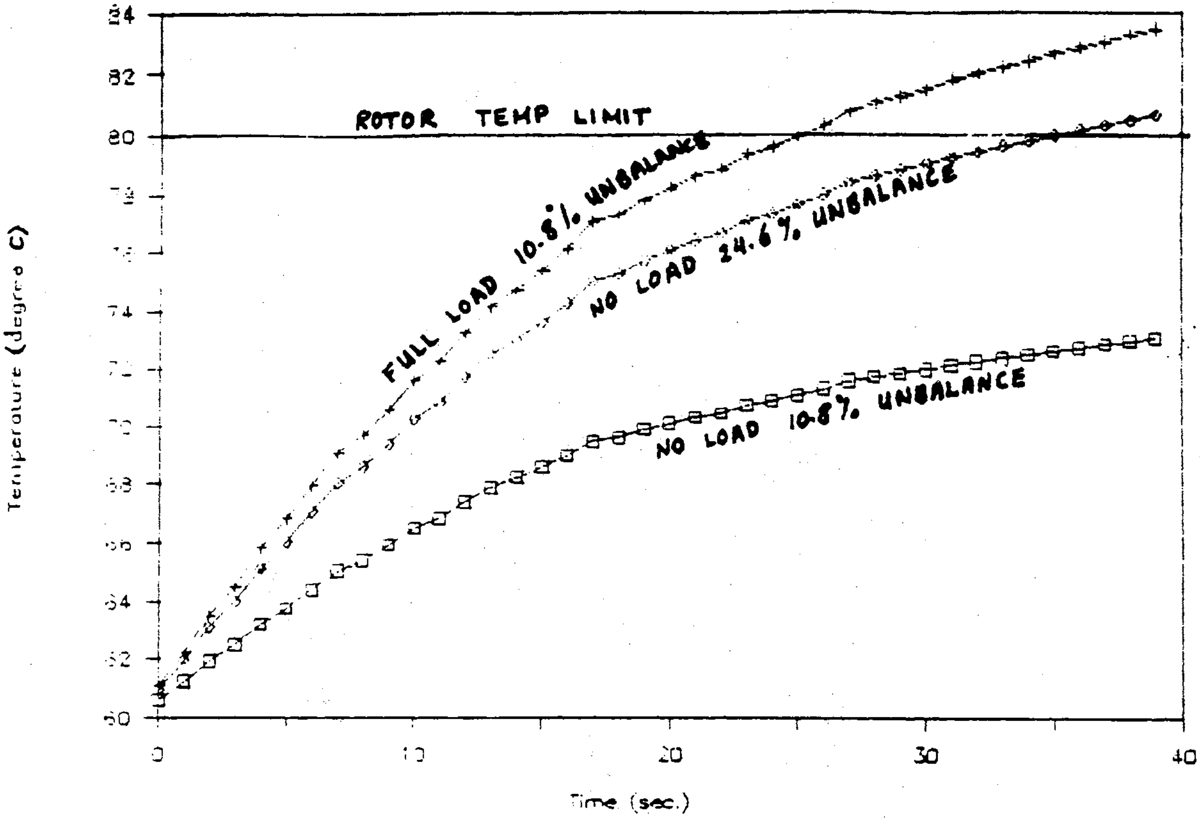


Figure 19

Rotor Temperature at various Unbalances while Running

APPENDIXES

APPENDIX A

INDUCTION MOTOR DATA

18.		{ Motor power in Megawatts}
0.00422	0.0977	{ R3 and X3, stator resistance and reactance, p. u.}
237.0	4.9	{ R6 and X6, magnetizing resistance and reactance, p. u.}
0.01982	0.00722	{ LR1 and RR0, locked and running rotor resistance, p. u.}
0.0696	0.0977	{ LX1 and RX0, locked and running rotor reactance, p. u.}
6.0	0.002	{ short circuit current limit & Gnd current limit, p. u.}
0.7	1.15	{ minimum and maximum voltage allowed, p. u.}
20.		{ Ri, thermal resistance between winding and stator}
20.	6.67	{ LRce,RRce, locked and running thermal R between stator and ambient}
7.5	75.	{ Cs,Cc, thermal capacitance of winding and stator}
130.	43.9	{ LRre,RRre, locked and running thermal R between rotor and ambient}
4.68		{ Cr, thermal capacitance of rotor}
100.	80.	{ limit temperatures of stator and rotor}

APPENDIX B

PROTECTION PROGRAM LISTING

```
PROGRAM Motor_Protection;
{     THIS PROGRAM WILL PERFORM THE FUNCTION OF PROTECTING
}

{     AN INDUCTION MOTOR USING AN INTERACTIVE ELECTRICAL
}
{     AND THERMAL MODELS TO DEDUCE THE TEMPERATURE OF THE
}
{     MOTOR COMPONENTS (STATOR, WINDING AND ROTOR)
}
{                                                    Rev   D
}
{    WRITTEN BY: ALI KHIDER,                         Feb. 7,1989
}
{                P. O. BOX 71071,
}
{                SUNNYVALE, CA 94086
}

CONST
  PI =   3.1415926536; { PI }
  HLF =  0.5; { ONE HALF }
  TRD =  0.333333333; { ONE THIRD }
  RT2 =  0.866025403; { ROOT THREE OVER TWO }
  PO2 =  1.570796327; { PI OVER TWO}
  TPO3 = 2.094395102; { 120 DEGREES IN RADIANS}
  FPO3 = 4.188790205; { 240 DEGREES IN RADIANS}
TYPE
 POLAR = RECORD
   val : real;
   ang : real;
end;

TYPE
 COMPLEX = RECORD
   rel : real;
   img : real;
 end;
```

```
VAR
  InFile,VIFile,AmTFile : TEXT;
  Inzie : STRING[14];
  Hour,Min,Sec,Frac : INTEGER;
  avail,CNT,K : INTEGER;
  Ia,Ib,Ic,Vpn : POLAR;
  I1,I2,I3 : POLAR;
  Ipos,Ineg,Izero : POLAR;
  MZ : POLAR;
  Irp,Irn : REAL;
  Rs,Xs,Rc,Xc,LRr,RRr,LXr,RXr : REAL;

  Rrp,Xrp,Rrn,Xrn :REAL;
  MR,SLIP,MPower,A,PSS: REAL;
  Ri,RRce,LRce,Cs,Cc,RRre,LRre,Cr,LTS,LTR: REAL;
  Rce,Rre : REAL;
  A1,B,C,D,E,F,G,H1 : REAL;
  Ts,Tc,Tr,Tamb : REAL;
  Vmax,Vmin,RVpn:REAL;
  II : INTEGER;
  Ps,Pr,Pc : REAL;
  IL,IOL: REAL; {MAX LOCKED ROTOR CURRENT & MAX GRND
CURRENT}
  Tindex :INTEGER;{TRIP INDEX}
PROCEDURE AMBIANT ( VAR Tamb : REAL);
      { Procedure to read the ambient temperature }
 BEGIN
   ASSIGN ( AmTFile,'AmTFile.DAT');
      {AmTFile is the file which has the ambient
temperature}
   RESET ( AmTFile);
   READLN ( AmTFile, Tamb);
      {Read ambient temperature}
   CLOSE ( AmTFile);
 END;
PROCEDURE VIRead;
 {Procedure to read currents and voltage information}
 BEGIN
   ASSIGN ( VIFile,'VIFile.DAT');
      {VIFile is the file which has the currents and
voltages}
   RESET ( VIFile);
 END;

PROCEDURE Initialize;
 BEGIN
   WRITE (' What is the name of the file which contain the
motor data ?  ');
   READLN (Inzie);
   ASSIGN (InFile,Inzie);
     {INFILE is the file that has the motor parameters}
```

```
   RESET ( InFile);
   READLN ( InFile, MPower);

   READLN ( Infile,Rs,Xs);
     {Read stator resistance and stator reactance}
   READLN  ( Infile,Rc,Xc);
     {Read resistance representing core losses}
   READLN ( InFile,LRr,RRr);
     {Read locked and running rotor resistance}
   READLN ( InFile,LXr,RXr);
     {Read locked and running rotor reactance}
   READLN ( InFile, IL, IOL);
     {Read maximum locked current permisable and max ground
current
     permisable}
   READLN ( InFile,Vmin,Vmax);
     {Read minimum and maximum allowable voltages}
   READLN (InFile, Ri);
            {Read thermal resistance between winding and
stator}
   READLN (InFile, LRce,RRce);
            {Read locked and running thermal resistance
between
             stator and ambient}
   READLN (InFile, Cs,Cc);
            {Read therrmal capacitance of winding and
stator}
   READLN (InFile, LRre,RRre);
            {Read locked and running thermal resistance
between
            rotor and ambiant}
   READLN (InFile, Cr);
            {Read thermal capacitance of rotor}
   READLN (InFile, LTS,LTR);
            {Read limit temperatures for stator and rotor}
   CLOSE (InFile);
   AMBIANT ( Tamb ); {Read ambient temperature}
   Xrp := LXr; {Initialize locked rotor reactance}
   Ts := Tamb;
   Tc := Tamb;
   Tr := Tamb;
   VIRead;
   CNT :=  0;
 END;

PROCEDURE CompPolar(InC : COMPLEX; VAR OutP :POLAR);
 {Procedure that convert vectors from Complex to Polar}
 BEGIN
   OutP.val := SQRT ( InC.rel*InC.rel + InC.img * InC.img
);
   OutP.ang := ARCTAN ( InC.img / InC.rel );
```

```
   IF InC.rel < 0.0 THEN OutP.ang := PI + OutP.ang;
 END;
PROCEDURE ZAdd( X,Y :  POLAR; VAR Z1: POLAR);
 {Procedure that inputs two vectors in Polar form and
output
  their sum in polar form also}
 VAR
  TempC : COMPLEX;
 BEGIN
  TempC.img := X.val*SIN(X.ang)+Y.val*SIN(Y.ang);
        {Imaginary value of addition}
  TempC.rel := X.val*COS(X.ang)+Y.val*COS(Y.ang);
        {Real value of addition}
  CompPolar ( TempC,Z1 );

        {Complex to Polar transformation}
 END;
PROCEDURE ZSub(VAR X,Y,Z: POLAR);
 {Procedure that input two vectors in polar form and
  subtract Y from X and output the result in Polar form}
 BEGIN
  Y.ang := PI + Y.ang;
  ZAdd(X,Y,Z);
 END;
PROCEDURE ZMul(X,Y :POLAR; VAR Z: POLAR);
 {Procedure that multiply two Polar vectors and give the
result
  in Polar}
 BEGIN
  Z.val := X.val * Y.val;
  Z.ang := X.ang + Y.ang;
 END;
PROCEDURE ZDiv( X,Y : POLAR; VAR Z : POLAR );
 {Procedure that divide Polar X by Polar Y and give the
result
  in Polar}
 BEGIN
  Z.val := X.val/Y.val;
  Z.ang := X.ang - Y.ang;
 END;
PROCEDURE ZParr(X,Y : POLAR; VAR Z : POLAR);
 {Procedure that add two polar impedances in parrallel and
  give the result in Polar}
 VAR
  Zt,Zd : POLAR;
 BEGIN
  ZMul(X,Y,Zt);
  ZAdd(X,Y,Zd);
  ZDiv(Zt,Zd,Z);
 END;

PROCEDURE THIRD ( Value : REAL ; VAR Third : REAL);
```

```
 {Procedure that return a one third the value of Value}
 BEGIN
  Third := TRD * Value;
 END;

PROCEDURE TIME ( VAR Hour,Min,Sec,Frac : INTEGER );
 TYPE
  RegPack    =  RECORD
                 AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags :
INTEGER;
                 end;
  VAR
   Regs :  RegPack;
  BEGIN
   WITH Regs DO
    BEGIN
     AX := $2C00;
     MsDos (Regs);
     Hour := hi (CX);
     Min  := lo (CX);
     Sec  := hi (DX);
     Frac := lo (DX);
    end;
  end;

PROCEDURE SeqCalc ( VAR Pseq,Nseq,Zseq :POLAR; I1,I2,I3 :
POLAR);
 {Procedure that input three phase vectors in polar form
  and return their posative, negative and zero sequence
  components in polar forms}
 VAR
  I2s1,I2s2,I3s1,I3s2 : POLAR;
  PseqT, NseqT, ZseqT : POLAR;
  Pseq3, Nseq3, Zseq3 : POLAR;
 BEGIN
  I2s1.val := I2.val;
  I2s1.ang := I2.ang + TPO3;
  I2s2.val := I2.val;
  I2s2.ang := I2.ang + FPO3;
  I3s1.val := I3.val;
  I3s1.ang := I3.ang + TPO3;
  I3s2.val := I3.val;
  I3s2.ang := I3.ang + FPO3;
  Zadd ( I2s2,I3s1,PseqT);
  Zadd ( I1,PseqT,Pseq3);
  Third ( Pseq3.val,Pseq.val);
  Pseq.ang := Pseq3.ang;
  Zadd(I2s1,I3s2,NseqT);
  Zadd(I1,NseqT,Nseq3);
  Third (Nseq3.val,Nseq.val);
  Nseq.ang := Nseq3.ang;
  Zadd(I1,I2,ZseqT);
```

```
  Zadd(ZseqT,I3,Zseq3);
  Third(Zseq3.val,Zseq.val);
 END;
PROCEDURE TherR ( VAR ThR : REAL; LR,RR : REAL );
 BEGIN
  ThR := ( LR - RR ) * SLIP + RR;
 END;
PROCEDURE SlipDcalc;
 BEGIN
  TherR ( Rce, LRce, RRce );
  TherR ( Rre, LRre, RRre );
  writeln('Slip Rre =',Rre,'Slip Rce =',Rce);
  E  := - ( Ri + Rce ) / ( Ri * Rce * Cc );
  G  := - ( 1. / ( Rre * Cr ));

 END;
PROCEDURE Initialcalc;
 BEGIN
  A1 := - (1. / ( Ri * Cs));
  B  := 1. / ( Ri * Cc) ;
  C  := 1. / Cs;
  D  := 1. / ( Ri * Cc );
  { E is slip dependent }
  F  := 1. / Cc ;
  { G is slip dependent }
  H1 := 1. / Cr;
 END;
PROCEDURE ThermalModel ( VAR Ts,Tc,Tr : REAL; Ps, Pc,
Pr:REAL);
 CONST
 H = 0.001;
 SXT = 0.166666666666;
 VAR
 CNT : INTEGER;
 K1s,K1c,K2s,K2c,K3s,K3c,K4s,K4c,K1r,K2r,K3r,K4r : REAL;
BEGIN
 CNT := CNT + 1;
  IF CNT > 1000 then
     BEGIN
      AMBIANT(Tamb);
      CNT := 0;
      end;
SlipDcalc;
{ Stator thermal solution }
{writeln ( 'power Ps=',Ps,'temp Ts =',Ts);
writeln;
writeln ( 'power Pc=',Pc,'temp Tc =',Tc);
writeln;
writeln ( ' power Pr = ',Pr,'temp Tr',Tr);
writeln;
writeln;
 }
```

```
 K1s := H * ( A1 * Ts + B * Tc + C * Ps );
 K1c := H *  ( D * Ts + E * Tc + F * Pc );
{ writeln ('K1c =',K1c);}
 K2s := H *  ( A1 * ( Ts + HLF * K1s ) + B * ( Tc + HLF *
K1c ) + C * Ps );
 K2c := H * ( D * ( Ts + HLF * K1s ) + E * ( Tc + HLF * K1c
) + F * Pc );
 K3s := H * ( A1 * ( Ts + HLF * K2s ) + B * ( Tc + HLF *
K2c ) + C * Ps );
 K3c := H * ( D * ( Ts + HLF * K2s ) + E * ( Tc + HLF * K2c
) + F * Pc );
 K4s := H * ( A1 * ( Ts + K3s ) + B * ( Tc + K3c ) + C * Ps
);
 K4c := H * ( D * ( Ts + K3s ) + E * ( Tc + K3c ) + F * Pc
);
 Ts  := Ts + SXT * ( K1s + 2. * K2s + 2. * K3s + K4s );
 Tc  := Tc + SXT * ( K1c + 2. * K2c + 2. * K3c + K4c );

{writeln ( 'G * Tr =', G*Tr, ' H1 * Pr =',H1*Pr);
} K1r := H * ( G * Tr + H1 * Pr );
 K2r := H * ( G * Tr + HLF * K1r + H1 * Pr );
 K3r := H * ( G * Tr + HLF * K2r + H1 * Pr );
 K4r := H * ( G * Tr + K3r + H1 * Pr );
 Tr  := Tr + SXT * ( K1r + 2. * K2r + 2. * K3r+ K4r );
{ Ts  := Ts +  K1s ;
 Tc  := Tc +  K1c ;
 Tr  := Tr + K1r;
}
end;
PROCEDURE TRIP(TINDEX:INTEGER);
 BEGIN
  WRITELN ('MOTOR TRIPPED');
  TIME (Hour,Min,Sec,Frac);
 { writeln('Timer ==',Hour,'==',Min,'==',Sec,'==',Frac);
 }
 END;

PROCEDURE OverCurrent(IA:REAL);
 BEGIN
  IF IA>IL THEN TRIP(1);
 END;
PROCEDURE GrndCurrent(IO:REAL);
 BEGIN
  writeln('IO  =',IO, 'IO limit =',IOL);
  IF IO>IOL THEN TRIP(2);
  END;
PROCEDURE OverVoltage(V:REAL);
 BEGIN
  IF V > Vmax THEN TRIP(3);
 END;

PROCEDURE UnderVoltage(V:REAL);
```

```
 BEGIN

  IF V < Vmin THEN TRIP(4);
 END;

BEGIN
  TIME (Hour,Min,Sec,Frac);
 { writeln('Timer ==',Hour,'==',Min,'==',Sec,'==',Frac);
 } Initialize;
  Initialcalc;
  TIME (Hour,Min,Sec,Frac);
 { writeln('Timer ==',Hour,'==',Min,'==',Sec,'==',Frac);
 } VIRead;
{  REPEAT   }
   CNT := CNT +1;
  { WRITELN (CNT);
  } READLN (VIFile,avail);
   READLN ( VIFile,Vpn.val,Vpn.ang);
      {Read phase to nutral voltage and its phase angle}
   READLN ( VIFile,Ia.val,Ia.ang);
   READLN ( VIFile,Ib.val,Ib.ang);
   READLN ( VIFile,Ic.val,Ic.ang);
   writeln ('Vp =',Vpn.val,'Ia =',Ia.val,'Ib =',Ib.val,'Ic
=',Ic.val);
      {Read three phase currents and their phase angles}
   OverVoltage(Vpn.val);
   UnderVoltage(Vpn.val);
   OverCurrent(Ia.val);
   OverCurrent(Ib.val);
   OverCurrent(Ic.val);
    {Electrical Model Calculations}
   SeqCalc(Ipos,Ineg,Izero,Ia,Ib,Ic);
   writeln('Ipos =',Ipos.val,'Ineg =',Ineg.val);
   GrndCurrent(Izero.val);
 {  writeln ('Izero =',Izero.val,'AIzero =',Izero.ang);
 }  Zdiv(Vpn,Ipos,MZ);
           {MZ is the total motor impedance}
           writeln('Motor Z =',MZ.val);
   A := ( Xrp + Xc ) / Xc;
   Irp := Ipos.val / A;
   Irn := Ineg.val / A;
   Ps := 3. * Rs * ( ( Ipos.val * Ipos.val ) + ( Ineg.val *
Ineg.val ) );
   Pc := 3. * ( Vpn.val * Vpn.val ) / Rc;
   MR := ( Vpn.val / Ipos.val ) * cos (MZ.ang);
         {MR is the induction motor apparent resistance}
 {  writeln ('RRr =',RRr);
   writeln ('MR =',MR);
   writeln ('Rs =',Rs);
 }  writeln ('LRr =',LRr);
   SLIP := RRr / ( A * ( MR - Rs ) - ( LRr - RRr ) );
    writeln ('slip =',slip);
```

```
      { writeln ('posative sequence current =',Ipos.val);
        writeln ('negative sequence current =',Ineg.val);
}     Rrp := RRr + ( LRr - RRr ) * SLIP;
    Xrp := RXr + ( LXr - RXr ) * SLIP;
    Rrn := RRr + ( LRr - RRr ) * ( 2. - SLIP );
    Xrn := RXr + ( LXr - RXr ) * ( 2. - SLIP );
    Pr := 3. * ( Rrp * Irp * Irp  + Rrn * Irn * Irn );
     {Thermal model calculations}
   { Ps := Ps * MPower;
    Pc := Pc * MPower;
    Pr := Pr * MPower;
    }writeln ('Ps =',Ps,'Pc =',Pc);
    writeln ('Pr =',Pr);
   for K := 1 to 10 do
   begin
      ThermalModel ( Ts, Tc, Tr, Ps, Pc, Pr);
      writeln ('Rre =',Rre);
      writeln ('Tr after =',Tr);
      writeln ('Ts after =',Ts,'Tc after =',Tc);
   end ;
   TIME (Hour,Min,Sec,Frac);
   writeln('Timer ==',Hour,'==',Min,'==',Sec,'==',Frac);
{ UNTIL AVAIL = 0; }
close (VIFile);
    WRITELN (' Data Unavailable');
 END.
```

TIME(SEC)	SPEED	STW TEMP	ROT TEMP
0	0	30	30
1	125.7361	32.0078	33.48895
2	259.2364	33.3891	35.70015
3	387.5724	34.50311	38.0652
4	503.6767	35.78991	40.8893
5	604.8601	37.11107	43.01268
6	698.0292	38.19880	44.98173
7	799.2759	40.00131	47.01018
8	889.1748	41.09098	48.39734
9	947.2735	42.18906	50.48801
10	1007.658	43.4984	52.0211
11	1065.970	44.4982	52.9648
12	1148.421	45.7025	54.6102
13	1187.873	46.9831	56.31004
14	1234.714	47.9713	57.29610
15	1260.388	48.88902	59.0005
16	1320.904	50.0028	60.0231
17	1393.197	51.0361	60.8994
18	1440.014	52.20117	62.10502
19	1511.546	53.39754	62.6967
20	1620.129	54.0207	63.60883
20.7745	1763.569	54.09666	63.60604
21	1785.399	54.01166	63.59023
22	1785.399	53.89466	63.43023
23	1785.399	53.77766	63.27023
24	1785.399	53.66066	63.11023
25	1785.399	53.54366	62.95023
26	1785.399	53.42666	62.79023
27	1785.399	53.30966	62.63023
28	1785.399	53.19266	62.47023
29	1785.399	53.07566	62.31023
30	1785.399	52.95866	62.15023
31	1785.399	52.84166	61.99023

APPENDIX C

MOTOR SIMULATION PROGRAM

```
{THIS PROGRAM WILL READ A SPECIFIED MOTOR DATA FILE WHICH
CONTAIN THE MOTOR
PARAMETERS THEN SOLVE THE MOTOR POSATIVE AND NEGATIVE
SEQUENCE
EQUIVELENT CIRCUITS, THEN WRITE THE VOLTAGE AND CUURENT
DATA INTO A FILE
SPECIFIED BY THE USER }


PROGRAM Motor_Protection;
CONST
  Vpn = 1.0;
  TVp = 0.0;
  ZER = 0.;
  TPO3 = 2.094395102;{ 2 pi by three in radians}
  FPO3 = 4.188790205;{ 4 pi by three in radians}
VAR
  FMotInf,FSVInf : TEXT;
  SVInf,MotInf : STRING[14];
  AIA,TIA,AIB,TIB,AIC,TIC,R3,X3,R6,X6,LR1,RRO,LX1,RXO :
REAL;
  S,R1,X1,R2,X2,R4,X4,R5,X5,MR,AI1,TI1,AI2,TI2,A:REAL;
  AZ1,TZ1,AZ6,TZ6,AZ4,TZ4,AZP,TZP,AMP,TMP : REAL;
  AZ5,TZ5,AZN,TZN,AMN,TMN :REAL;
  NVPC : REAL; {-ve seq percent out of +ve seq}
  AVAIL : INTEGER;
  SP :INTEGER;
  PIOT,ATEST,TTEST,MPOWER:REAL;
PROCEDURE Initialize;
BEGIN
  WRITE(' What is the name of the file which has the motor
data ?  ');
  READLN(MotInf);
  ASSIGN (FMotInf,MotInf);
  RESET(FMotInf);
  READLN (FMotInf,Mpower);
  READLN (FMotInf,R3,X3);
 { WRITELN('Stator Resistance =',R3,'    Stator Reactance
=',X3);}
  READLN (FMotInf,R6,X6);
 { WRITELN('Mutual Resistance =',R6,'    Mutual Reactance
=',R6);   }
```

```
  READLN (FMotInf,LR1,RR0);
 { WRITELN('Locked Rotor Resistance =',LR1,'   Running
Rotor Reactance =',RR0);}
  READLN (FMotInf,LX1,RX0);
{  WRITELN('Locked Rotor Reactance =',LX1,'   Running Rotor
Reactance =',RX0);}
  CLOSE(FMotInf);
  PIOT := PI/2.;
  X4 := LX1;
  A := (X4 + X6)/X6;
END;
PROCEDURE Polar(R,X:REAL;VAR AZ,TZ:REAL);
BEGIN
{  WRITELN(R,X);}
  AZ := SQRT(R*R + X*X);
  TZ := ARCTAN(X/R);
  IF R < 0.0 THEN TZ := PI + TZ;
END;
PROCEDURE ZAdd(AX,TX,AY,TY:REAL;VAR AZ,TZ:REAL);
VAR
  ZR,ZI : REAL;
BEGIN
  ZI := AX*SIN(TX)+AY*SIN(TY);
  ZR := AX*COS(TX)+AY*COS(TY);
  AZ := SQRT(ZR*ZR+ZI*ZI);
  TZ := ARCTAN(ZI/ZR);
  IF ZR < 0.0 THEN TZ := PI + TZ;
{  WRITELN(ZR,ZI,AZ,TZ); }
END;
PROCEDURE ZSub(AX,TX,AY,TY:REAL;VAR AZ,TZ:REAL);
BEGIN
  TY := PI + TY;
  ZAdd(AX,TX,AY,TY,AZ,TZ);
END;
PROCEDURE ZMul(AX,TX,AY,TY:REAL;VAR AZ,TZ:REAL);
BEGIN
 AZ := AX * AY;
 TZ := TX + TY;
END;
PROCEDURE ZDiv(AX,TX,AY,TY:REAL;VAR AZ,TZ:REAL);
 BEGIN
  AZ := AX/AY;
  TZ := TX - TY;
 END;
PROCEDURE ZParr(AX,TX,AY,TY:REAL;VAR AZ,TZ:REAL);
VAR
ZMA,ZMT,ZAA,ZAT:REAL;
  BEGIN
    ZMul(AX,TX,AY,TY,ZMA,ZMT);
    ZAdd(AX,TX,AY,TY,ZAA,ZAT);
    ZDiv(ZMA,ZMT,ZAA,ZAT,AZ,TZ);
  END;
```

```
PROCEDURE Fill_In_Motor_Operating_Values;
BEGIN
  AVAIL := 1;
  WRITE('What percentage is the -ve seq voltage in % ?  ');
  READLN(NVPC);
  NVPC := NVPC / 100.;
  WRITE(' What file is to contain motor current and voltage
data ?  ');
  READLN(SVInf);
  ASSIGN (FSVInf,SVInf);
  REWRITE(FSVInf);
  FOR SP := 0 TO 199 DO
   BEGIN
    S := 1.-SP/200.;
 {   WRITELN(S); WRITELN(R4,X4);}
    X4 := S*(LX1-RX0)+RX0;
    R4 := S*(LR1-RR0)+RR0;
{    WRITELN(R4,X4);      }
    Polar(R4/S,X4,AZ4,TZ4);{+ve seq for rotor}
{    WRITELN(AZ4,TZ4);}
    R5 := (LR1-RR0)*(2.-S)+RR0;
    X5 := (LX1-RX0)*(2.-S)+RX0;
    Polar(R5/(2.-S),X5,AZ5,TZ5);{-ve seq for rotor}
    ZParr(R6,ZER,X6,PIOT,AZ6,TZ6);{core loss equivalent for
+ve and -ve}
{    WRITELN(AZ6,TZ6);}
    ZParr(AZ4,TZ4,AZ6,TZ6,AZP,TZP);{+ ve seq equivalent}
    ZParr(AZ5,TZ5,AZ6,TZ6,AZN,TZN);{-ve seq equivalent}
{    WRITELN(AZP,TZP);}
    Polar(R3,X3,AZ1,TZ1); {stator imped for +ve and -ve}
{    WRITELN(AZ1,TZ1);}
    ZAdd(AZ1,TZ1,AZP,TZP,AMP,TMP);
    ZAdd(AZ1,TZ1,AZN,TZN,AMN,TMN);
{    WRITELN(AMP,TMP);}
    ZDiv(VPN,TVP,AMP,TMP,AI1,TI1);
    ZDiv(NVPC,ZER,AMN,TMN,AI2,TI2);
  {  WRITELN(AI2,TI2);
    WRITELN(FSVInf,AVAIL,S);
    WRITELN(FSVInf,AI1,TI1);
    WRITELN(FSVInf,AI2,TI2); }
    {convert from sequence currents to phase currents}
    ZAdd(AI1,TI1,AI2,TI2,AIA,TIA);
    ZAdd(AI1,TI1 + TPO3,AI2,TI2 + FPO3,AIB,TIB);
    ZAdd(AI1,TI1 + FPO3,AI2,TI2 + TPO3,AIC,TIC);
    WRITELN (FSVInf,AVAIL);
    WRITELN (FSVInf,VPN,TVP);
    WRITELN(VPN,TVP,TVP*180/PI);
    WRITELN(FSVInf,AIA,TIA);
    WRITELN(AIA,TIA,TIA*180/PI);
    WRITELN(FSVInf,AIB,TIB);
    WRITELN(AIB,TIB,TIB*180/PI);
    WRITELN(FSVInf,AIC,TIC);
```

```
    WRITELN(AIC,TIC,TIC*180/PI);

   END;
  CLOSE(FSVInf);

END;
BEGIN
  Initialize;
  Fill_In_Motor_Operating_Values;
  {WRITELN('R3 = ',R3,'X3 = ',X3);
  WRITELN('R6 = ',R6,'X6 = ',X6);
  WRITELN(LR1,RR0);
  WRITELN(LX1,RX0);
  WRITELN(AIA,TIA,AIB,TIB,AIC,TIC,VPN);
  WRITELN('ADDITION',RS,XS);}
END.
```

```
1
  1.0000000000E+00  0.0000000000E+00
  5.9409325413E+00 -1.4302731383E+00
  5.9409325413E+00  6.6412196368E-01
  5.9409325413E+00  2.7585170667E+00
1
  1.0000000000E+00  0.0000000000E+00
  5.9359826735E+00 -1.4301813887E+00
  5.9359826735E+00  6.6421371325E-01
  5.9359826735E+00  2.7586088163E+00
1
  1.0000000000E+00  0.0000000000E+00
  5.9310393646E+00 -1.4300876550E+00
  5.9310393646E+00  6.6430744702E-01
  5.9310393646E+00  2.7587025500E+00
1
  1.0000000000E+00  0.0000000000E+00
  5.9261025695E+00 -1.4299919093E+00
  5.9261025695E+00  6.6440319270E-01
  5.9261025695E+00  2.7587982957E+00
1
  1.0000000000E+00  0.0000000000E+00
  5.9211722426E+00 -1.4298941235E+00
  5.9211722426E+00  6.6450097853E-01
  5.9211722426E+00  2.7588960815E+00
1
  1.0000000000E+00  0.0000000000E+00
  5.9162483377E+00 -1.4297942686E+00
  5.9162483377E+00  6.6460083337E-01
  5.9162483377E+00  2.7589959364E+00
1
  1.0000000000E+00  0.0000000000E+00
  5.9113308082E+00 -1.4296923153E+00
  5.9113308082E+00  6.6470278670E-01
  5.9113308082E+00  2.7590978897E+00
1
  1.0000000000E+00  0.0000000000E+00
  5.9064196067E+00 -1.4295882334E+00
  5.9064196067E+00  6.6480686859E-01
  5.9064196067E+00  2.7592019716E+00
1
  1.0000000000E+00  0.0000000000E+00
```

2

VITA

Ali Mohamed Khider

Candidate for the degree of

Doctor of Philosophy

Thesis: MICROPROCESSOR-BASED PROTECTION RELAY FOR LARGE INDUCTION MOTORS

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Baghdad, Iraq, on September, 24, 1945. Married, Three children.

Education: Attended primary and secondary schools in Baghdad, Iraq; received the Bachelors of Science Degree in Electrical Engineering from the University of Baghdad, Baghdad, Iraq in June, 1965; Master of Science Degree in Electrical Engineering from Oklahoma State University in May, 1982.

Professional Experience: Area Electrical Engineer, Northern Petroleum Organization, Kirkuk, Iraq, 1966-1980; Teaching and Research Assistant, Oklahoma State University, 1982-1984; Senior Reliability Engineer, National Semiconductors, Santa Clara, CA, 1984-1987; Principal Component Engineer, Digital Equipment Corporation, Palo Alto, CA, 1987-1988; Senior Quality Assurance Engineer, E G & G Reticon, Sunnyvale, CA 1988-Present.

Professional Organizations: Member of The Institute of Electrical Engineers, London, England; Charter Engineer, Council of Engineering Institutions, London, England.

Honor Society: ETA Kappa Nu.