To the Graduate Council:

I am submitting herewith a dissertation written by William Merle Cash entitled "The Direct Digital Control of a Dynamic Adiabatic Calorimeter." I recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Chemical Engineering.

Charles F. Moore

Charles F. Moore, Major Professor

We have read this dissertation
and recommend its acceptance:

E. E. Stansbury

Deane D Bruns

Walter L. Evans

C R Brooks

Accepted for the Council:

L Evans Reed

Vice Chancellor
Graduate Studies and Research

Thesis
79b
.C329
cop. 2

THE DIRECT DIGITAL CONTROL OF A DYNAMIC ADIABATIC CALORIMETER

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

William Merle Cash

December 1979

1406089

ACKNOWLEDGMENTS

I would like to extend my gratitude to all those who aided me in completing this research:

My parents, George and Georgie Cash, for all the love and encouragement any son could ever receive.

My advisor, Charlie Moore, for his patient guidance. He never failed to provide a small light at the end of the tunnel by suggestions of possible solutions to difficult problems.

All the guys in the electronics shop, especially Ted Long. Without their aid, the data acquisition system might never have reached its full potential.

All the guys in the machine shop. Although this project required little machine work, their friendship and jovial nature always added a note of cheer. I especially would like to thank Mr. Tommy "Coon-Pick" Thompson for his enlightenment.

Dennis Hetzner, the man from Northern Dixie (Baltimore, Maryland), without whose ride every day I would never have gotten in to UT to do any work. Also I appreciate "chasing trains," hunting, and fishing together (even though he outshot me and caught more and bigger fish than I did).

My wife, Miriam, whose love and understanding made all this hard work easier and worthwhile.

ABSTRACT

This work sets forth a procedure used to place a dynamic adiabatic calorimeter under direct digital control. This process is inherently non-linear and multivariable as typical of many industrial processes. The process was placed under control by developing a detailed model of the process on both a theoretical and experimental basis. This model was then used to design a control strategy, select controllers to implement that strategy, and tune the controller algorithms. The process non-linearities were controlled by changing controller tuning parameters based upon operating conditions. The multivariable interactions were not of sufficient strength to warrant special consideration in the final control strategy.

This direct digital control strategy was then used to determine the heat capacity of titanium from 320 to 1020K within ± 1.0% accuracy over the entire temperature range.

TABLE OF CONTENTS

CHAPTER		PAGE
INTRODUCTION		1
I.	BACKGROUND	3
	Approach to the Design of a Digitally Based Control System	9
	Justification	9
	Data Acquisition System	10
	Theoretical Process Model	10
	Experimental Process Model	11
	Control Strategy	15
II.	EQUIPMENT	20
	Improvements in Data Acquisition System	22
	Computer Interface to Process Inputs	24
	Process Monitoring System	26
III.	EXPERIMENTAL PROCESS MODEL	29
	Review of Theoretical Process Model	29
	Selection of Theoretical Modeling Domain	32
	Experimental Testing Procedure	34
	Qualitative Analysis of Process Responses	44
	Quantitative Analysis of Process Response	52
	Critical Shield Identification	78
	Summary	80
IV.	CONTROL STRATEGY	82
	Control Scheme	83
	Controller Selection	84
	Controller Tuning	96
	Final Adjustment of Controller Parameters During Operation	109
	Summary	113
V.	SYSTEM SOFTWARE AND OPERATION	115
VI.	HEAT CAPACITY OF TITANIUM	122
VII.	CONCLUSIONS AND RECOMMENDATIONS	126
REFERENCES		128
APPENDICES		132

CHAPTER PAGE

A. RATIONALE FOR MODELING THE DIFFERENCE IN THE SHIELD TEMPERATURE AND SAMPLE TEMPERATURE 133

B. SPECIAL METHOD FOR MODELING PROCESS INTERACTIONS WITH THE NON-TESTED SHIELDS REMAINING IN AUTOMATIC. 135

C. CHARACTERIZATION OF THE CYL-G1 INTERACTION 138

D. SAMPLE DRIFT RATE MODEL. 142

E. DETERMINATION OF THE DISCRETE VERSION OF A LEAD COMPENSATOR USING THREE METHODS. 147

F. FIRST ORDER DEADBEAT ALGORITHM 151

G. COMPLETE LISTING OF ALL RSX PROGRAMS 155

H. MAJOR VARIABLES USED IN THE PROGRAMS LISTED IN APPENDIX G. . 188

I. CALORIMETER OPERATING MANUAL 191

VITA 208

LIST OF TABLES

TABLE		PAGE
I.	Conditions and Legend for Modeling Runs	57
II.	Constants for Temperature and Thermocouple Emf Dependency Equations for Shields G3, G1, Cyl, and G2	59
III.	Constants in Damping Ratio Dependency Equation for Shields Cyl and G2.	71
IV.	Process Parameters and Deadbeat Parameters for Shield G3. .	100
V.	Process Parameters and Deadbeat Parameters for Shield G1. .	101
VI.	Final Controller Parameters for Shield G3	105
VII.	Final Controller Parameters for Shield G1	107
VIII.	Final Controller Parameters for Shield Cyl.	110
IX.	Final Controller Parameters for Shield G2	112
X.	PDP-15 Program Names and Functions RSX Monitor.	116
XI.	DEC-10 Program Names and Functions.	121
XII.	Pseudo-Heat Transfer Coefficients from Drift Rate Analysis.	143
XIII.	Data Used in the Determination of Drift Rate Parameters . .	145
XIV.	Analog Controller Values.	195

LIST OF FIGURES

FIGURE		PAGE
I-1.	Schematic Diagram of the Calorimeter	4
I-2.	Adiabatic Nature of the Calorimeter.	6
I-3.	Dynamic Nature of the Calorimeter.	8
I-4.	LaPlace Domain Block Diagram for the Calorimeter	12
I-5.	P-Canonical Model for a 2x2 System	14
I-6.	V-Canonical Model for a 2x2 System	14
I-7.	Multivariable Process with Conventional Analog Control	16
I-8.	Pseudo-Process Resulting from the Combination of the Decoupler and Process into One Unit.	18
I-9.	P-Canonical Model with Matrix Decoupler.	19
I-10.	V-Canonical Model with Block Diagram Decoupler	19
II-1.	Data Acquisition Flow Diagram.	21
II-2.	Schematic Diagram of the New Eight Channel Multiplexer	23
II-3.	Schematic Diagram of the Complete Computer Link to the Calorimeter for a Typical Shield	25
III-1.	Modeling Procedure for a Multivariable Process Using Step Testing.	36
III-2.	Overall Flow Diagram of Process from Computer Output to Computer Input	37
III-3.	Experimental Response of a Shield during a Typical Step Test	39
III-4.	Standard Modeling Technique with All Controllers in Manual	41
III-5.	Special Modeling Technique with All Controllers in Automatic Except One to be Stepped	43
III-6.	Typical Step Response for the Shield G3.	45
III-7.	Typical Step Response for the Shield G1.	46

FIGURE PAGE

III-8. Typical Step Response for the Shield Cyl 47

III-9. Typical Step Response for the Shield G2. 48

III-10. Effect on the Response and Controller Output of the Shield G1 During the Step Test of the Shield Cyl in Figure III-8 51

III-11. Step Response for the Shield G3 from Figure III-6 with the Fitted Curve and Process Parameters. 53

III-12. Step Response for the Shield G1 from Figure III-7 with the Fitted Curve and Process Parameters. 54

III-13. G3 Gain vs Thermocouple Emf. 55

III-14. G3 Time Constant vs Thermocouple Emf 56

III-15. G3 Dead Time vs Thermocouple Emf 56

III-16. Ratio of Time Constant to Gain for Shield G3 vs Thermocouple Emf 60

III-17. G1 Gain vs Thermocouple Emf. 62

III-18. G1 Time Constant vs Thermocouple Emf 62

III-19. G1 Dead Time vs Thermocouple Emf 63

III-20. Step Response for the Shield Cyl from Figure III-8 with the Fitted Curve and Process Parameters. 65

III-21. Step Response for the Shield G2 from Figure III-9 with the Fitted Curve and Process Parameters. 66

III-22. Cyl Gain vs Thermocouple Emf 68

III-23. Cyl Reciprocal Frequency vs Thermocouple Emf 69

III-24. Cyl Damping Ratio vs Thermocouple Emf. 70

III-25. G2 Gain vs Thermocouple Emf. 73

III-26. G2 Reciprocal Frequency vs Thermocouple Emf. 74

III-27. G3 Damping Ratio vs Thermocouple Emf 74

III-28. Typical Plot of G1 Controller Output vs Cyl Response . . . 76

III-29. Cyl-G1 Interaction Gain vs Thermocouple Emf. 77

FIGURE PAGE

III-30. Sample Drift Rate vs Offset in Thermocouple Emf for Shields G3, G1, $\overline{Cyl}$, and G2 at 1020 °K 79

IV-1. Final Control Strategy Consisting of Single Loop Control for Each Shield. 85

IV-2. Comparison of Controller Types for a Set Point Change to a Second Order Lag Process 88

IV-3. Load Response for the Bode Designed Lead Compensator in Figure IV-2. 89

IV-4. Cascade Approach to the Control of a Second Order Lag Process. 91

IV-5. Cascade Approach to the Control of a Second Order Lag Process Using Feedback Around Part of the Process Model. 93

IV-6. Cascade Approach to the Control of a Second Order Lag Process Using Feedback Around Part of the Process Model Using Z-Transforms 94

IV-7. Simulation Loop for Controller Tuning. 97

IV-8. First Order Lag Deadbeat Response Using Unrestrained Tuning 98

IV-9. Typical First Order Response to a Set Point and Load Change Using Load Tuning and a Throttled Controller. . . 106

IV-10. Typical Second Order Response to a Set Point and Load Change Using Load Tuning and a Throttled Controller. . . 111

VI-1. Heat Capacity of Titanium from 320 to 1020K. 123

B-1. Control Loop with an Interaction Entering Loop as a Disturbance. 136

C-1. Block Diagram for Cyl-G1 Type Interaction. 141

F-1. Digital Control Loop with Noise. 152

INTRODUCTION

During the last fifteen years, digital computers have greatly enhanced the field of process control. This is due to three major aspects of the digital computer:

1) Readily accessible storage of vast quantities of data,
2) Extreme flexibility via software changes, and
3) Capability of the implementation of more sophisticated control strategies than possible with conventional analog devices.

With the advent of more powerful and less costly digital computers available through improved technology, digitally based operating systems are becoming more and more attractive. Not only are they capable of collecting and storing vast amounts of data for on-line as well as off-line analysis, but they are also capable of assuming all aspects of process control. In addition to the low-level control of temperature, pressure, and flow loops, digital computers can be used to make decisions based upon economic or other factors in order to operate the overall process in the most efficient manner. This is commonly known as supervisory control.

In the area of process control on a low-level basis, direct digital control (DDC) can be used to implement more sophisticated control strategies than possible with conventional analog devices. The improved control gained by these strategies can be very effective in enhancing product quality, reducing operating costs, and minimizing the production of off-quality product during transitions from one

product line to another. This can result in a substantial increase in the profitability of the process.

In order to assist the advancement of DDC in industry, The University of Tennessee, Knoxville, Department of Chemical, Metallurgical, and Polymer Engineering has undertaken a program involving the theoretical investigation and practical application of DDC to control problems of industrial importance (1,2,3,4,5,6).

It is the intent of this author to further this study by the practical application of a digitally based control system to a process which is inherently non-linear and multivariable.

CHAPTER I

BACKGROUND

The process chosen to study the practical application of a digitally based control and data analysis system is that of a dynamic adiabatic calorimeter. This calorimeter is part of the equipment in a program undertaken in Metallurgical Engineering at The University of Tennessee, Knoxville in 1950. It is used to determine the heat capacity of metals from 300 to 1300K.

Many researchers (7,8,9,10,11,12,13) have been involved in the design and use of this calorimeter shown in Figure I-1. Basically, it consists of a sample of which the heat capacity is to be determined and four surrounding adiabatic shields, designated as G3, G1, Cyl, and G2. Detailed specifications of dimensions and materials of construction are given by Peng (12) and Stout (13).

Due to the operation of the calorimeter under a vacuum, heat transfer is mainly accomplished between shields by radiation (14,15, 16,17). Since this depends on the fourth power of the absolute temperature, the process is non-linear (18,19). This results in process parameters which vary with temperature and therefore time.

Since heat can be radiated from one shield to all the others, an increase in one shield temperature affects all the others. This fact makes the process multivariable (18,19,20).

Therefore, the calorimeter contains the following control problems:

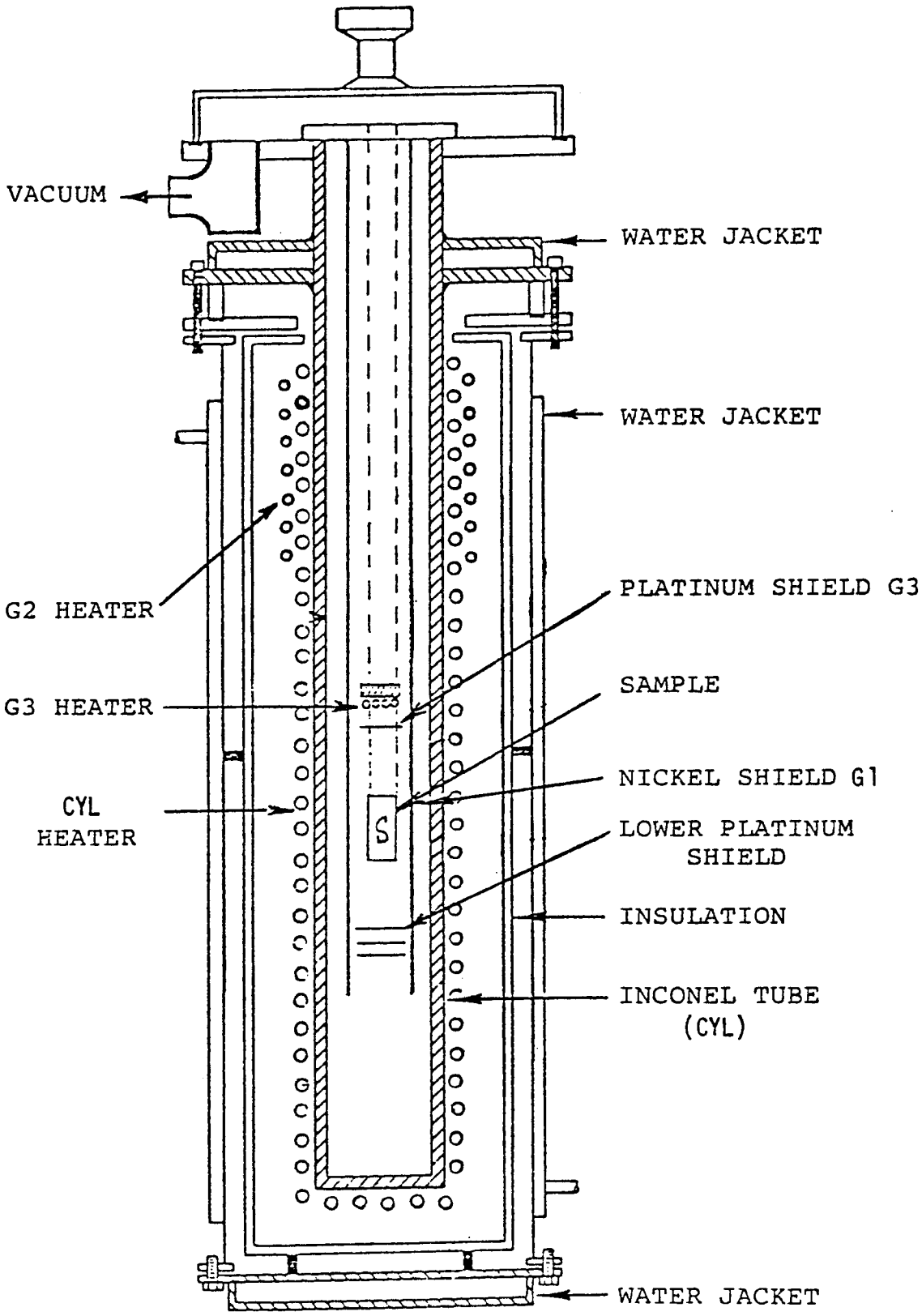


Figure I-1. Schematic Diagram of the Calorimeter.

1) Non-linear process,

2) Time varying process parameters,

3) Multivariable interactions, and

4) Data collection requirements.

A digital computer based control system offers certain advantages over conventional analog control in dealing with such problems. These advantages lie in two distinct process areas:

1) Data collection and analysis, and

2) Improved process control.

Improvements in data collection and analysis lie in the replacement of manual data taking and subsequent calculations. Operation of the calorimeter is accomplished by continuously supplying a DC voltage to the sample via an internal heater (Figure I-2). By measuring the applied DC voltage (E), the induced heater current (I), and the time ($\Delta\Theta$) it takes the sample temperature to increase a prespecified amount (ΔT), the average heat capacity of the sample can be calculated from:

$$C_p = \frac{EI\ \Delta\Theta}{M\ \Delta T} \qquad 1\text{-}1$$

where M = total sample mass including the heater.

This equation is obtained by assuming zero heat loss from the sample as indicated in Figure I-2. This results in the adiabatic nature of the calorimeter. If this assumption is violated by poor control of the shield temperatures to the sample temperature, severe errors in the heat capacity will result.

The dynamic nature of the calorimeter is a direct result of the adiabatic conditions. If the heat loss is maintained at zero, all the power supplied to the heater will result in a continual increase in

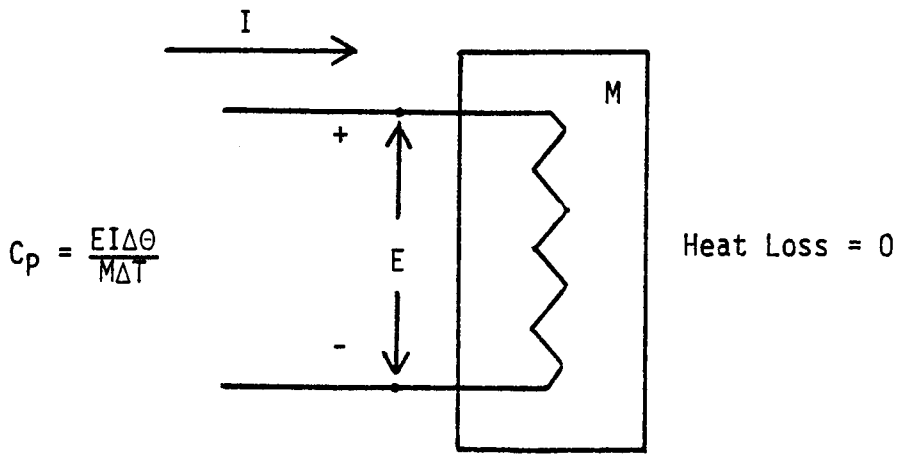


Figure I-2. Adiabatic Nature of the Calorimeter.

sample temperature with time (Figure I-3-A), i.e., the dynamic nature. As a result, heat capacity measurements can be made every ΔT interval in temperature, and these values can be plotted against the absolute temperature (T) as in Figure I-3-B.

The digitally based data acquisition system can assume all these data collecting tasks. Not only can it take more data than possible manually (which results in higher accuracy by using average rather than point values), it can also be used for more sophisticated data analysis on an off-line basis (21).

Improvements in control over the currently used, conventional analog control system arise from the fact that the digital system can be designed to take into account the process non-linearities and multivariable interactions (20). Since the non-linearity implies time (temperature) varying parameters, the control parameters must be adjusted to the changing process in order to maintain a "well-tuned" controller. This is easily accomplished in the computer by continually updating the control parameters based upon changes in the major process variable--temperature. The multivariable interactions can be accounted for using a decoupler type approach (20). These improvements will aid in the accomplishment of adiabatic conditions; therefore, the zero heat loss assumption in equation 1-1 will be maintained.

Not only does the computer allow the use of more sophisticated control strategies (20,22,23,24,25), it also permits the control of all the process loops from one device. This eliminates the need for having a separate hardware device for each control loop. This is important in the sense that a change in control strategy can be

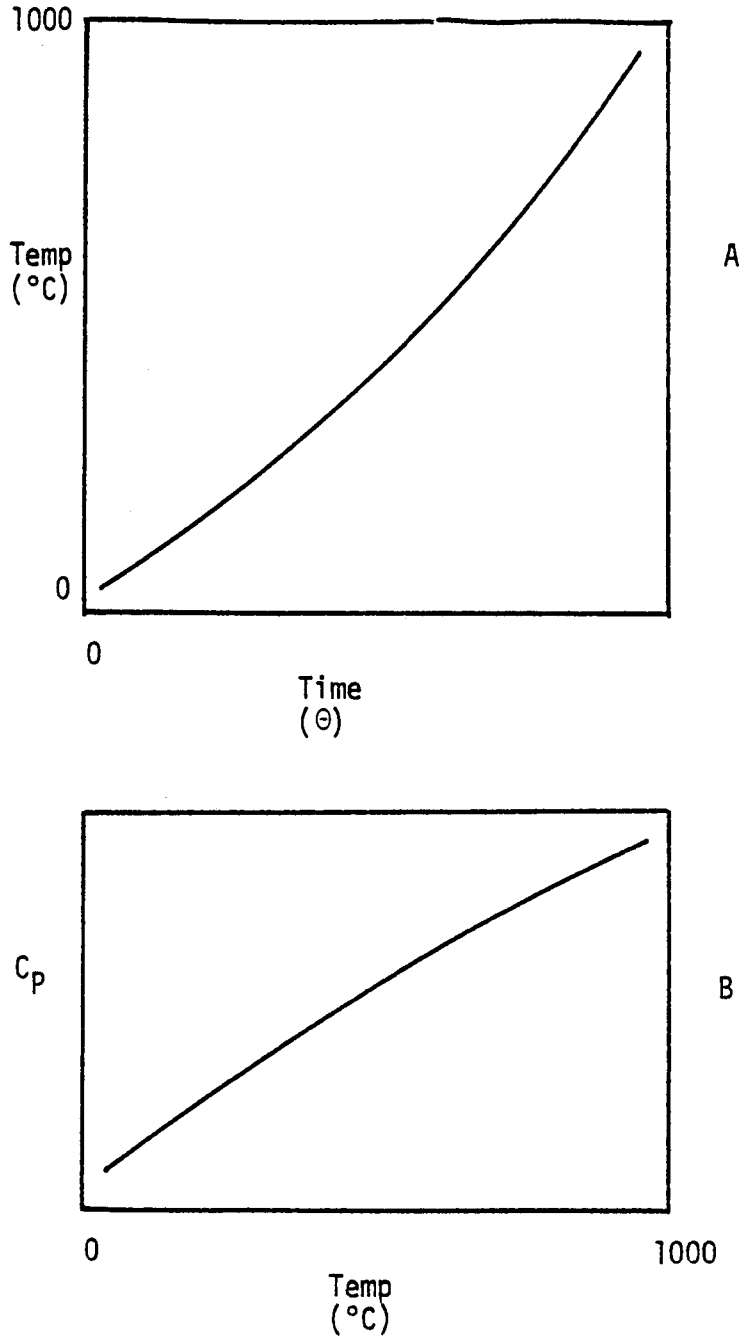


Figure I-3. Dynamic Nature of the Calorimeter.

accomplished by software, rather than hardware, changes.

Approach to the Design of a Digitally Based Control System

Having realized the advantages afforded by a digitally based control system, there are five major steps required to develop the entire system as deemed necessary by this author:

1) Justification of the change from conventional analog control to digital control based upon economics and the actual advantage to be gained in data collection and control;
2) Development of a data acquisition system capable of collecting process data within a required accuracy and rate;
3) Development of a process model, both in the theoretical and experimental domains, which accurately predicts the process response characteristics;
4) Development of the digital control system necessary to achieve the best possible control of the process using the process model as a guideline; and
5) Implementation of the final computer system which will assume all manual operations and process control on a permanent basis.

Justification

The switch to digital control is becoming more justifiable on a cost and improved control basis every day. With microprocessors, cost has been greatly reduced and industries are converting to digital control at a rapidly increasing rate. But each project must be viewed on individual merit. The current project is justifiable on an academic basis.

Data Acquisition System

The data acquisition system for the calorimeter was installed and characterized previously by the author (21). It has been used to successfully determine heat capacity data suitable for publication (26).

Theoretical Process Model

Before a meaningful experimental model can be developed, the qualitative nature of the process response must be determined through theoretical means; therefore, a theoretical process model must be developed.

The theoretical model has its basis in equations which describe the time behavior of the process (18,19,27,28,29) in differential equation form. The purpose of the theoretical model is to aid in:

1) Understanding of the process,

2) Guide to data collection and analysis, and

3) Identification of special control problems.

Without this understanding, modeling may become a shot-in-the-dark approach in which process data may be fitted to equations having no physical relationship to the process. Without realizing the existence of special control problems, the final control system would be inferior to one which takes into account these special problems.

In the case of the calorimeter, previous work by this author (21) suggested three possible domains from which the theoretical model could be viewed. The first was the time domain which consisted of nine differential equations arising from energy balances on the sample and

the adiabatic shields which surround it. This domain identified the non-linear nature of the process exhibited by temperatures raised to the fourth power in radiative heat transfer. These equations were then transferred to the discrete domain by approximating the differential operator $d/d\Theta$ by a discrete difference $\Delta/\Delta\Theta$. This resulted in a set of nine recursive digital relationships with unknown heat transfer coefficients. Another possible solution was to transfer the time domain equations to the LaPlace domain. This resulted in a block diagram description of the process as seen in Figure I-4. It displays the multivariable nature of the process. Also, in the linearization of the non-linear term T^4, time (temperature) varying parameters were obtained.

In summary, the theoretical model identified the special control problems of process non-linearities and multivariable interactions. It also demonstrated the temperature variation of the process parameters. These process characteristics should be quantified in the experimental model.

Experimental Process Model

The experimental process model can be obtained by optimizing the parameters in the theoretical model. This procedure has been found to be very useful in determining experimental process models (27,28, 29,30). Once these parameters have been determined, they can be incorporated into the system of equations which constitutes the theoretical model and used to accurately predict the process response to any set of process inputs. This system of theoretical model

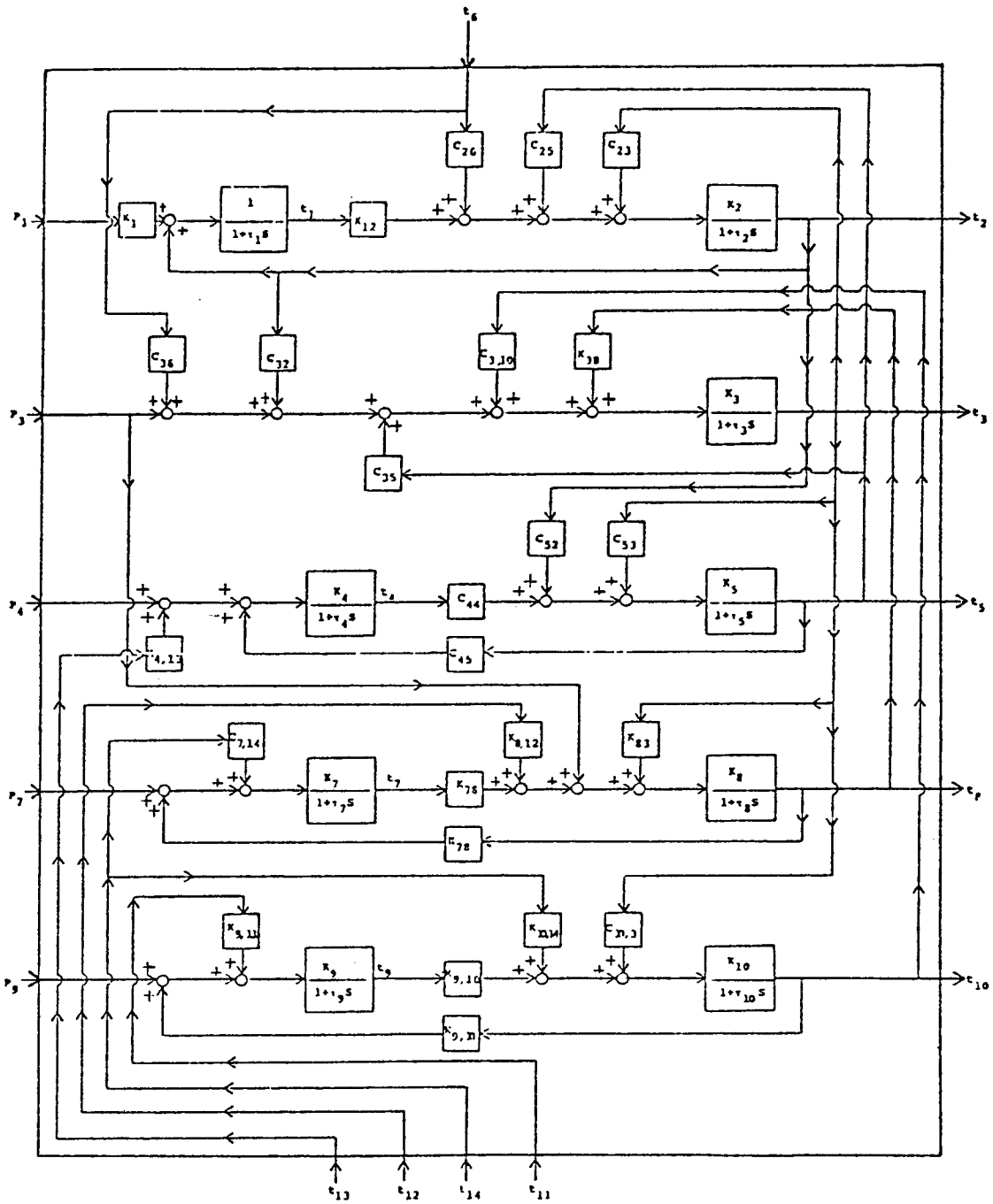


Figure I-4. LaPlace Domain Block Diagram for the Calorimeter.

equations and its optimum parameters constitute the experimental model.

Optimization involves the direct comparison of the theoretical model response to a set of experimentally obtained data. By using the same input to the theoretical model as was used to collect the experimental data, the model parameters can be systematically varied until the error between the model response and experimental response becomes a minimum. These parameters are then called the "optimum" parameters.

Several optimum seeking methods have been proposed and used to successfully determine parameters in theoretically based equations (30,31,32). They range from the well-known method of least squares (31) to methods which are well suited to recursion relationships, such as the PATTERN Search technique (30).

In the case of the calorimeter, the experimental model can be obtained from any of the three suggested theoretical domains. The time domain involves the solution of the nine differential equations and calculation of the heat transfer coefficients. The discrete domain allows determination of these coefficients by optimization. The LaPlace domain involves step-testing of the process and determination of the process transfer functions and their parameters.

This LaPlace domain model may be expressed in two convenient forms. These are the P-Canonical form of the model and the V-Canonical form (20). These forms are depicted in Figures I-5 and I-6, respectively, for a 2x2 system. The transfer functions in these models can be determined by stepping the process inputs and monitoring the process outputs.

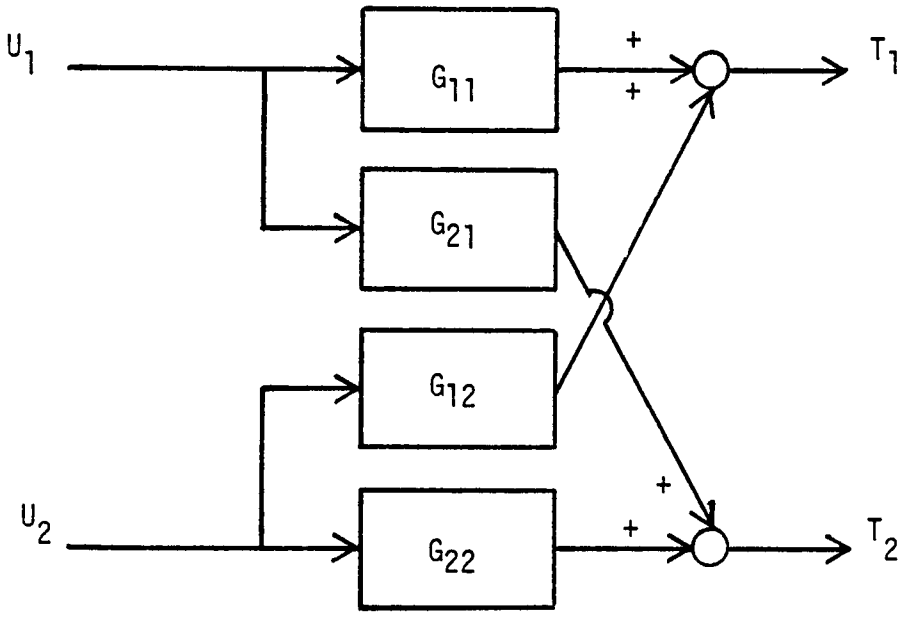


Figure I-5. P-Canonical Model for a 2x2 System.

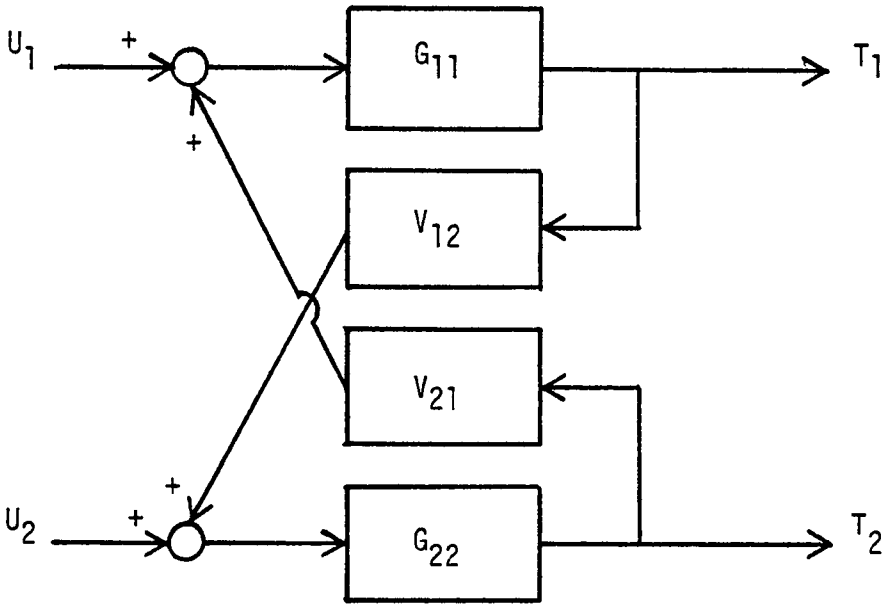


Figure I-6. V-Canonical Model for a 2x2 System.

In summary, the experimental model involves selection of an appropriate theoretical model domain from which to work. The parameters in that model are then to be determined using optimization and experimental data. These optimum parameters can then be incorporated into the model equations and used to develop and tune the appropriate control strategy.

Control Strategy

Digital process control of linear, non-interacting processes has been studied in depth (18,19,33,34,35,36,37,38), but these methods become less effective when applied to non-linear processes with multivariable interactions.

In the case of non-linearities, special methods must be employed to achieve adequate control. Some non-linearities, such as dead time, have been analyzed in depth (1,2,24,25). But in the present study, the strategy involves the changing of controller tuning parameters as the process parameters change (which implies changing temperature). This means the solution to the non-linear control problem is to determine the temperature dependence of the controller tuning parameters.

In the case of multivariable interactions, additional control actions must be added to compensate for the interactions. Figure I-7 displays a multivariable process and the conventional means of analog control. Each loop is tuned separately and then all controllers are placed in automatic. If the interactions are of sufficient strength, the process response on a closed loop basis will be degraded and may even become unstable.

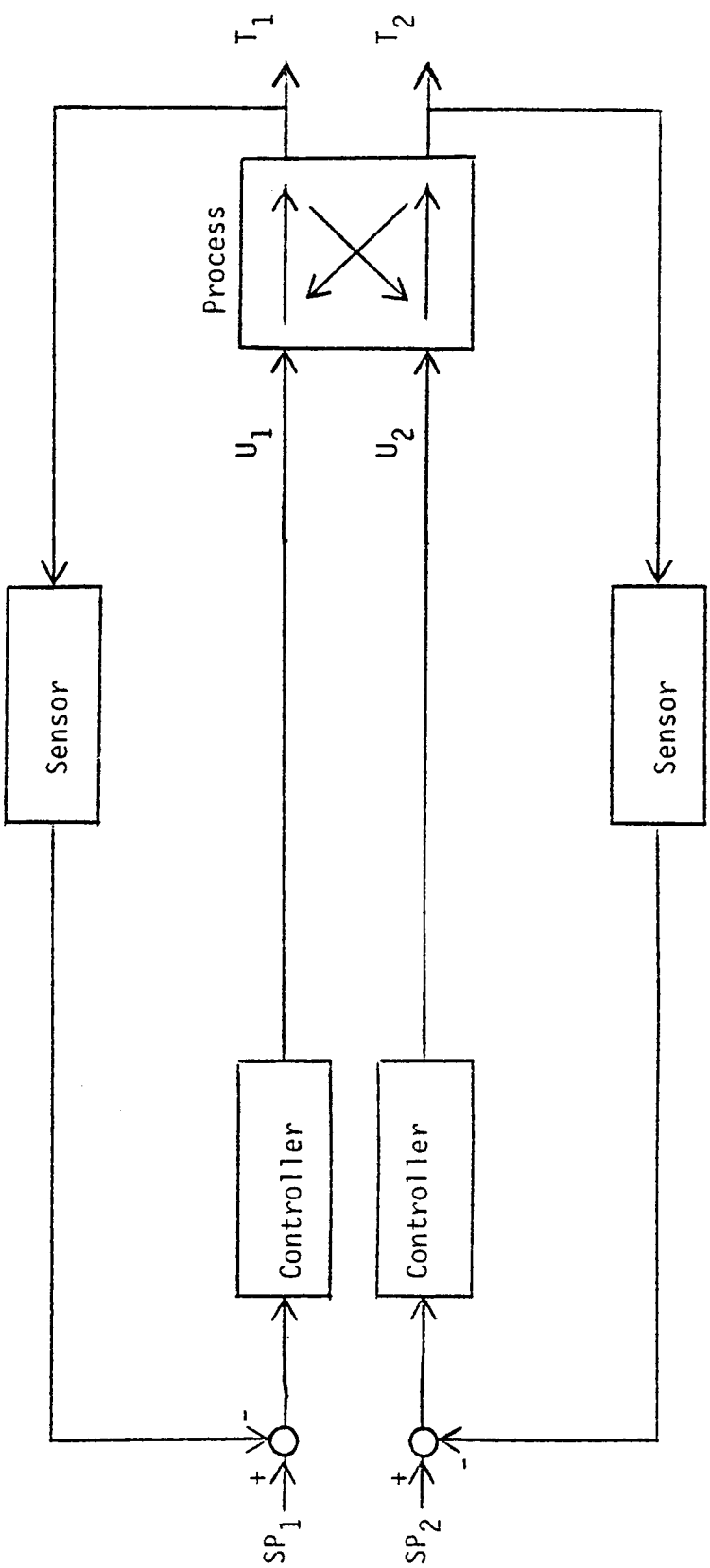


Figure I-7. Multivariable Process with Conventional Analog Control.

Multivariable control strategies (20,39,40,41,42,43) can be employed to reduce this degradation in control. One of these methods employs a decoupler to eliminate process interactions. Figure I-8 depicts the location of the decoupler in the control loop. The main controller then sees a new pseudo-process (made up of the decoupler plus process) with no interactions. The decoupler itself is based upon the type model used. Processes modeled using the P-Canonical model can be decoupled, as shown in Figure I-9, using a P-Canonical Matrix Decoupler (20). Processes modeled using the V-Canonical model can be decoupled, as shown in Figure I-10, using a V-Canonical Block Diagram Decoupler (20).

In summary, the final control strategy must account for the process non-linearities and multivariable interactions if necessary. The overall controller types must be selected and tuned using the experimental model. This final strategy must then be implemented and used to control the process.

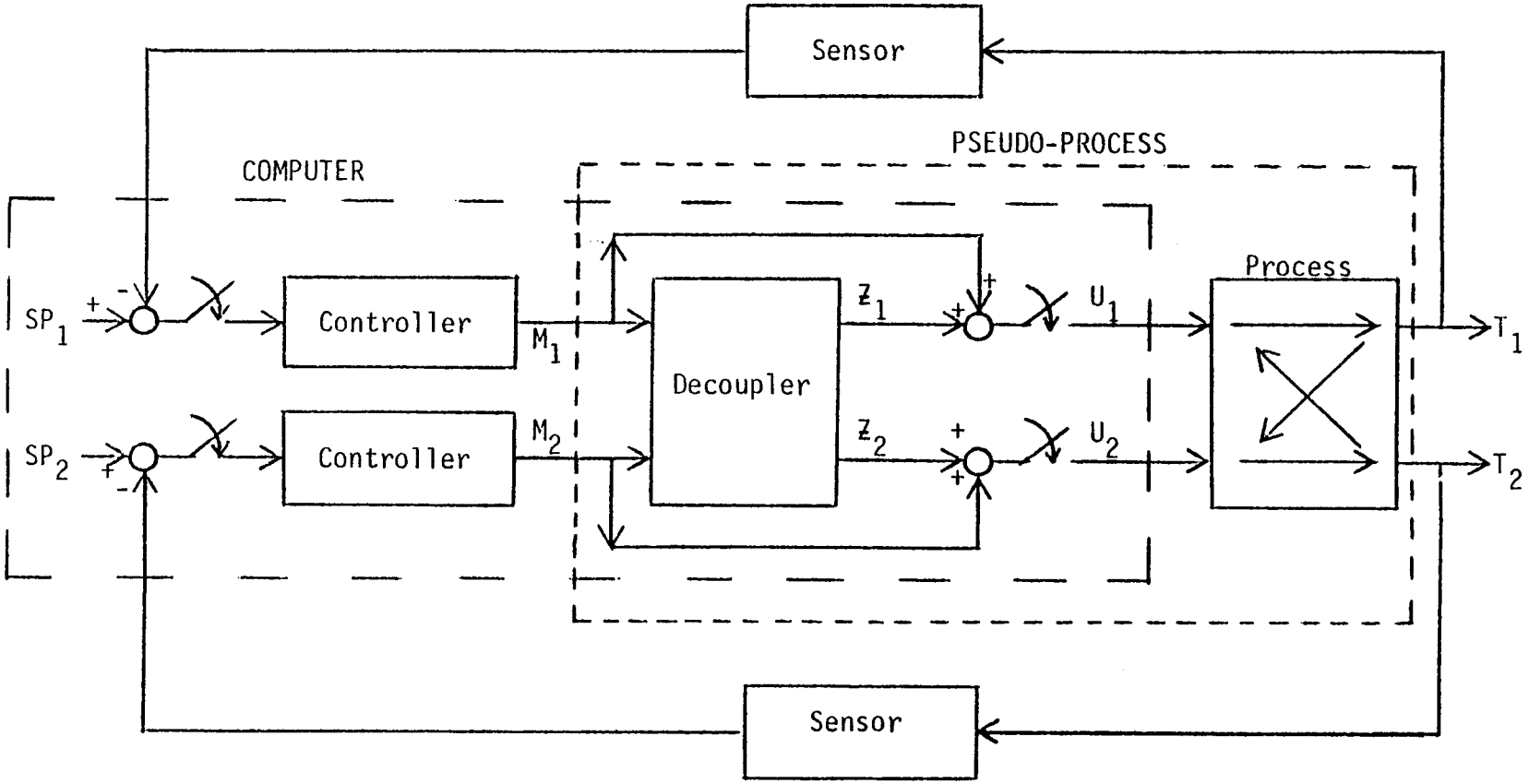


Figure I-8. Pseudo-Process Resulting from the Combination of the Decoupler and Process into One Unit.

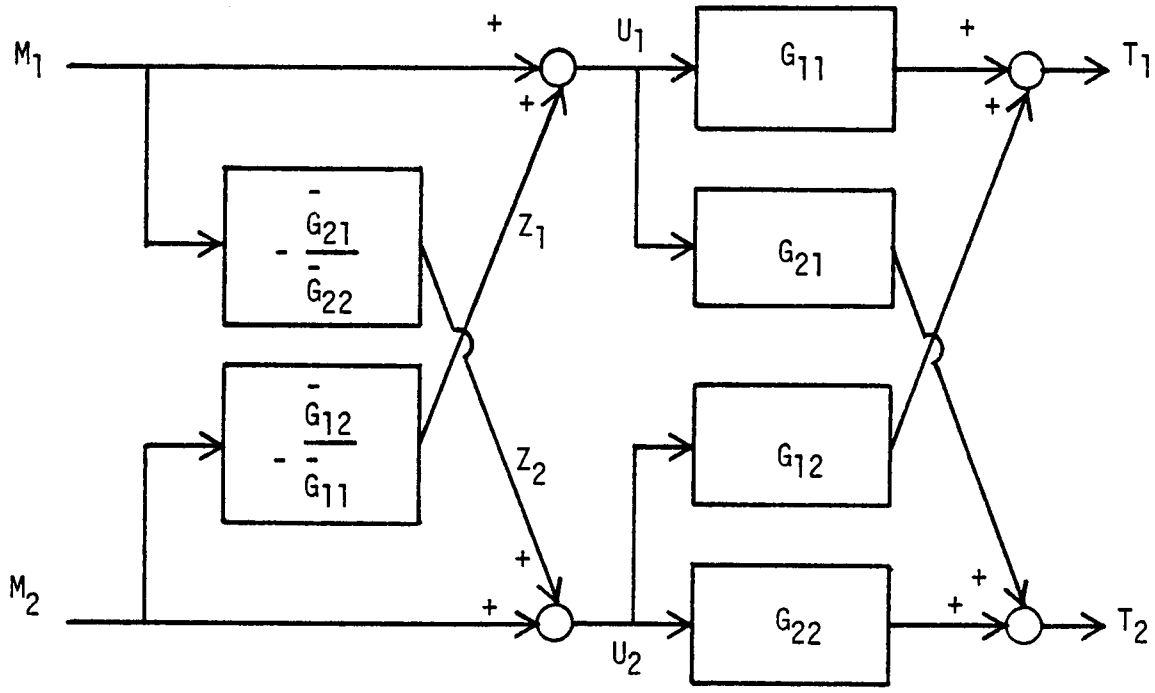


Figure I-9. P-Canonical Model with Matrix Decoupler.

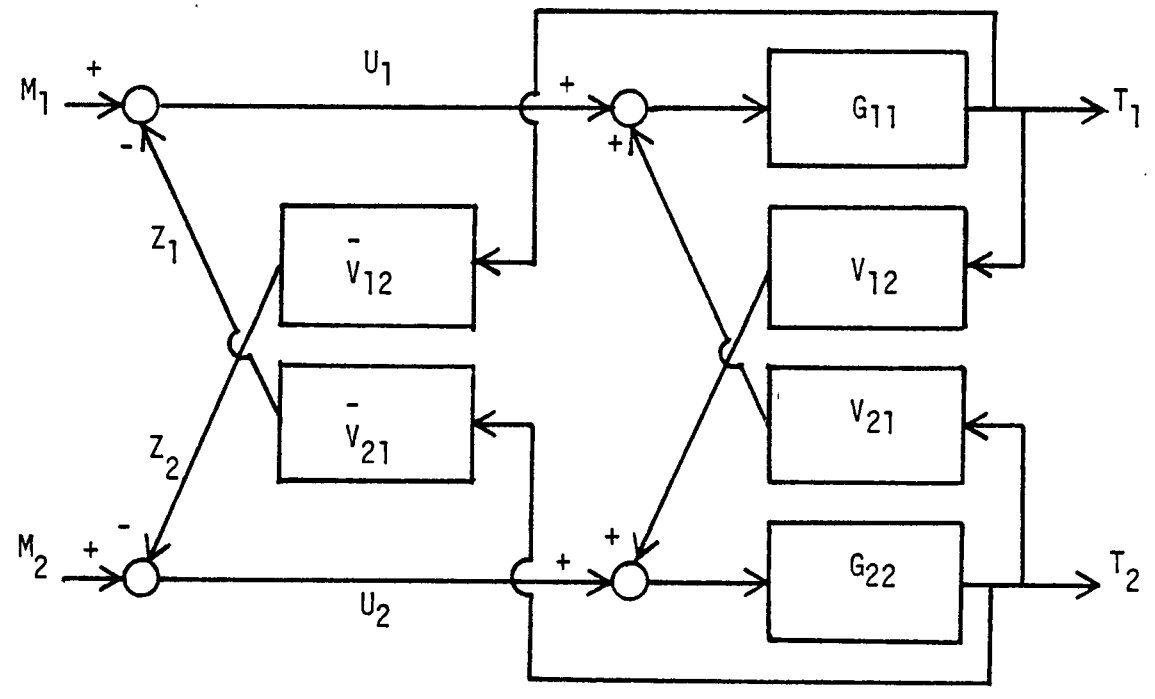


Figure I-10. V-Canonical Model with Block Diagram Decoupler.

CHAPTER II

EQUIPMENT

In previous work by the author, a data acquisition system for the calorimeter was designed and characterized. This system (depicted in Figure II-1) was designed to obtain an accuracy in thermocouple emf of less than or equal to one microvolt, the equivalent to 0.1° C. This accuracy was necessary for the calculation of heat capacity data within ± 1.0%. This resulted in the use of a suppression and high gain amplification scheme. Also, a multiplexer was essential in order to monitor all the thermocouples necessary for control purposes. Since that time, some improvements have been made in this system in order to enhance the accuracy in the sampled thermocouple signal.

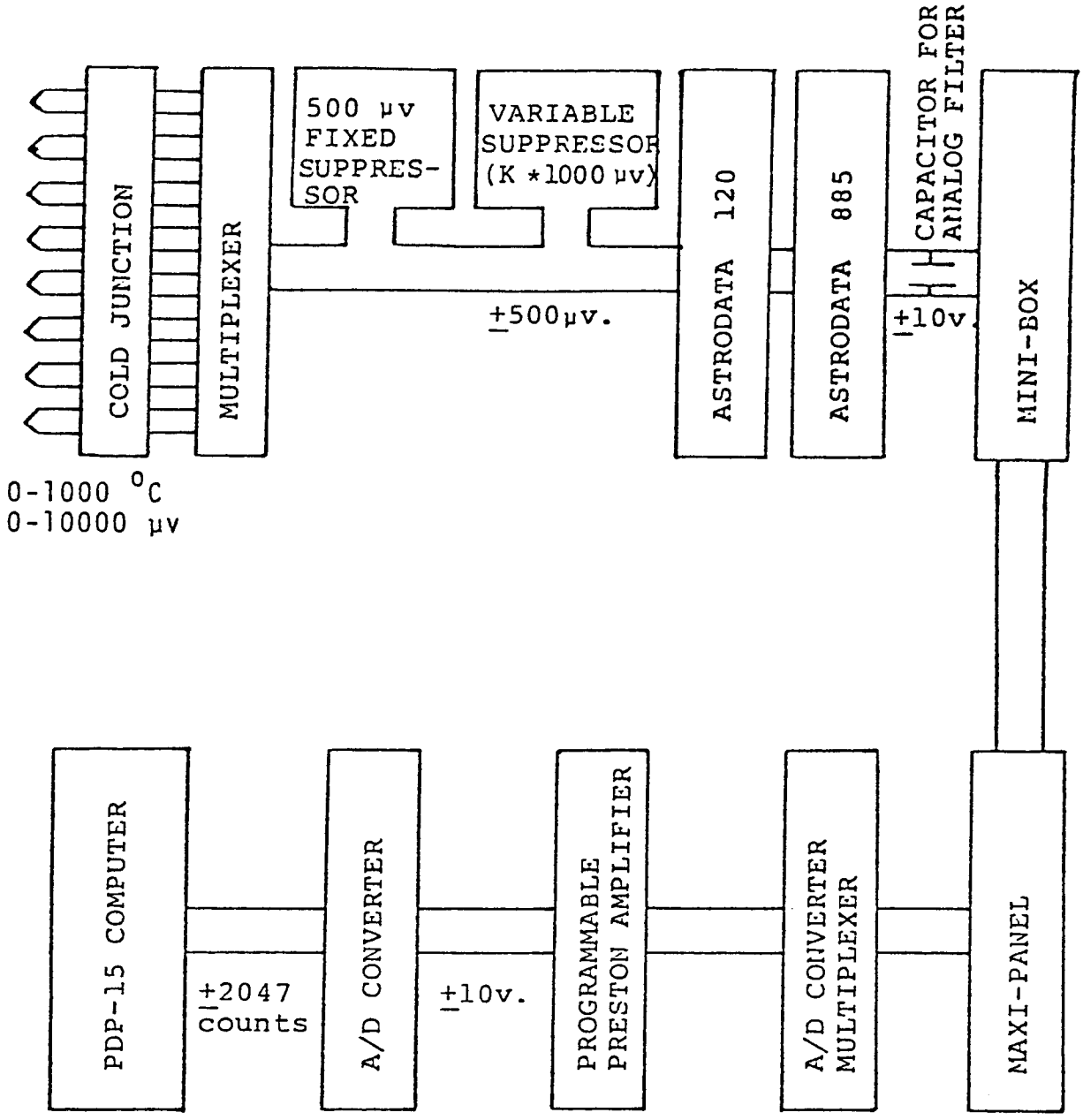
In order to complete the link of the computer to the calorimeter, the interface between the computer output and process input must be designed and installed. Since the calorimeter (Figure I-1) consists of shields powered by heaters, the power amplifiers must be capable of being driven by a DAC (Digital to Analog Converter) output.

Also, in order to assist in process modeling, a visual monitoring system on the process inputs and outputs should be installed.

The purpose of this chapter is to document the following:

1) Improvements in the data acquisition system,

2) Computer interface to process inputs, and



1 T/C μv. = 0.1°C = 0.020 volts A/DC input = 4 counts

Figure II-1. Data Acquisition Flow Diagram.

3) Process Monitoring System

Improvements in Data Acquisition System

Two improvements were made in the data acquisition system. One was the rebuilding of the eight-channel multiplexer, the other was the proper grounding of the Astrodata® 885 amplifier.

Due to poor quality relays in the multiplexer used in the previous work and a temporary design structure, noise levels of unacceptable proportions were encountered in monitoring the thermocouple signal. Therefore, a new multiplexer design and higher quality relays were used to solve this problem.

The new multiplexer (shown schematically in Figure II-2) employs a double-box construction using high quality James Micro-Scan Relays (Model Number C-2209). It consists of a small metal box surrounded by one inch thick styrofoam insulation contained inside a larger metal box. The smaller, inside box contains all the connections of the thermocouples to the relay socket pins and common output. Since relay switching is done on a microvolt level prior to amplification, this metal box is grounded in order to minimize pick-up of extraneous voltages in the air caused by short-wave radios, high frequency equipment, etc. The insulation serves as a thermal lag, as well as an electrical insulator, to prevent thermal effects at the solder joints. Also, thermal free solder was used to minimize this effect.

The improvement in the Astrodata® 885 amplifier arose from a grounding problem. This problem was characterized by sudden changes in the thermocouple signal at steady state room temperature of up to 20 microvolts. This problem was traced to intermittent loss of ground

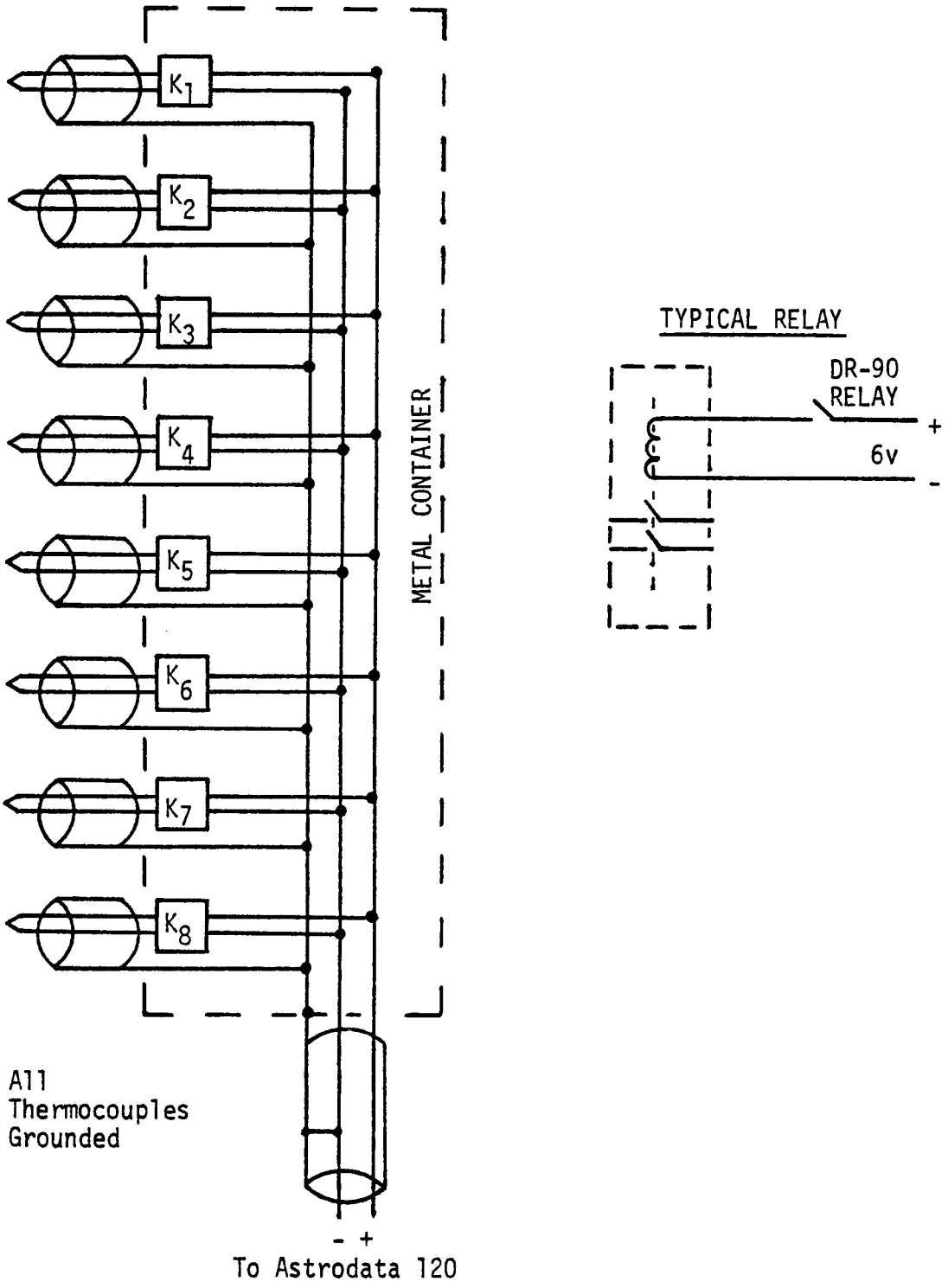


Figure II-2. Schematic Diagram of the New Eight Channel Multiplexer.

on the amplifier AC plug. The problem was solved very easily by strapping a copper wire from the case of the amplifier to ground.

In summary, these two modifications greatly enhanced the accuracy in the data acquisition system by reducing noise levels. The practical accuracy of this system has now become equal to the theoretical accuracy, which is $\pm$ 0.25 microvolts, as developed in previous work by the author (21).

Computer Interface to Process Inputs

In order to interface the computer DAC output to the power amplifiers, the DAC output must be scaled to the acceptable input of the amplifiers. Since the power amplifiers are actuated by 0-5 ma. inputs and the DAC output ranges from 0-10 VDC, it was a simple matter of installing variable resistors of approximately 20K ohms to accomplish this scaling for each of the four shields. (The complete loop from computer output to input through the process for a single loop is shown in Figure II-3.) In general, this completes the linkage of the computer to the process.

In the case of shields G3 and Cyl (Figure I-4), this installation of resistors was all that was necessary. These shields are both heated by externally wound AC heaters. In the case of G3, the 0-10 volt DAC output is converted to 0-20 watts by the power amplifier. In the case of the Cyl, this DAC output is converted to 0-1000 watts.

Special consideration had to be given to the shield G1 since it is a directly heated shield with a DC current conducted through its length. Since the thermocouple attached to this shield is grounded, a ground loop developed when the DAC (which is also grounded) was used

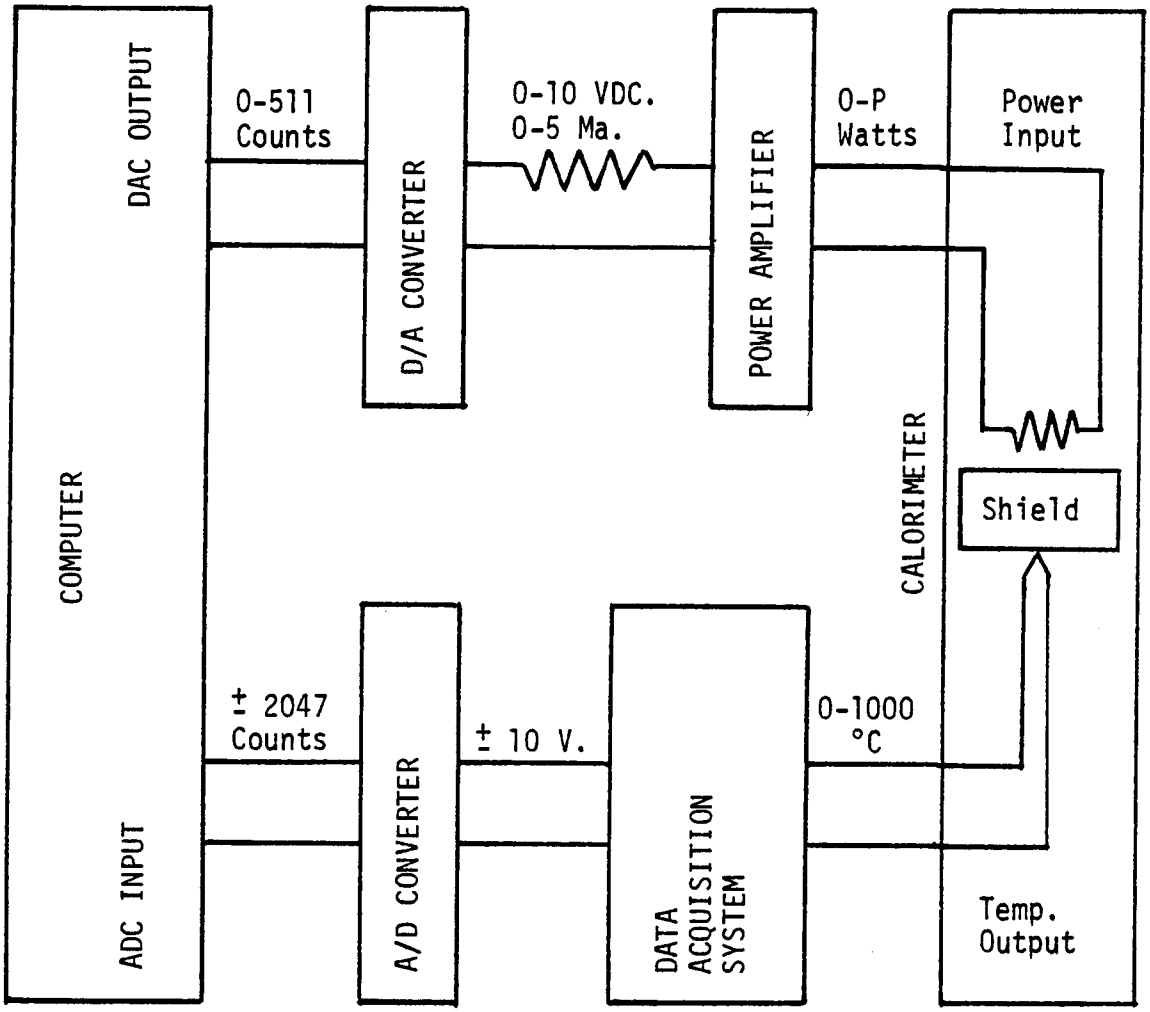


Figure II-3. Schematic Diagram of the Complete Computer Link to the Calorimeter for a Typical Shield.

to actuate the power amplifier. This loop was broken by installing a ground isolation device (Model Number 825-WM-TV, 0-10 VDC input, 0-10 VDC output, manufactured by Acromag, Inc., Wixon, Michigan). The DAC output could then be used to actuate a power level of 0-100 watts in this shield.

In the case of shield G2, the conventional analog device (an on-off controller) was replaced by continuous action from the computer. The power amplifier was interfaced to the DAC by a simple resistor (as in the case of the Cyl and G3) since it is also heated by an externally wound AC powered heater. The DAC output is converted to 0-1000 watts.

In summary, the computer output was made compatible with the input to the power amplifiers by the installation of resistors. This enabled the computer to be used in the direct digital control of the process. Also, an immediate improvement in the control of the shield G2 was accomplished by replacing the on-off analog controller by a continuous action digital controller.

Process Monitoring System

In order to have a visual system for monitoring the process, voltmeters were installed to display the DAC outputs to the power amplifiers and the error between the shield and sample temperatures.

In the case of the digital controller output in DAC volts, 0-10 VDC meters were installed. This allowed monitoring of the behavior of the controller output. This was especially helpful during modeling of the calorimeter since steady state controller outputs must be maintained prior to testing.

In the case of the error between the sample temperature and the heater temperature, voltmeters with a range of ± 10 VDC were installed. Although these meters have a range of ± 10 volts, it can be software-controlled to display an error in thermocouple emf of ± M microvolts. This is done by calculating the error in ADC counts between the shield temperature and heater temperature in microvolts and then scaling this to a DAC output of ± 10 VDC. This is done by back-calculating the effect of the data acquisition system and then scaling this value. If the error in ADC counts between the shield temperature and sample temperature is IE counts, the error in microvolts can be found by dividing by the maximum output range of the A/D converter (± 2047 counts), multiplying by its maximum input range (± 10 volts), dividing by the total gain applied in the data acquisition system (GAIN), and multiplying by 10^6 to convert absolute volts to microvolts. By dividing this result by the specified microvolt error range (IRNG) and multiplying by 511 to convert to the full output range of the DAC in counts, the error in DAC counts (M) is found to be:

$$M = \frac{IE \cdot 2.5 \times 10^6}{GAIN \cdot IRNG} \qquad 2\text{-}1$$

This allows monitoring of the error during actual operation. Without this display, there would be no way of determining how well the digital control system is working during actual operation.

In summary, the process monitoring system was installed in order to aid in process modeling and in the evaluation of the control strategy on an on-line basis. It was used to determine steady state conditions before testing during the modeling procedure. It was then

used to evaluate how well the final control strategy worked and to make necessary adjustments in the final tuning parameters.

CHAPTER III

EXPERIMENTAL PROCESS MODEL

As discussed in Chapter I, the steps leading to the final experimental model are:

1) Selection of an appropriate theoretical modeling domain,
2) Experimental testing to obtain process data,
3) Determination of the parameters in the theoretical model domain using a suitable optimization technique which will minimize the error between the model response and experimental data,
4) Development of the relationship between the optimum model parameters and temperature, and
5) Description of the overall model in such a way that it can be used in the development and tuning of an appropriate control strategy.

Review of Theoretical Process Model

In order to facilitate the understanding of the experimental model development, a short review of the theoretical model developed previously by the author (21) is required.

The calorimeter, as depicted schematically in Figure I-1, consists of a sample surrounded by four adiabatic shields. The theoretical process model for this calorimeter has its origin in nine time domain differential equations arising from energy balances on the calorimeter shields. They are of the form:

$$M_i C_{p_i} \frac{d T_i}{d \Theta} = P_i - \sum_{j=1}^{r} K_{ij} (T_i - T_j) - \sum_{h=1}^{s} K_{ih} (T_i^4 - T_h^4) \qquad 3\text{-}1$$

where M = shield mass

C_p = shield heat capacity

T = temperature

Θ = time

P = input power

K = heat transfer coefficient.

These equations relate the rate of heat accumulation in the ith shield to the rate of heat gained from the input power less the rate of heat lost by conduction $(T_i - T_j)$ and radiation $(T_i^4 - T_h^4)$. Immediately, this equation shows that the process is non-linear (due to the T^4 term), and one special control problem has now been identified.

Another useful domain in which to analyze the process is the discrete domain. It involves the approximation of the differential operator by a discrete difference, i.e.,

$$\frac{d T}{d \Theta} = \frac{\Delta T}{\Delta \Theta} = \frac{T_{n+1} - T_n}{\Delta \Theta} \qquad 3\text{-}2$$

where n = integer sampling time (0,1,---)

$\Delta\Theta$ = sampling time.

By making $\Delta\Theta$ appropriately small, the differential equations become discrete equations of the form:

$$T_{i,n+1} = T_{i,n} + \frac{\Delta \Theta}{M_i C_{p_i}} \left[P_i - \sum_{j=1}^{r} K_{ij} (T_i - T_j) - \sum_{h=1}^{s} K_{ih} (T_i^4 - T_h^4) \right]_n \qquad 3\text{-}3$$

These equations can be easily programmed and used to simulate the process once heat transfer coefficients have been specified.

Another domain which is classically used to analyze processes is the LaPlace domain By using deviations from steady state conditions (implying zero initial conditions), the differential operator is replaced by the LaPlace operator s. This transforms the differential equations into algebraic equations in terms of s. In order to do this, the T^4 term must be linearized as $4\ T_a^3(T - T_a)$ where T_a is the steady state temperature about which the equation is being linearized. When this is done, the resulting equations can be shown in block diagram form (18,19) as in Figure I-4. This diagram shows that a step input to one of the shield powers will not only affect its major output but will also be reflected into other temperature loops. (Physically, this is due to heat transfer between the hot shield to colder ones.) This phenomenon makes the process multivariable. This identifies a second special control problem.

The LaPlace domain analysis demonstrates the effect of the nonlinearity (T^4). The linearization of this term demonstrates that the process gains and time constants are functions of absolute temperature (T_a) and vary with T_a as follows:

$$K_i = \frac{1}{C_1 + C_2 T_a^3} \qquad 3\text{-}4$$

$$\tau_i = \frac{M_i\ C_{P_i}}{C_1 + C_2 T_a^3} \qquad 3\text{-}5$$

where K = process gain

τ = process time constant

C_1, C_2 = constants.

It also shows that the process gain and time constant are related by:

$$\tau_i = M_i \, Cp_i \, K_i. \qquad 3\text{-}6$$

Therefore, the ratio of the process time constant to the process gain varies only with changes in the heat capacity of the shield material and thus is approximately constant.

In summary, the theoretical analysis has shown several qualitative facts about the process. It is non-linear and multivariable. It can be modeled in any of three domains which can all be related to one another. The process parameters (in the linearized LaPlace domain) are heavily temperature dependent with a known form of functionality. Also, the process parameters are related to each other as well as to temperature.

Selection of Theoretical Modeling Domain

As discussed above, three suggested domains exist from which the final model can be determined. They are the time domain, discrete domain, and the LaPlace domain.

The time domain was abandoned from the start as it would involve the simultaneous solution of nine differential equations with 30 or 40 constants to be determined. Usually, this makes this domain unattractive as a true theoretical basis.

The discrete domain is selected on many occasions as the basis for model development since it involves the simulation of the discrete version of the time domain equations. In this case the task is to determine the heat transfer coefficients in the nine discrete energy balance relations of the form given in equation 3-3. Testing is

accomplished by collecting temperature and power data at an appropriately small sampling time ($\Delta\Theta$). The task is to then determine all the constants in all these equations using an optimization technique.

The design of a testing procedure necessary to extract all these constants becomes fairly complex upon close examination. Under normal operating conditions, all shields will be heating at a steady state rate equal to that of the sample since all shields are controlled to the sample temperature. This means that under these conditions only one constant ($\Delta\Theta/M_j\ Cp_j$) can be determined for each shield. This is due to the fact that the term in brackets in equation 3-3 will be a constant and a linear increase in T_j will be predicted (which is the correct result). Thus, the shield temperatures must be offset and the transient responses recorded.

Regardless of the complexity of testing, there are more important considerations to be made concerning this domain. First of all, absolute values of the process powers and temperatures must be measured very accurately. This is very difficult in the case of the process powers because G3, Cyl, and G2 are AC powered using chopped waveforms. This makes accurate and immediate measurement difficult. One solution is to determine a power output vs computer DAC input curve, but this entire curve changes with temperature due to the change in the heater resistance.

More serious problems arise in the measurement of the absolute value of the temperatures. This can be demonstrated by considering the effect of shutting off all power to the sample heater and keeping all shield temperatures controlled to the sample temperature. When this

is done, the term in brackets in equation 3-3 will become zero. Therefore, it predicts that the sample temperature will remain unchanged with time. This is rarely, if ever, true. The sample temperature under these conditions will drift at a slow rate, either positively or negatively, depending on thermocouple location. This is due to the fact that the shields are not isothermal but have gradients along their lengths.

Based upon these facts, the discrete domain was abandoned as a domain in which to determine the experimental model.

The domain chosen in which to model was the LaPlace domain. It affords tremendous advantages over the two previous domains. The first major advantage is the manner in which manipulations for data collection can be made. By placing the process controllers in manual, direct manipulation of the process inputs can be made rather than having to manipulate temperature offsets as in the discrete model. Also, all responses can be measured in terms of deviations since this is a basic premise of the LaPlace domain derivations; therefore, ΔP and ΔT can be measured without the problems associated with the accurate measurements of absolute values. But the greatest advantage lies in the fact that these changes or deviations can be measured in terms of variables which are representative of the process variables (P, T) rather than the actual variables themselves. Also, a black-box approach can be used, and this greatly simplifies the model.

Experimental Testing Procedure

By examining more closely the methods of obtaining data and by conducting a few preliminary tests, many simplifications were made in

the testing procedure and even in the model itself.

First of all, step testing involves the manipulation of the process input parameters and recording the response in the process outputs. (A simple representation of this can be seen in Figure III-1.) This can be done in terms of deviations from steady state and on a black-box basis. In general, the transfer functions obtained are in the P-Canonical form of the model and can be represented by the general equation:

$$X = [G]\,U \qquad 3\text{-}7$$

where X = vector of process outputs

U = vector of process inputs

G = matrix of transfer functions.

For a simple 2x2 system, this leads to:

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}. \qquad 3\text{-}8$$

By manipulating U_1 and recording the response in X_1 and X_2, G_{11} and G_{21} can be determined. Similarly, a change in U_2 results in the determination of G_{12} and G_{22}. It should be noted here that these are lumped transfer functions and represent the process in exactly the same way as does Figure I-4 but with one very important difference: it is no longer necessary to determine each and every block in the complex block diagram in Figure I-4.

Since modeling in the LaPlace domain is based upon deviations from steady state values, variables other than power and temperature can be used in the modeling procedure. A simplified schematic of the link between the computer and the calorimeter is shown in Figure III-2

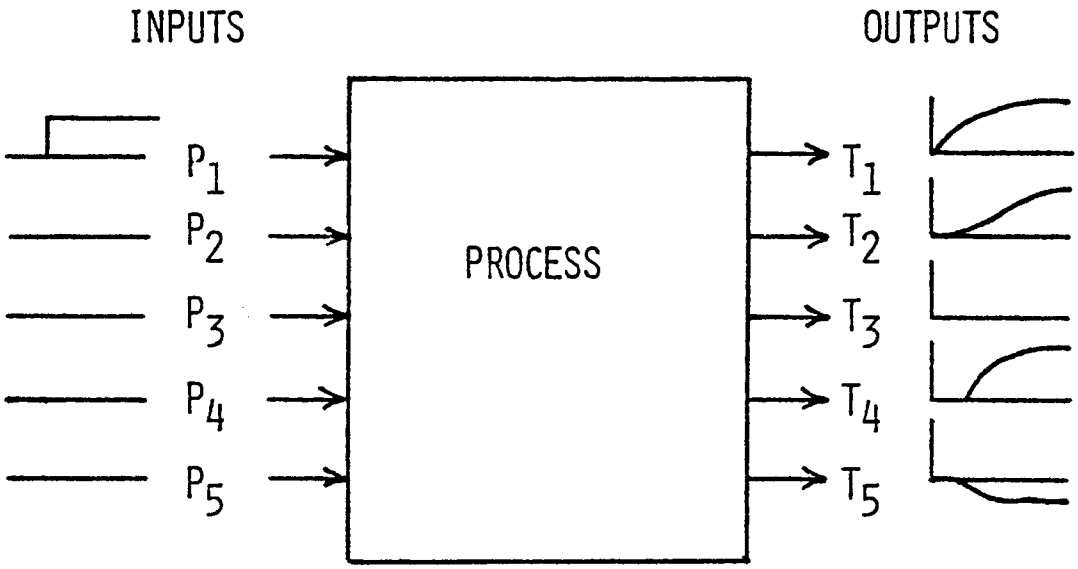


Figure III-1. Modeling Procedure for a Multivariable Process Using Step Testing.

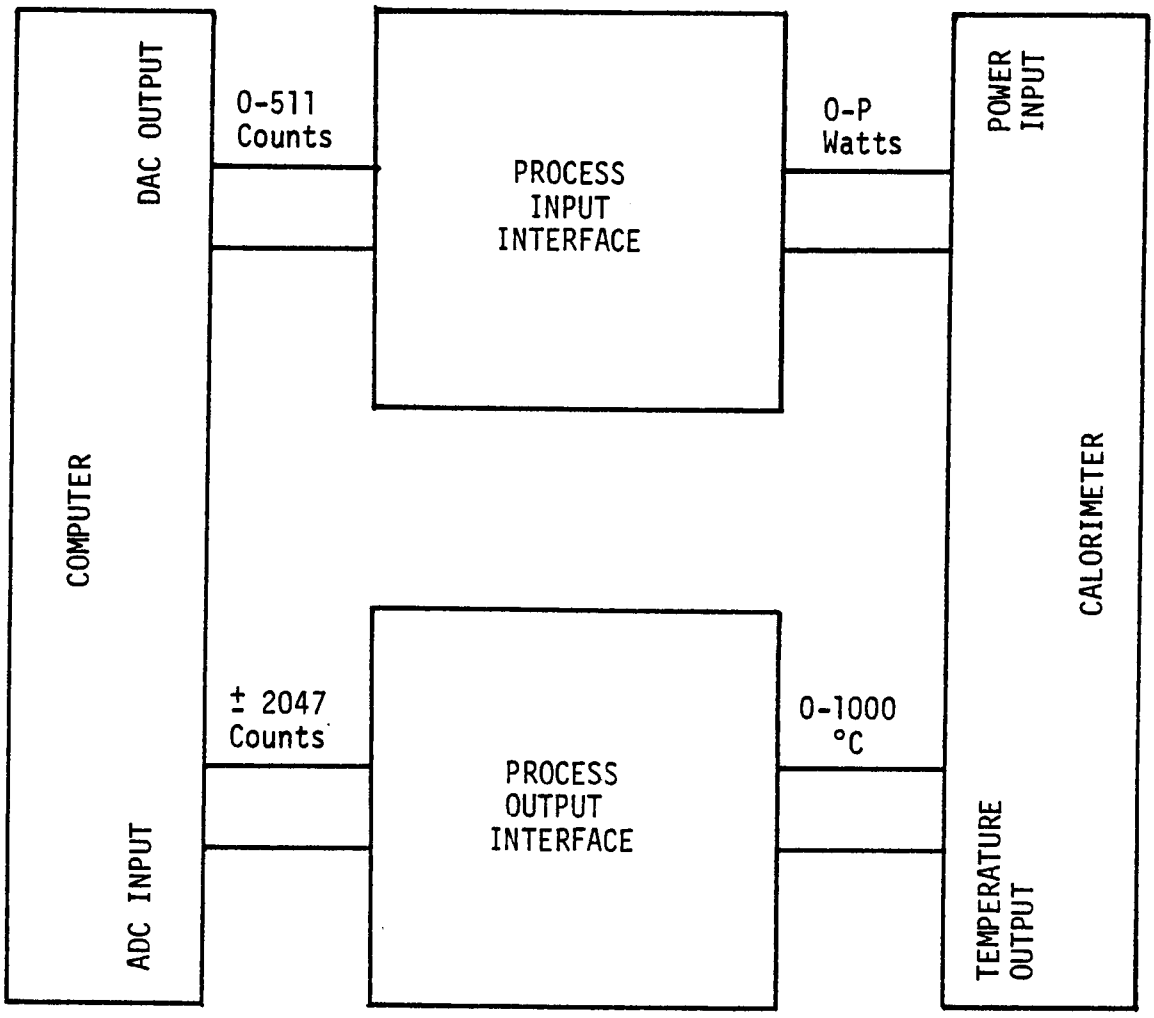


Figure III-2. Overall Flow Diagram of Process from Computer Output to Computer Input.

and demonstrates that the DAC output is representative of the power input and the ADC input is representative of the temperature output. Since the final controller will be a computer algorithm dealing with ADC counts as inputs and DAC counts as output, it is unnecessary to convert the computer input to temperature, execute the control algorithm to determine output powers, and convert these powers to a DAC output. By manipulating the DAC output and recording the response in ADC counts, all these conversions are unnecessary. This method also insures that all parameters which will affect the final controller algorithm design and tuning will be taken into account in one overall transfer function. Therefore, in order to test the process, step testing will be employed using DAC counts as the process input and ADC counts as the process output.

Another consideration to be made is what exactly is the parameter to be modeled and subsequently controlled. Under actual operating conditions, the controlled parameter is the difference between the shield temperature and the sample temperature; therefore, this should be the parameter to be modeled. (Appendix A contains a more detailed explanation of why this is true, but a short explanation based upon the process response will be given here.) Figure III-3 depicts the time response of a step to the power of any shield H. Under steady state conditions, the power to the sample (P_S) will be constant and the sample temperature (T_S) will be increasing linearly with time. Under controlled conditions, the shield power (P_H) will also be constant, and the shield temperature (T_H) will be increasing at the same rate as T_S since the difference between T_H and T_S is controlled

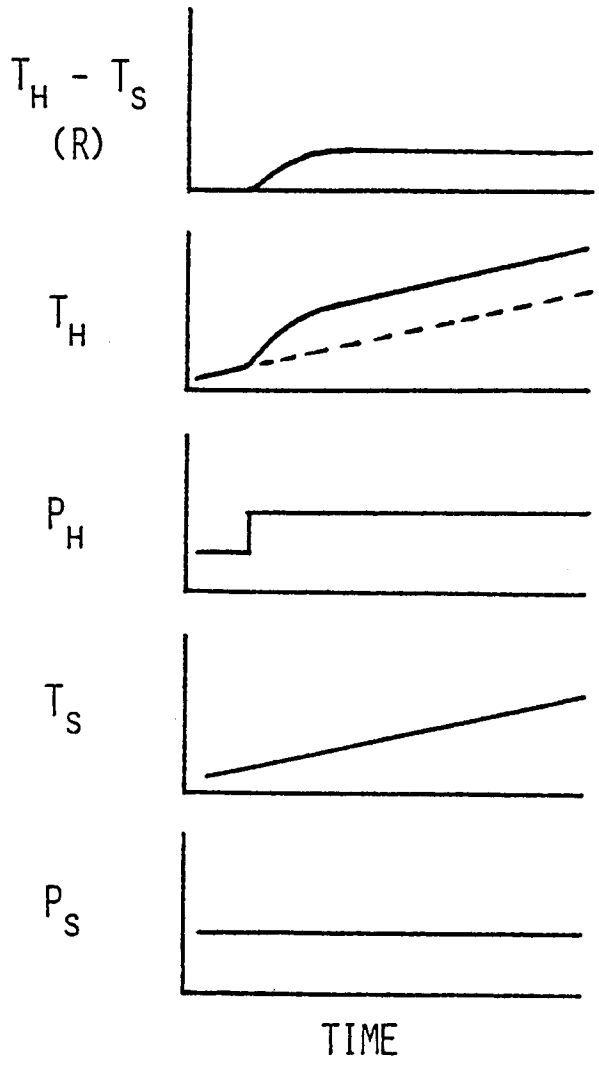
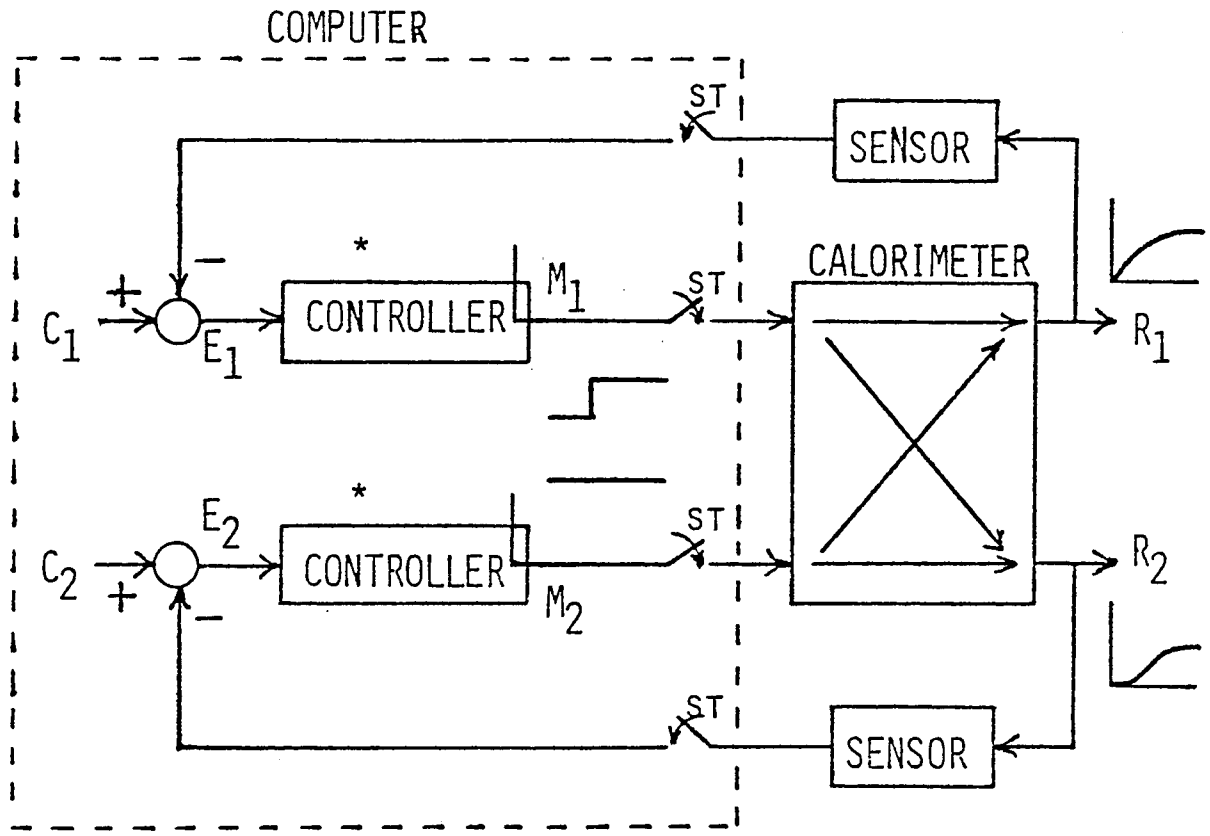


Figure III-3. Experimental Response of a Shield during a Typical Step Test.

to a constant value (usually zero). When a step is made in P_H, T_H will begin to rise above its normal value. As this shield becomes hotter than its surroundings, this excess power will be cancelled by the increased heat loss; and the shield will return to its original heating rate but at some new offset between the shield and sample temperature. If T_S is then subtracted from T_H, the response will appear as shown for $T_H - T_S$ which is denoted by the variable R ($R = T_H - T_S$). As previously stated, this offset ($T_H - T_S$) is the controlled variable and should be the modeled variable.

One final problem was encountered in developing the modeling procedure. Normally, step testing is conducted by placing all the controllers in manual and making a step in one input at the time (as depicted in Figure III-4). Then all responses are recorded and subsequently modeled. Since modeling on the calorimeter was done under dynamic conditions using hand-tuned digital controllers, this procedure produced data that was totally useless. When all controllers were placed in manual and no steps were issued, all the shield temperatures began to slowly drift. This drift would be negligible compared to a step input but would lead to erroneous conclusions regarding the effects of interactions. This was particularly true in the case of the shields Cyl and G2 since they lose heat to the outer water-cooled jacket of the calorimeter. As their temperature increases, the water-cooled jacket remains at room temperature; therefore, more and more heat must be supplied to maintain the same shield heating rate. Thus, placing the controller in manual causes the shield temperature to begin to fall below the sample temperature at a slow rate. (The



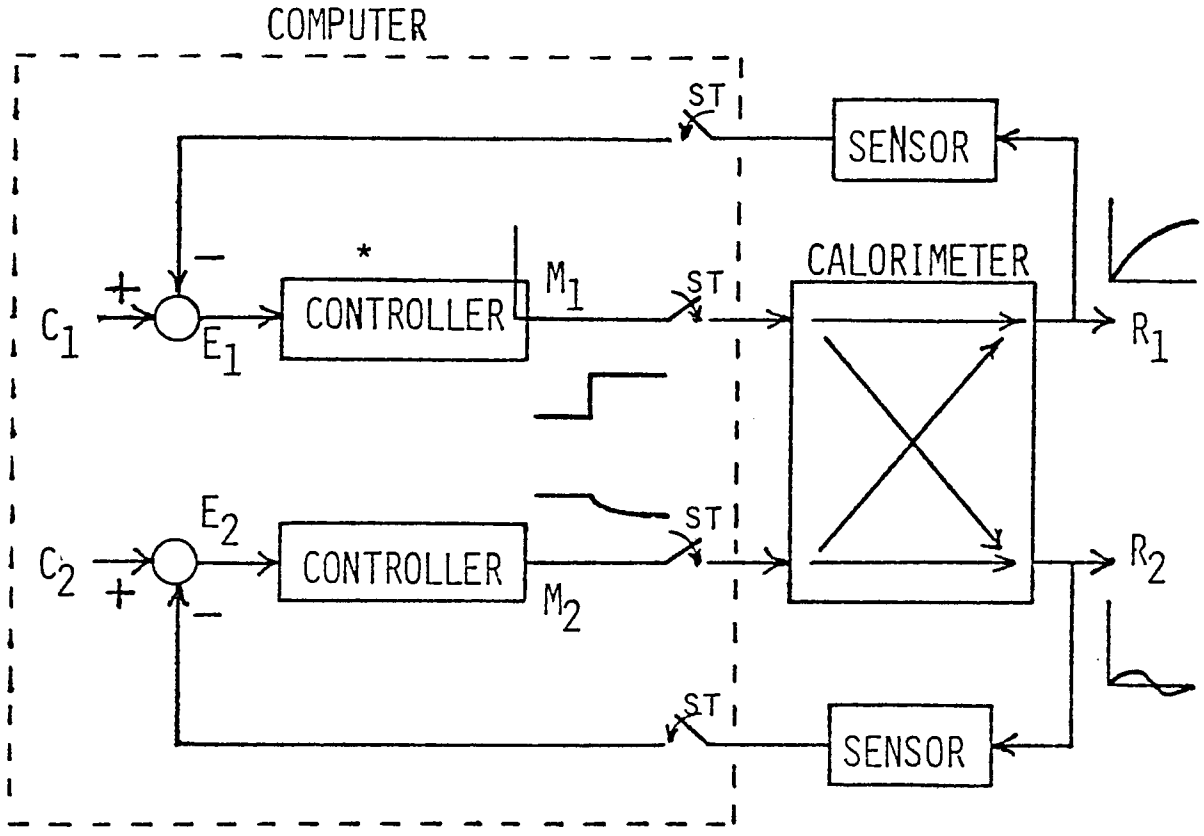
*Controller in Manual Mode.

Figure III-4. Standard Modeling Technique with All Controllers in Manual.

question might be raised concerning this effect on the actual step test data for the major shield being tested, but it is easily realized that this effect is part of the overall response and is accounted for in the black-box approach.)

Due to the above effect, a modified step testing procedure (depicted in Figure III-5) was employed. This procedure requires that only the controller to which the step is to be issued be placed in the manual mode while all others are left in automatic. This will produce temperature responses and controller actions in the loops left in automatic as shown in Figure III-5. This complicates determination of the interaction transfer functions, but they can be determined using the method in Appendix B.

These facts led to a testing procedure which consisted of the following principles. Step tests were conducted by issuing steps to various controller outputs while under digital control using hand-tuned PID algorithms. Testing was done under dynamic conditions with all controllers in automatic except the one to be stepped. Once the step was issued in DAC counts, all controller outputs were stored in terms of DAC counts along with all shield and sample temperatures in terms of ADC counts. The data were stored on Dectapes. After a run was completed, these data were used to obtain the shield responses by calculating and outputting the difference between each shield temperature and the sample temperature (in ADC counts) as well as the controller outputs (in DAC counts). These responses were then entered into the DEC-10 to be analyzed.



*Controller in Manual Mode.

Figure III-5. Special Modeling Technique with All Controllers in Automatic Except One to be Stepped.

Qualitative Analysis of Process Responses

Figures III-6, III-7, III-8, and III-9 show typical responses for shields G3, G1, Cyl, and G2, respectively, to step inputs to their powers. As stated previously, the response is in terms of ADC counts. It should be noted here that an increase in one ADC count is equivalent to an increase in temperature of 0.025° C; thus, all steps were small on a temperature basis. Also, it should be noted that the sampling time for all testing was 2.5 seconds. This is important to note since all points are not plotted for better clarity, and it might appear that a higher sampling time was used.

It is obvious from Figures III-6 and III-7 that G3 and G1 are characterized by first order lag plus dead time dynamics. The general form of this response is well known (18,20). The LaPlace domain equation (in terms of the LaPlace operator S) for this response is:

$$\frac{X(S)}{U(S)} = \frac{K\, e^{-\Theta S}}{1 + \tau S} \qquad 3\text{-}9$$

where $X(S)$ = process output

$U(S)$ = process input

K = process gain

Θ = process dead time

τ = process time constant.

Therefore, the response curve can be characterized by optimizing three process parameters--K, Θ, and τ. This can be done by graphical techniques or computer curve-fitting.

Figures III-8 and III-9 show that shields Cyl and G2 are characterized by second order lag plus dead time dynamics. The general form of this response is also well known (18,20) and is easily

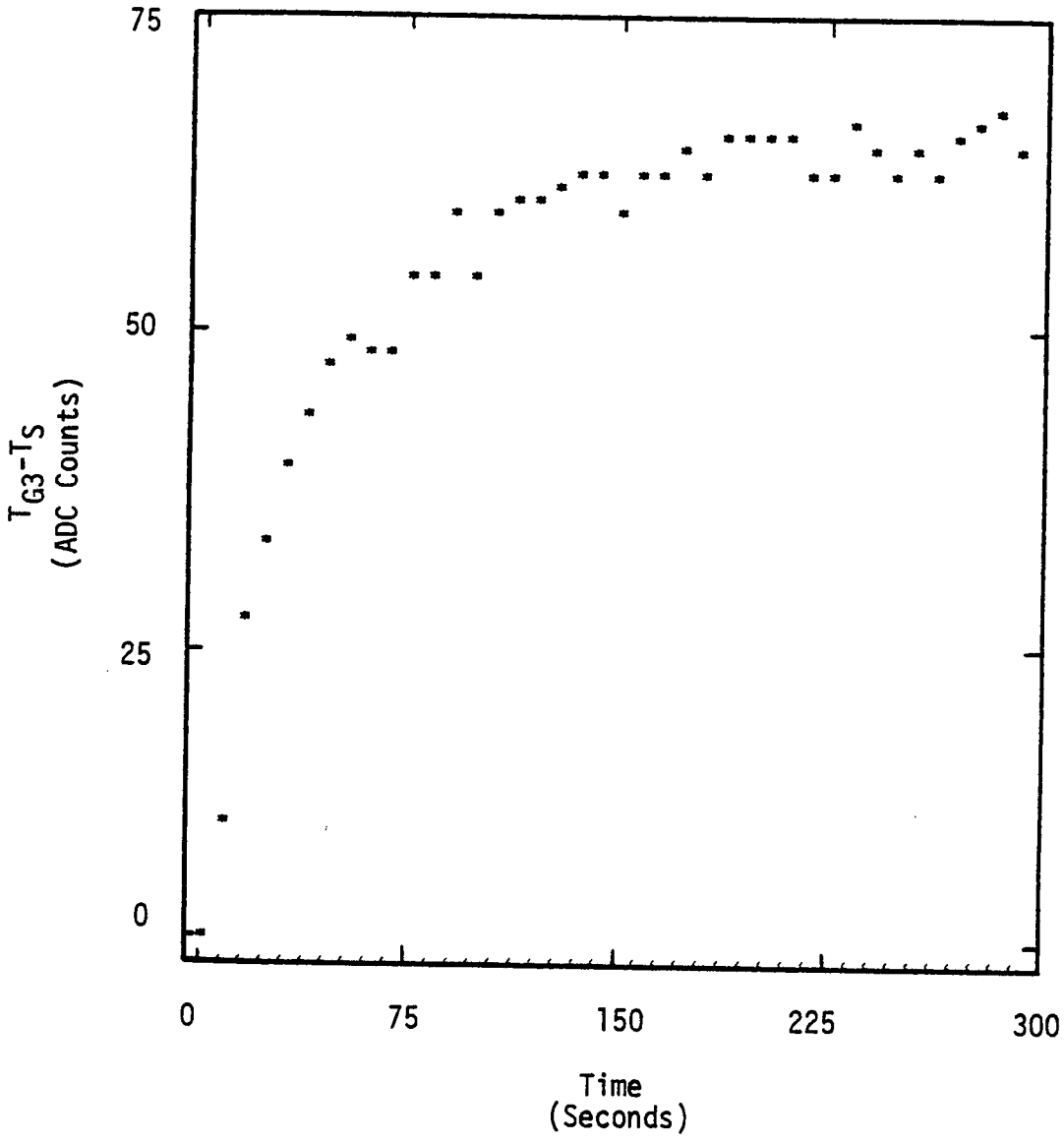


Figure III-6. Typical Step Response for the Shield G3.

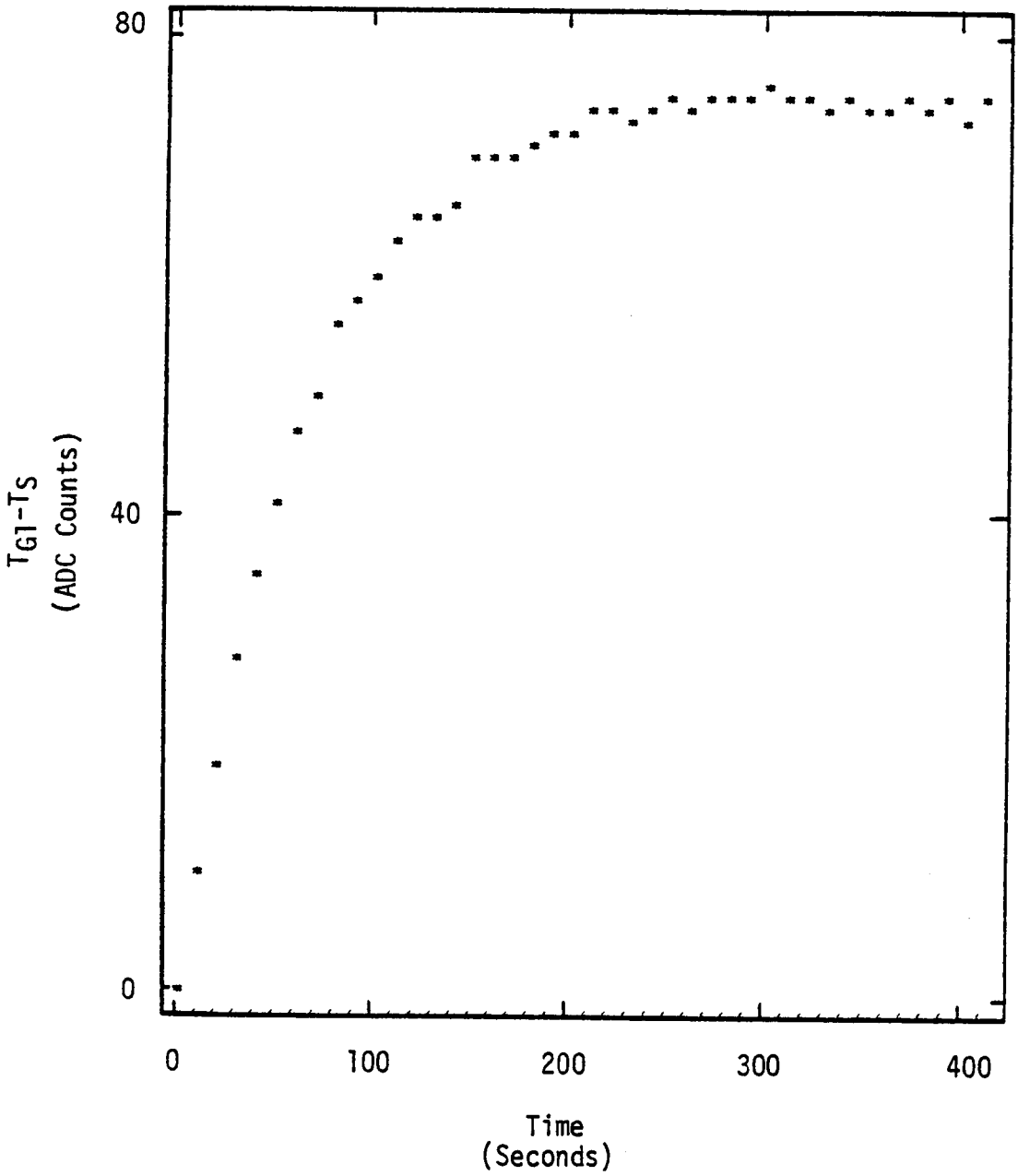


Figure III-7. Typical Step Response for the Shield G1.

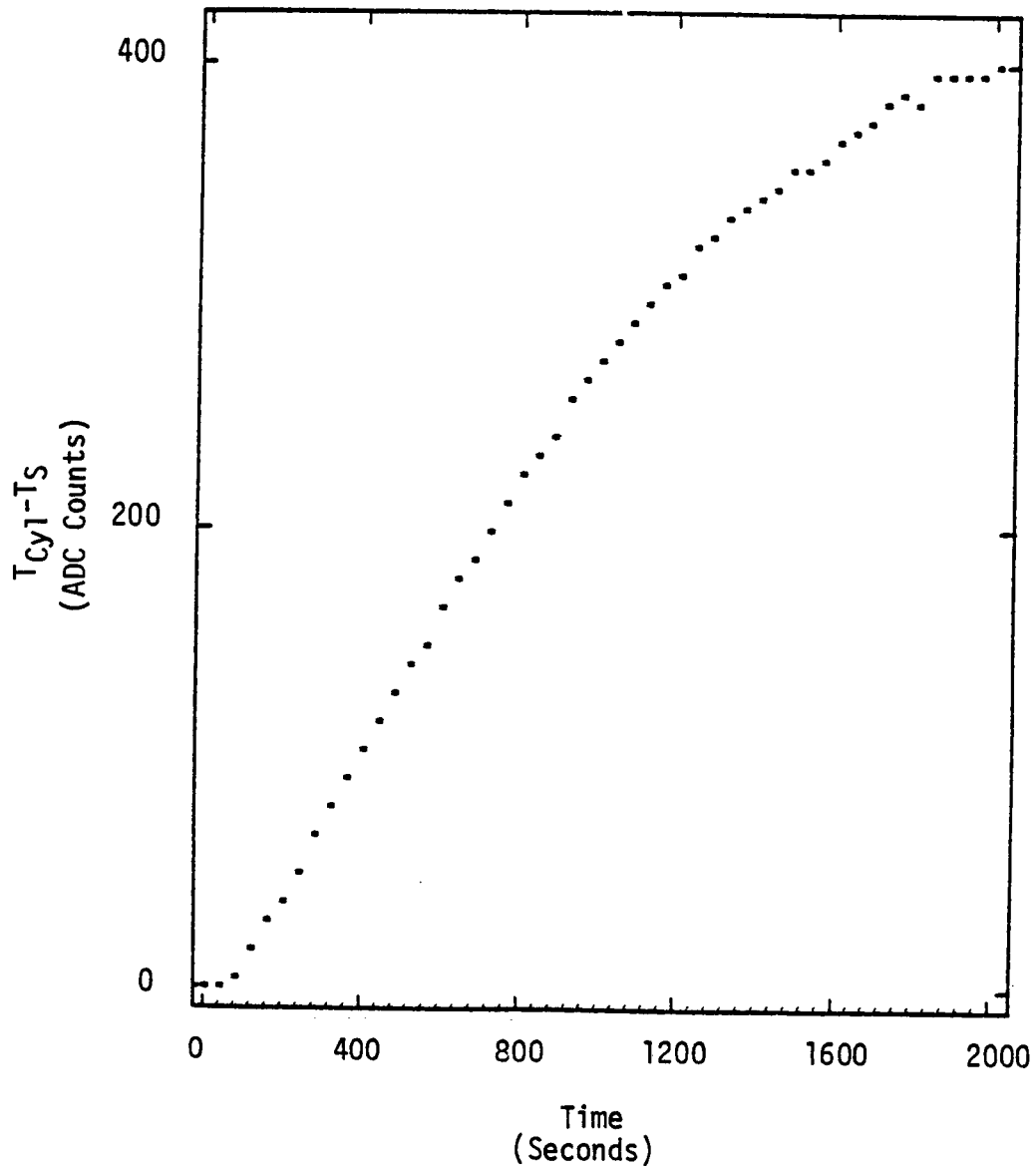


Figure III-8. Typical Step Response for the Shield Cyl.

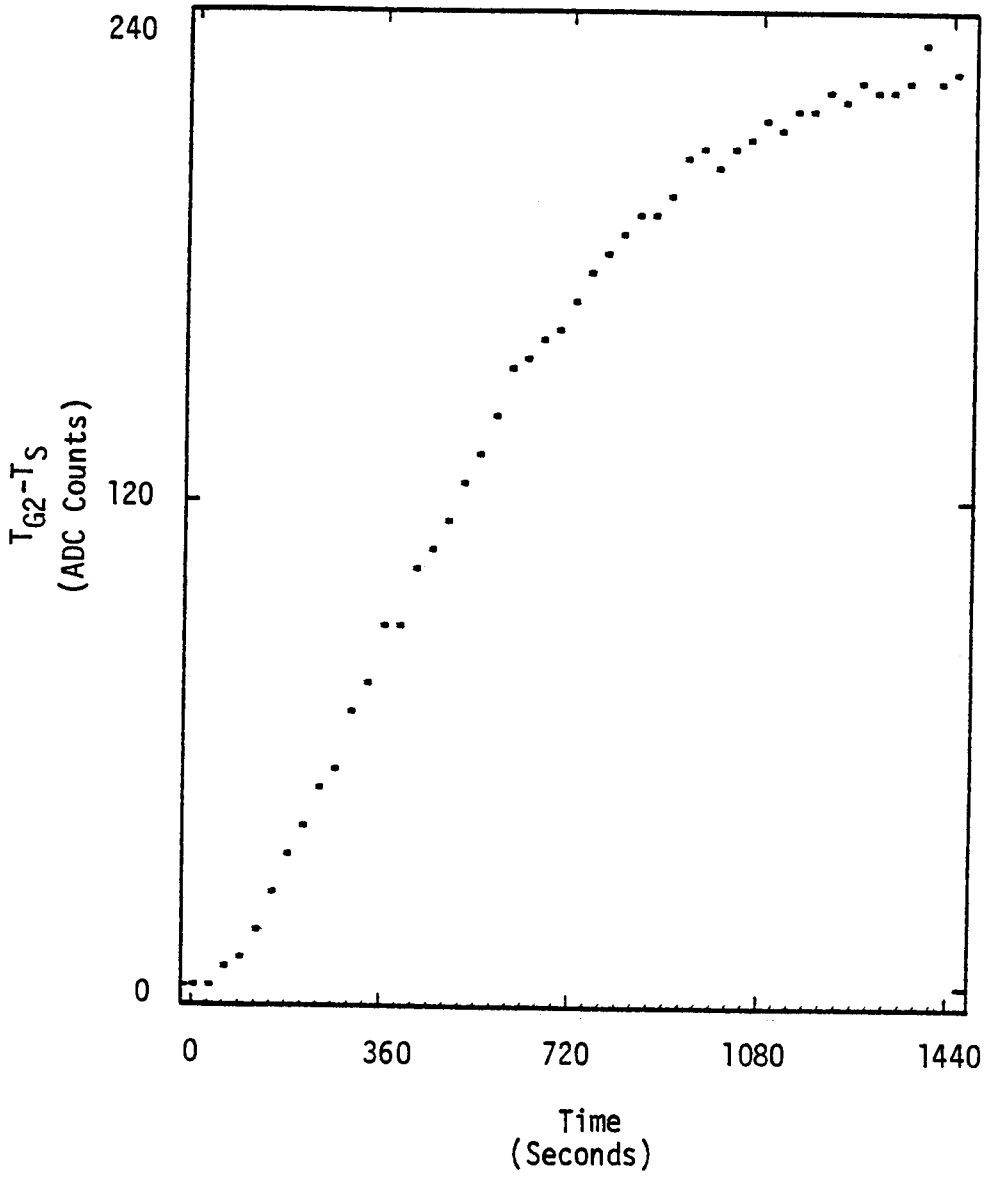


Figure III-9. Typical Step Response for the Shield G2.

recognized by the typical "S" shaped curve. This response is given by:

$$\frac{X(S)}{U(S)} = \frac{Ke^{-\Theta S}}{(1 + \tau_1 S)(1 + \tau_2 S)} = \frac{Ke^{-\Theta S}}{1 + \frac{2\xi}{W_n} S + \frac{1}{W_n^2} S^2} \qquad 3\text{-}10$$

where K = process gain

Θ = dead time

τ_1, τ_2 = process time constants

ξ = damping ratio

W_n = process frequency.

It is obvious that an equivalency of these equations leads to the relationships:

$$\xi = \frac{\tau_1 + \tau_2}{2\sqrt{\tau_1 \tau_2}} \qquad 3\text{-}11$$

$$\frac{1}{W_n} = \sqrt{\tau_1 \tau_2} \; . \qquad 3\text{-}12$$

It is advantageous to use τ_1 and τ_2 in parameter determination but ξ and W_n in data presentation; therefore, these relationships must be developed. Physically, this second order response arises from two first order lags occurring in series. One is due to the heater (nichrome wire encased in a ceramic holder and wound around the shield), and one is due to the shield itself. The parameters in this model are best determined by computer curve fitting.

One interesting note is that the first order shields have fast responses and come to steady state in less than five minutes while the second order shields are very slow and come to steady state in 30-60 minutes. This is expected since the time constant is proportional to

the mass of the shield (as seen in equation 3-5). Shields G3 and G1 are of low mass (both are very thin, 0.010 inches) while the Cyl and G2 are very thick (0.375 inches) and therefore much more massive. This causes the slower response in the Cyl and G2.

After characterizing the four major transfer functions, there remain twelve interaction transfer functions to be characterized. When steps were made, no interactions existed except between the Cyl and G1. All others had such low gains or long time constants that they were undiscernible. The Cyl-G1 interaction occurred only in one direction. Figure III-10 shows the response in the shield G1 along with its controller output during the step test of the Cyl in Figure III-8. As the Cyl responded, it radiated heat to G1 and caused its controller output to drop, but G1 remained in control all during this test. Although an error must occur for the controller to respond, the hand-tuned PID controller used in modeling performed so well that no appreciable response in G1 error can be viewed. This was the only observable interaction.

It can be shown in Appendix C that, by using the V-Canonical form of the model, the interaction between the Cyl and G1 can be characterized by a gain term only. This fact is also seen in Figure I-4 since all the signals reflected from one loop to another pass through gains only. Also, if the dynamics of the closed loop system into which the disturbance enters are fast enough, as they are in this case, the interaction gains can be determined by direct calculation rather than simulation as suggested in Appendix B.

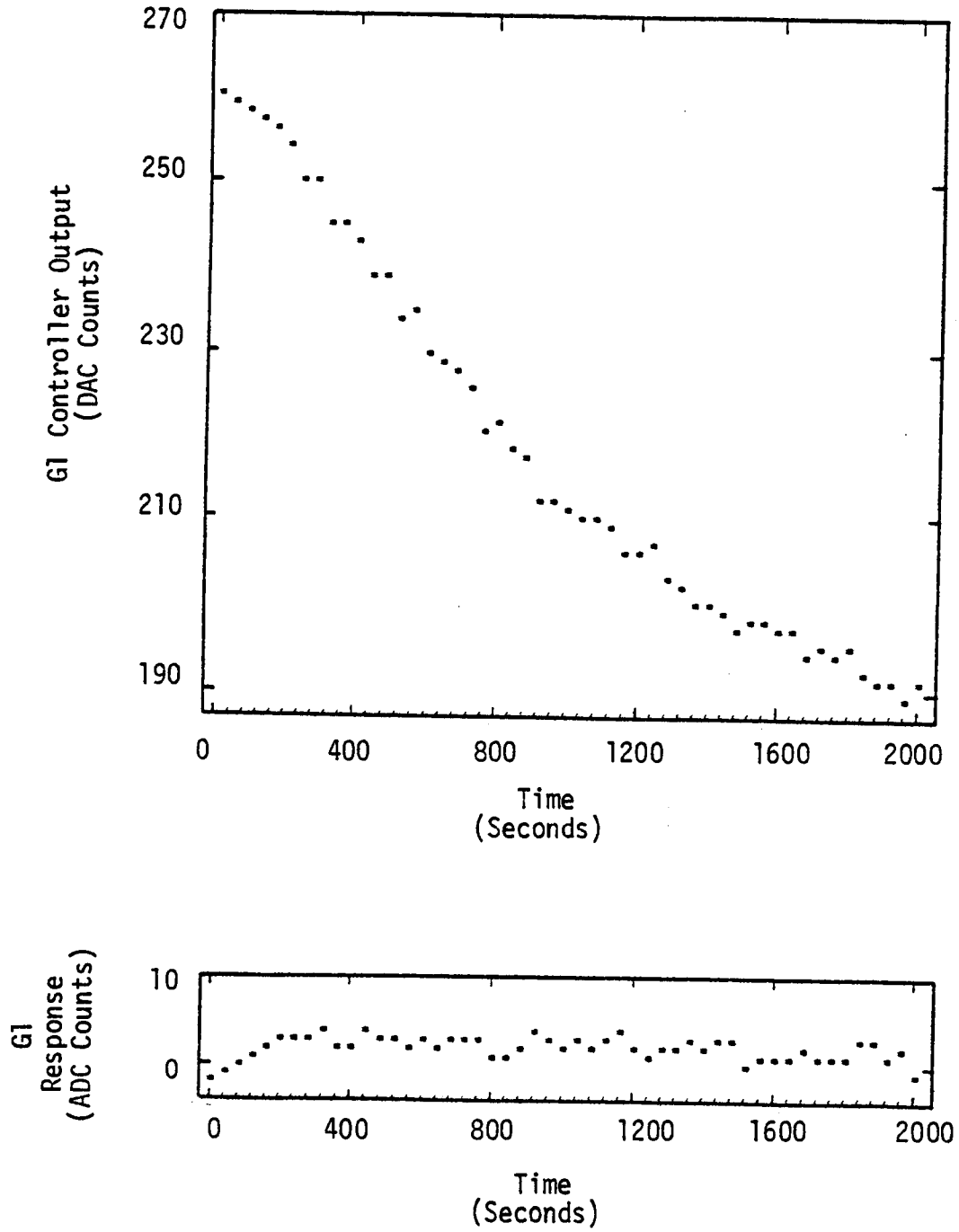


Figure III-10. Effect on the Response and Controller Output of the Shield G1 During the Step Test of the Shield Cyl in Figure III-8.

Quantitative Analysis of Process Response

The first order process parameters (K, Θ, and τ) for shields G3 and G1 were determined by computer curve-fitting using the z-transform of equation 3-9 in series with a zero-order hold. This produces a recursive digital equation of the form:

$$X_i = \alpha X_{i-1} + K(1-\alpha)\, U_{i-N-1} \qquad 3\text{-}13$$

where i = sample number

N = Θ/ST; ST = digital sampling time

$\alpha = e^{-ST/\tau}$.

The three process parameters (K, Θ, τ) were optimized by simulating equation 3-13 and using a PATTERN Search technique (30) which systematically varies all three parameters until the sum of the squares of the deviations between the experimental data and the curve-predicted data becomes a minimum. The data sets in Figures III-6 and III-7 are presented again in Figures III-11 and III-12 with their computer-fitted curves and values for K, Θ, and τ. All the first order curves for shields G3 and G1 were analyzed in this manner.

Figures III-13, III-14, and III-15 show the process parameters for shield G3 plotted against thermocouple emf which is proportional to temperature. (The various data points were determined from several different runs. The conditions and symbols for these figures and all similar figures for the remaining shields are given in Table I.) The data show that the process gain (K) and time constant (τ) do decrease with increasing temperature as predicted by equations 3-4 and 3-5. (The plots were made against emf because it eliminates the unnecessary conversions to actual temperatures.) These variables were curve-fitted

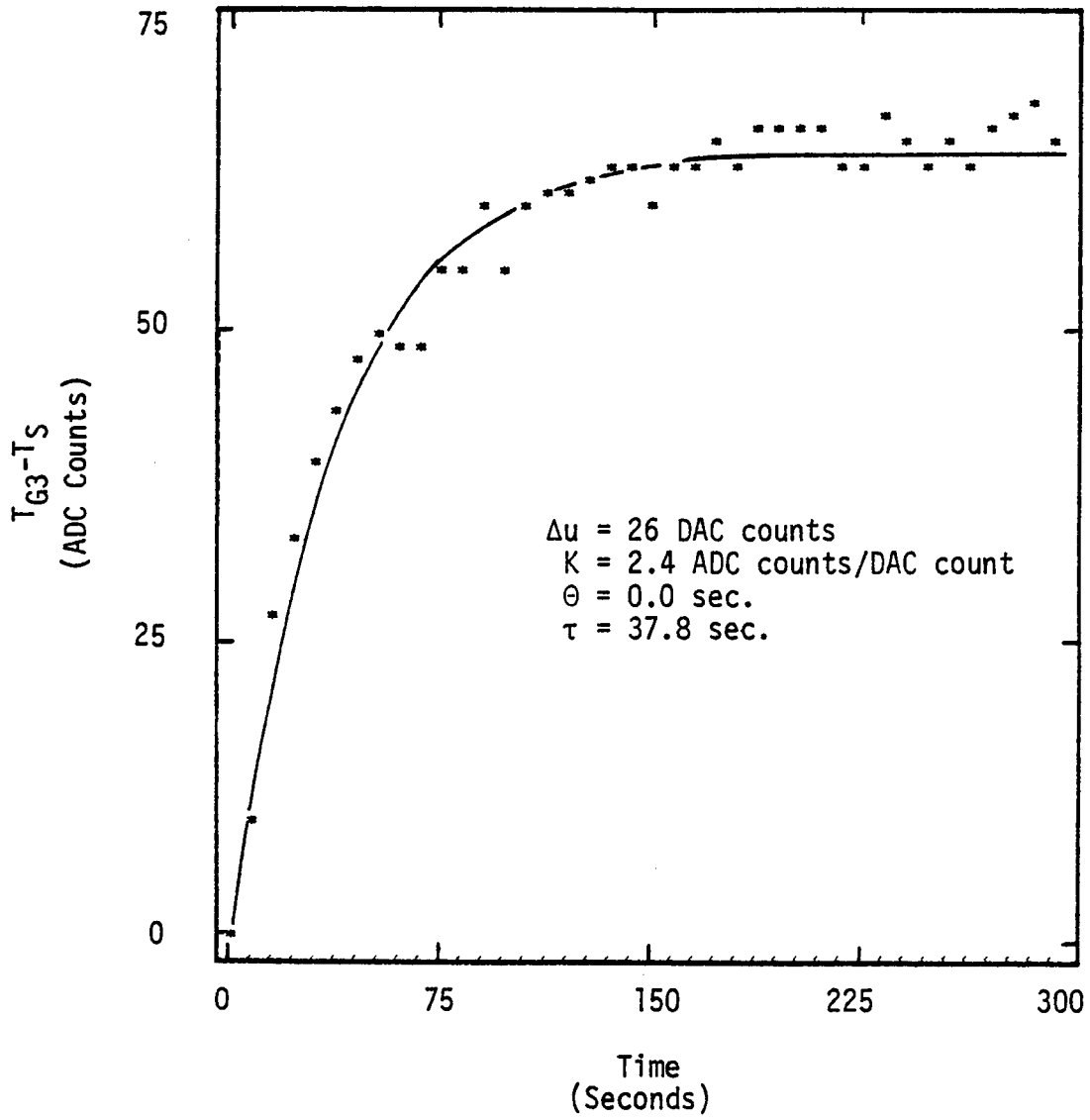


Figure III-11. Step Response for the Shield G3 from Figure III-6 with the Fitted Curve and Process Parameters.

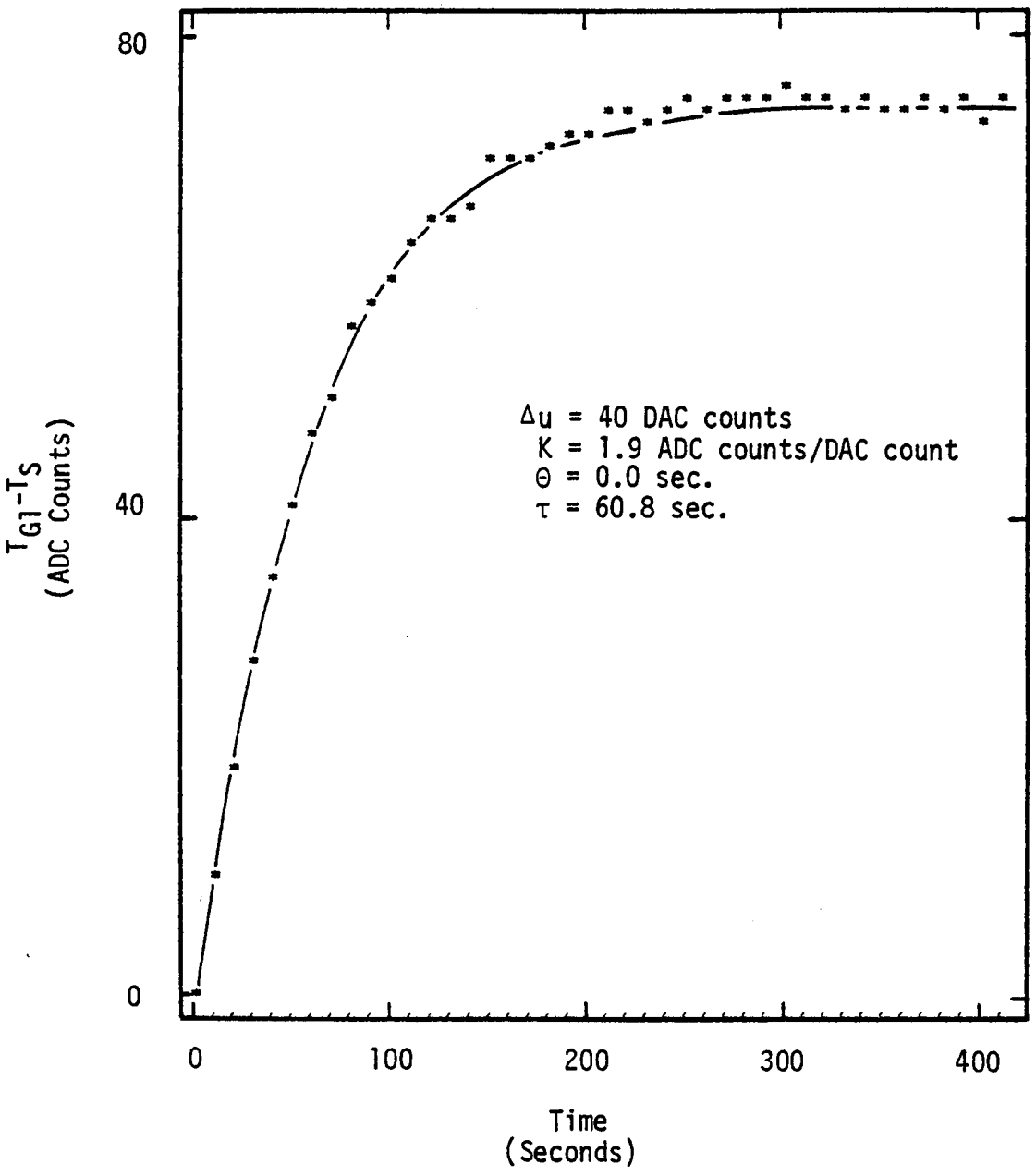


Figure III-12. Step Response for the Shield G1 from Figure III-7 with the Fitted Curve and Process Parameters.

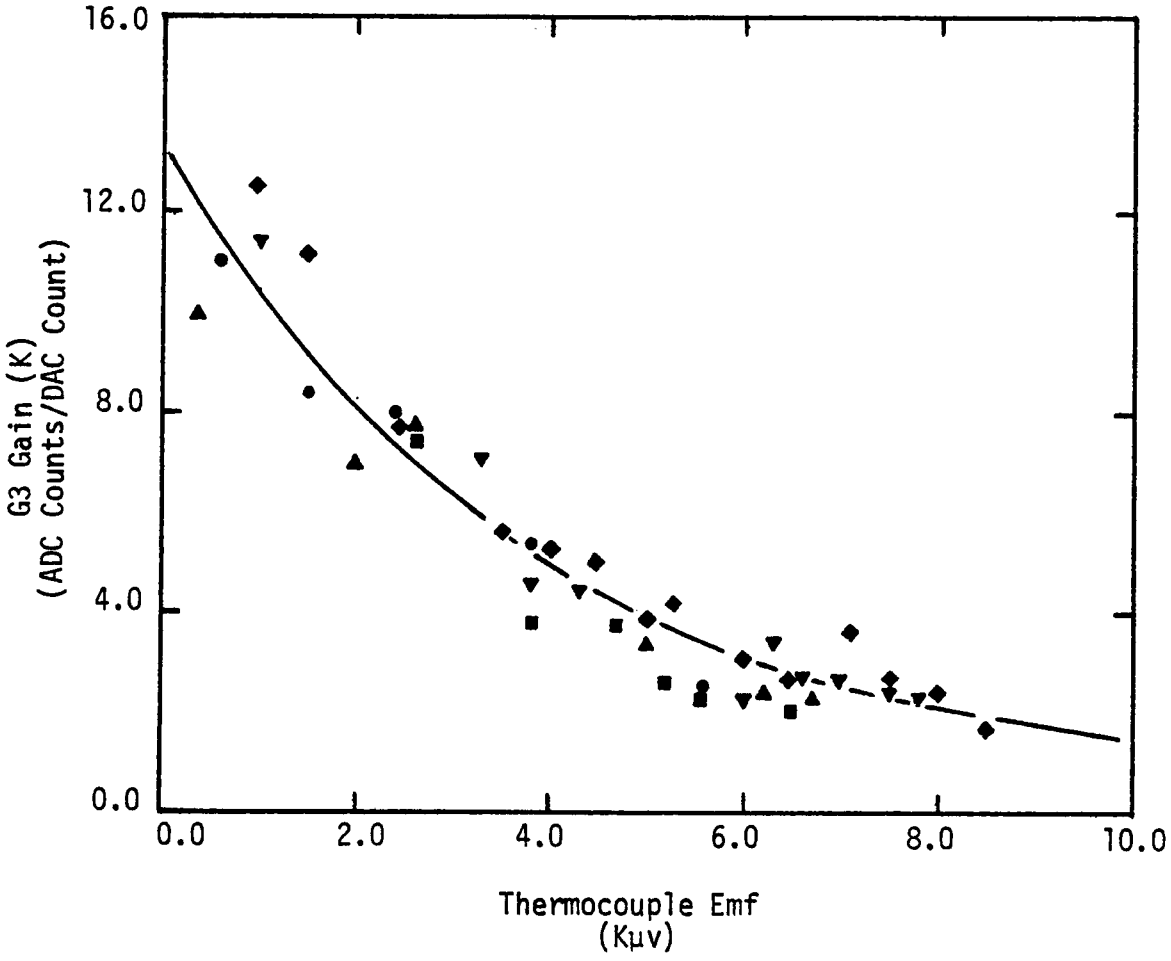


Figure III-13. G3 Gain vs Thermocouple Emf.

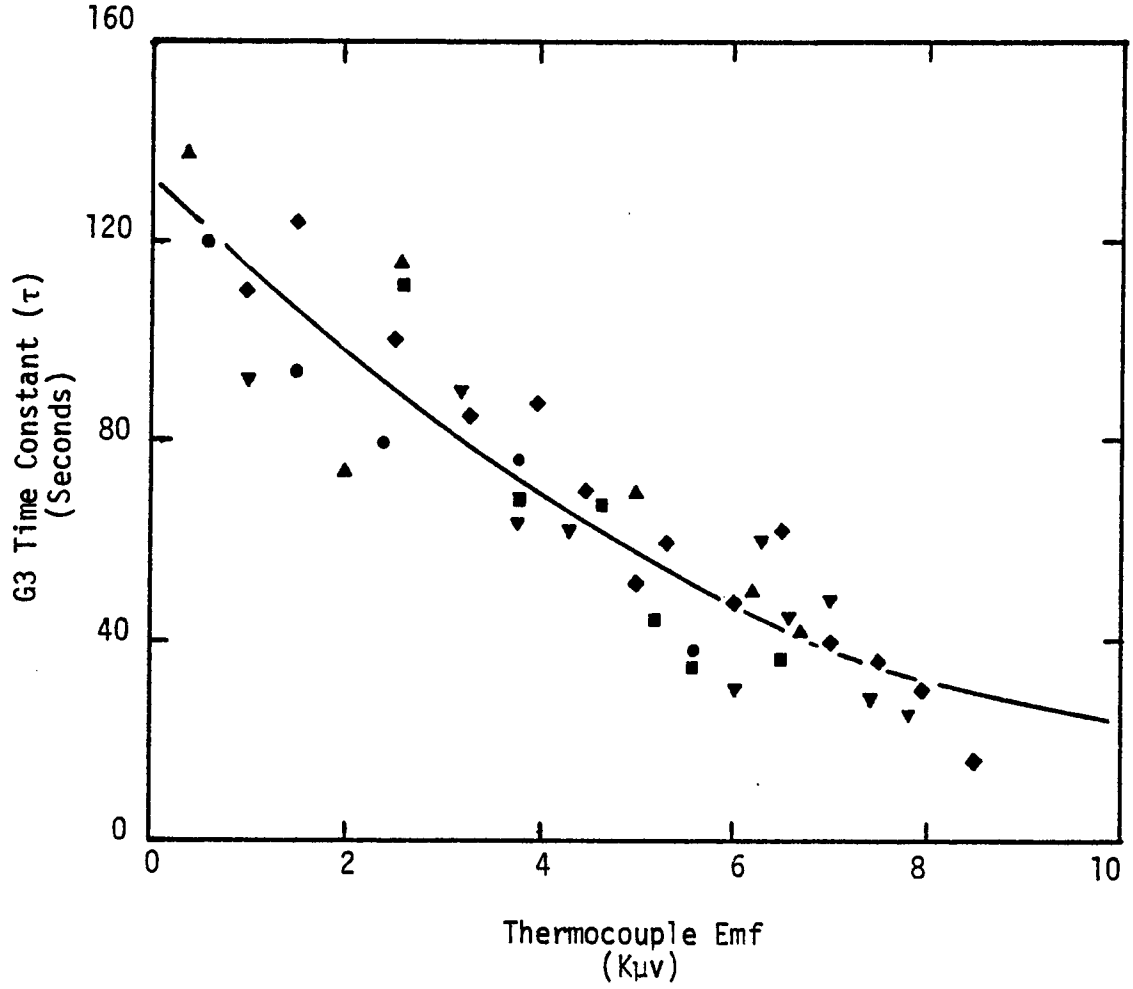


Figure III-14. G3 Time Constant vs Thermocouple Emf.

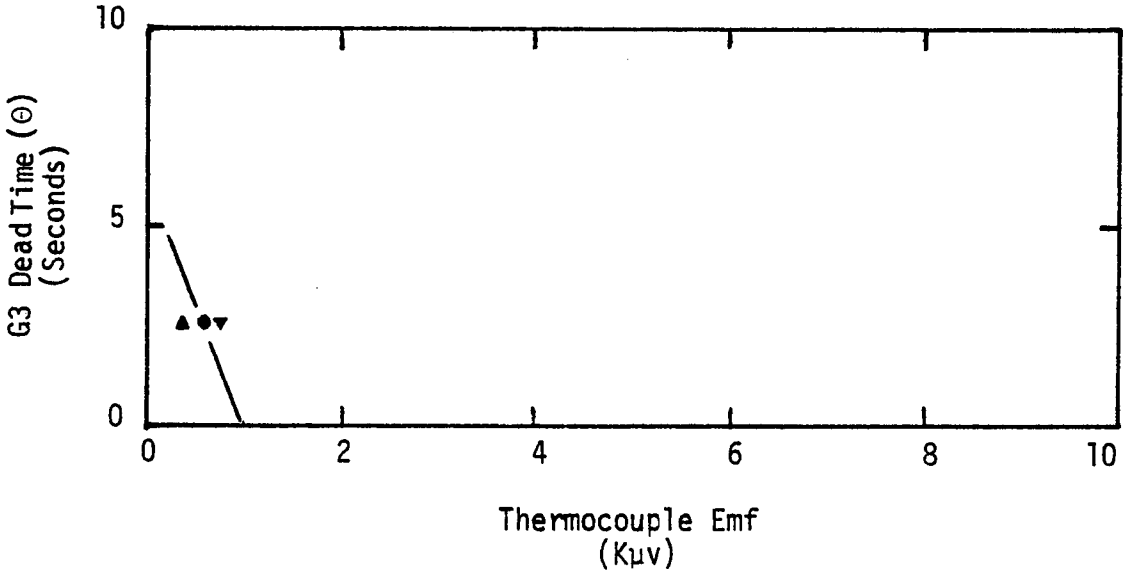


Figure III-15. G3 Dead Time vs Thermocouple Emf.

TABLE I

CONDITIONS AND LEGEND FOR MODELING RUNS

Run	Date	Plotting Symbol
I	1/11/78	●
A	4/6/78	■
B	4/14/78	▲
C	7/7/78	▼
D	7/11/78	⊙
E	8/1/78	◆

using PATTERN (30) to a modified form of equation 3-4:

$$V = \frac{A}{B + T^3} \qquad 3\text{-}14$$

where V = process variable (K or τ)

T = absolute temperature (°K)

A, B = constants.

Since the data were plotted against thermocouple emf, it was also fitted to the form:

$$V = \frac{A'}{B' + (E + C')^3} \qquad 3\text{-}15$$

where E = thermocouple emf

A', B', C' = constants.

These constants appear in Table II for the shield G3, and the curves are plotted along with the data in Figures III-13 and III-14. It should be noted that this form fits the data very well. The dead time for G3 (Figure III-15) drops very rapidly to zero by 1000 μvs. Here an average value was used up to 1000 μvs and a value of zero thereafter. (All values equal to zero were not plotted.)

In the theoretical model, equation 3-6 predicts that the ratio of the time constant to the gain should be a constant which slightly increases with temperature due to changes in heat capacity. This ratio (calculated from curve predicted values of K and τ) appears in Figure III-16. It shows that the theoretical model has once again correctly predicted the characteristics of the process parameters.

Figures III-17, III-18, and III-19 show the same first order parameters for shield G1. The gain (K) and time constant (τ) were also curve-fitted for this shield using equations 3-14 and 3-15. (A good fit

TABLE II

CONSTANTS FOR TEMPERATURE AND THERMOCOUPLE EMF DEPENDENCY EQUATIONS
FOR SHIELDS G3, G1, CYL, AND G2

Shield	Parameter	Temperature		Thermocouple Emf		
		A	B	A'	B'	C'
G3	K	2.96×10^{9}	2.06×10^{8}	5.34×10^{12}	2.53×10^{11}	5329
	τ	5.61×10^{10}	4.10×10^{8}	1.06×10^{14}	6.35×10^{11}	5567
G1	K	9.78×10^{8}	0	6.50×10^{11}	0	2598
	τ	-	-	1.03×10^{13}	0	1200
Cyl	K	9.69×10^{10}	3.59×10^{9}	1.95×10^{14}	7.03×10^{12}	5898
	$1/W_n$	6.79×10^{11}	9.86×10^{8}	1.27×10^{15}	1.71×10^{12}	5540
G2	K	1.71×10^{10}	1.00×10^{9}	2.73×10^{13}	1.52×10^{12}	4802
	$1/W_n$	2.00×10^{11}	4.42×10^{8}	3.51×10^{14}	6.64×10^{11}	5258

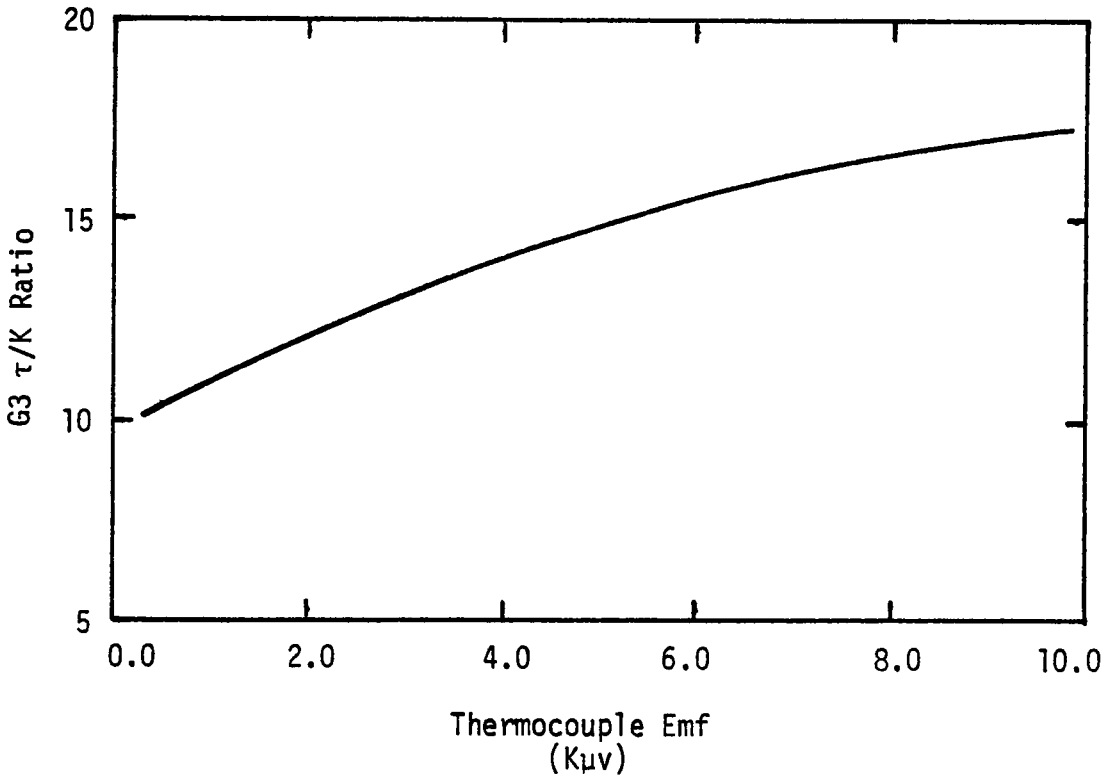


Figure III-16. Ratio of Time Constant to Gain for Shield G3 vs Thermocouple Emf.

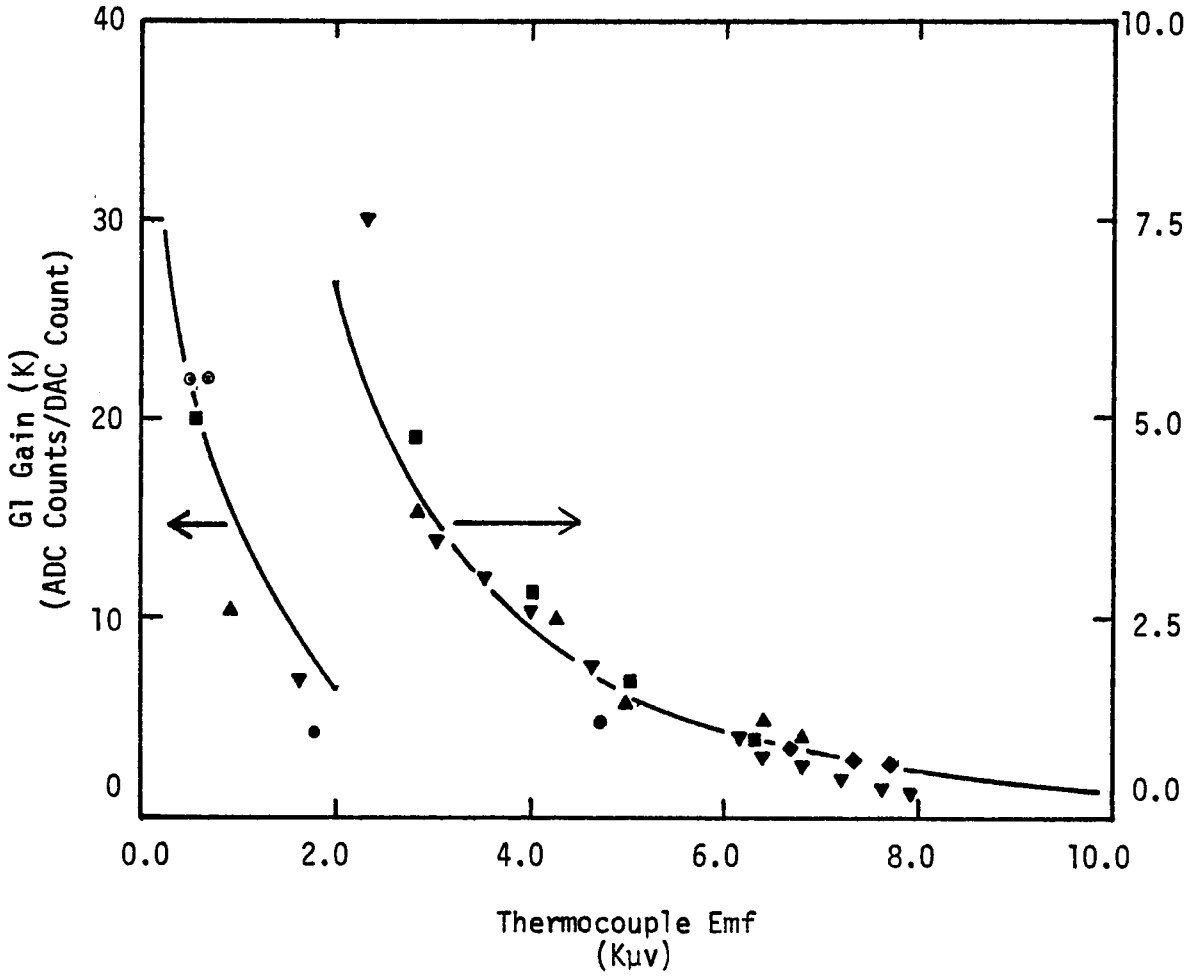


Figure III-17. G1 Gain vs Thermocouple Emf.

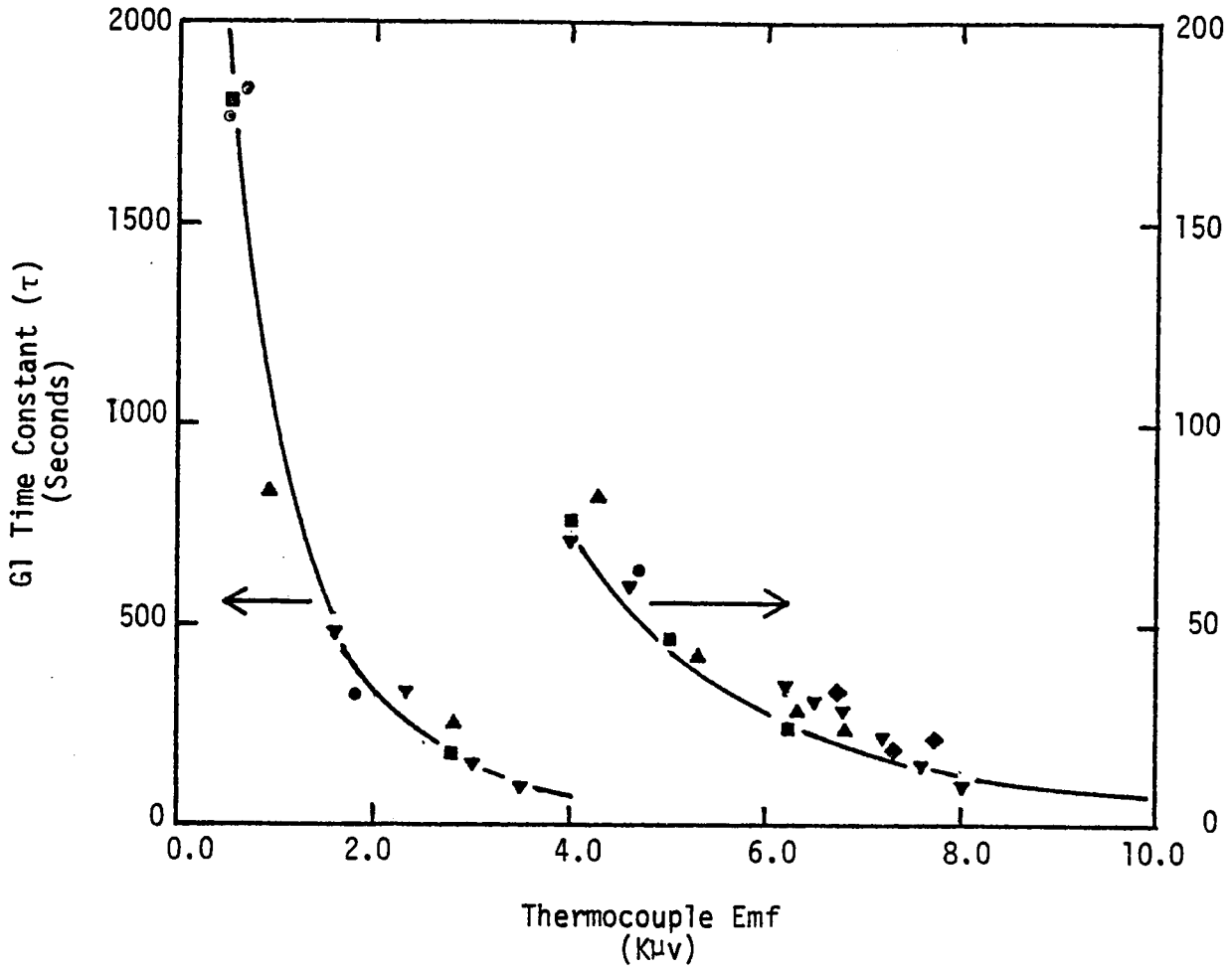


Figure III-18. G1 Time Constant vs Thermocouple Emf.

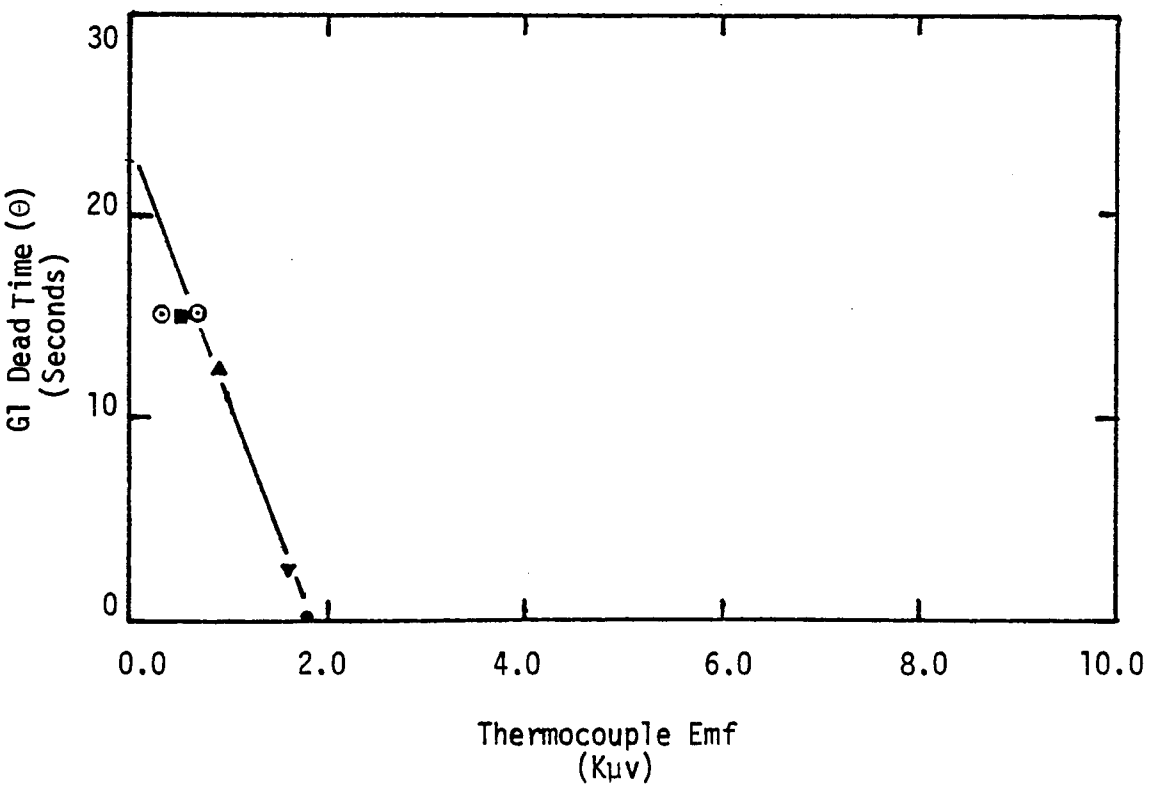


Figure III-19. G1 Dead Time vs Thermocouple Emf.

between temperature and the time constant for G1 could not be found due to the extremely high values at low emfs.) It is easily seen that the process parameters for this shield are much more sensitive to temperature than those of G3. Also, the dead time (Figure III-19) is larger for this shield and remains greater than zero until 2000 μvs.

The second order process parameters (K, Θ, τ_1, τ_2) for the shields Cyl and G2 were also determined by computer curve-fitting, using the z-transform of equation 3-10 in series with a zero-order hold. This produces a recursive digital equation of the form:

$$X_i = A_1 X_{i-1} - A_2 X_{i-2} + K (B_1 U_{i-N-1} + B_2 U_{i-N-2}) \qquad 3\text{-}16$$

where i = sample number

$$N = \Theta/ST$$

$$A_1 = A + B$$

$$A_2 = A * B$$

$$A = e^{-ST/\tau_1}$$

$$B = e^{-ST/\tau_2}$$

$$B_1 = \frac{\tau_2 - \tau_1 + A\tau_1 - B\tau_2}{\tau_2 - \tau_1}$$

$$B_2 = \frac{(\tau_2 - \tau_1)\, AB + B\tau_1 - A\tau_2}{\tau_2 - \tau_1}$$

ST = digital sampling time.

The four process parameters (K, Θ, τ_1, and τ_2) were also optimized by simulating equation 3-16 and using the PATTERN Search technique. After determining τ_1 and τ_2, the parameters ξ and $1/W_n$ were determined using equations 3-11 and 3-12. The data sets in Figures III-8 and III-9 are presented again in Figures III-20 and III-21 along with their computer-

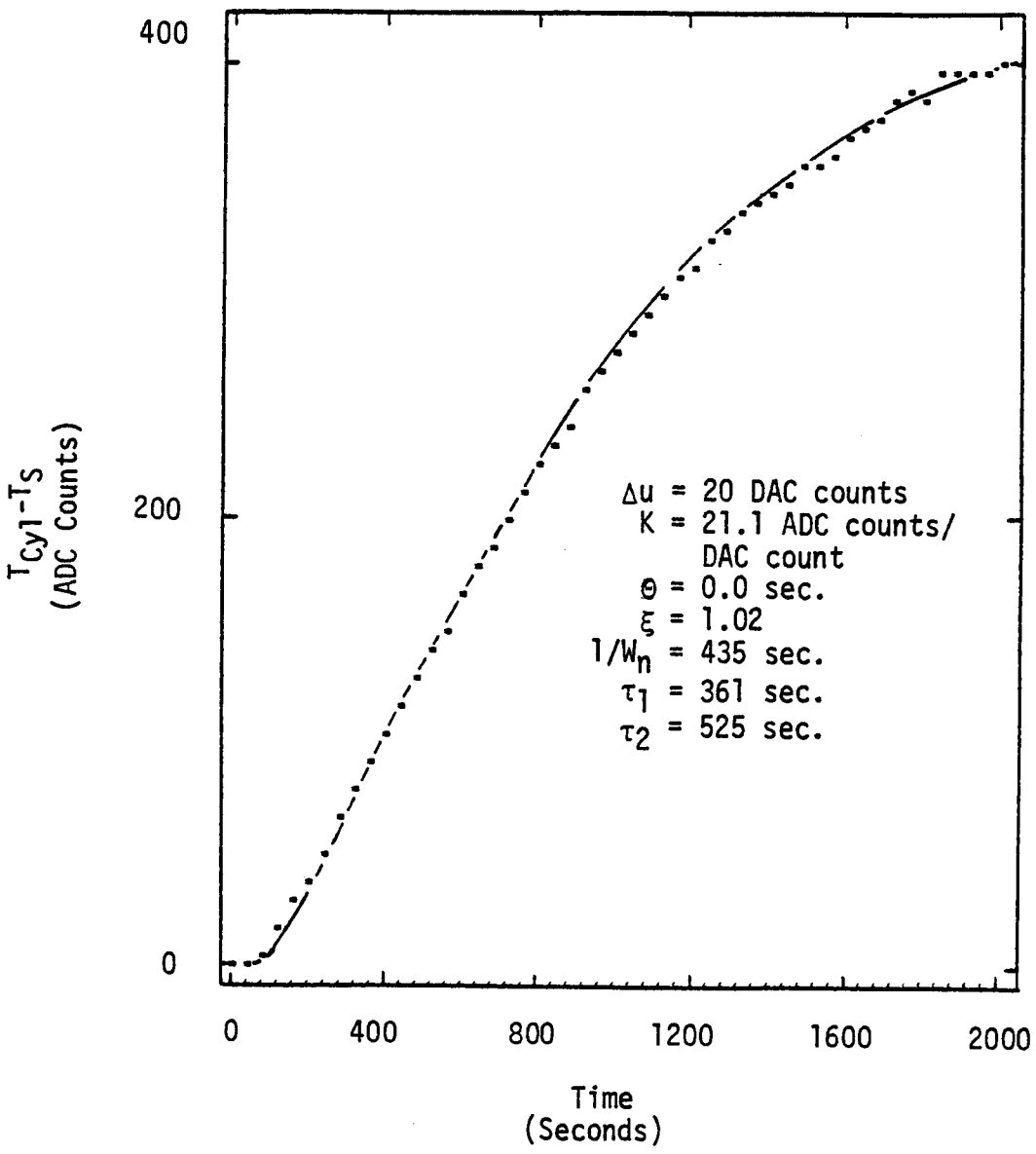


Figure III-20. Step Response for the Shield Cyl from Figure III-8 with the Fitted Curve and Process Parameters.

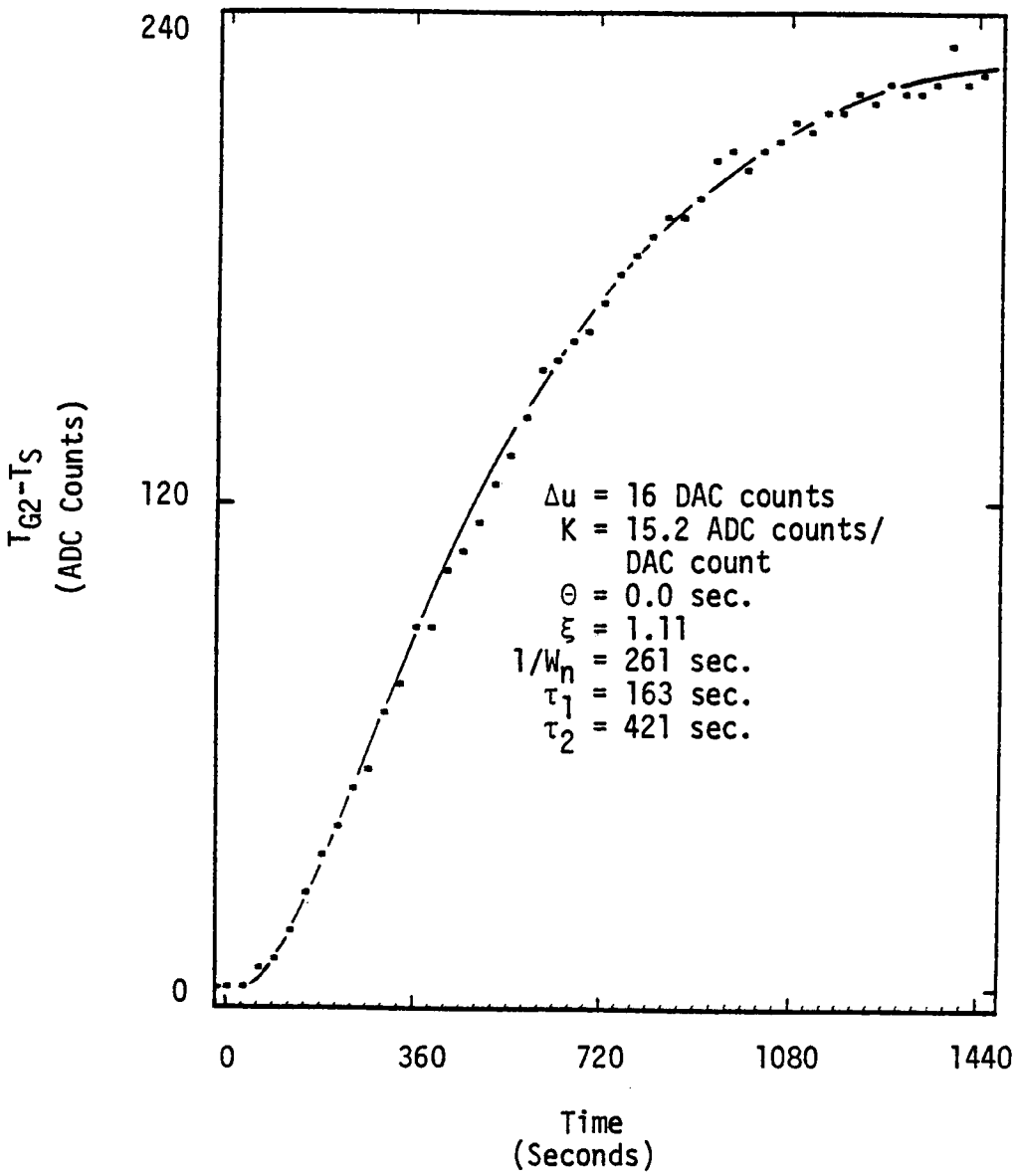


Figure III-21. Step Response for the Shield G2 from Figure III-9 with the Fitted Curve and Process Parameters.

fitted curves and parameters. All data sets were analyzed in this manner for the shields Cyl and G2.

Figures III-22, III-23, and III-24 show the process parameters K, $1/W_n$, and ξ for the Cyl plotted against thermocouple emf. It should be noted that ξ and $1/W_n$ are more sensible to plot than τ_1 and τ_2. The data show that the process gain (K) and frequency ($1/W_n$) do decrease with increasing emf (temperature). The gain drops as predicted by equation 3-4. In analyzing $1/W_n$, it can be easily seen that it is merely the geometric average of τ_1 and τ_2 (equation 3-12) and therefore should have the same temperature dependence (as in equation 3-5) as does a single time constant. Therefore, these two parameters were also curve-fitted using the form of equations 3-14 and 3-15. These values appear in Table II, and the curves are plotted along with the data in Figures III-22 and III-23. The process damping ratio (ξ), plotted in Figure III-24, appears to be almost constant with emf (change is less than 10%). Equation 3-11 predicts this behavior since ξ is the ratio of the arithmetic average of the time constants to the geometric average; therefore, it should be almost constant. But in order to obtain the best possible fit, ξ was fitted to the form:

$$\xi = A_0 + A_1 E \qquad 3\text{-}17$$

where E = thermocouple emf

A_0, A_1 = constants.

These values appear in Table III. All dead times for the Cyl were found to be zero and, therefore, were not plotted. Although the process did not start to respond for several sample times after a

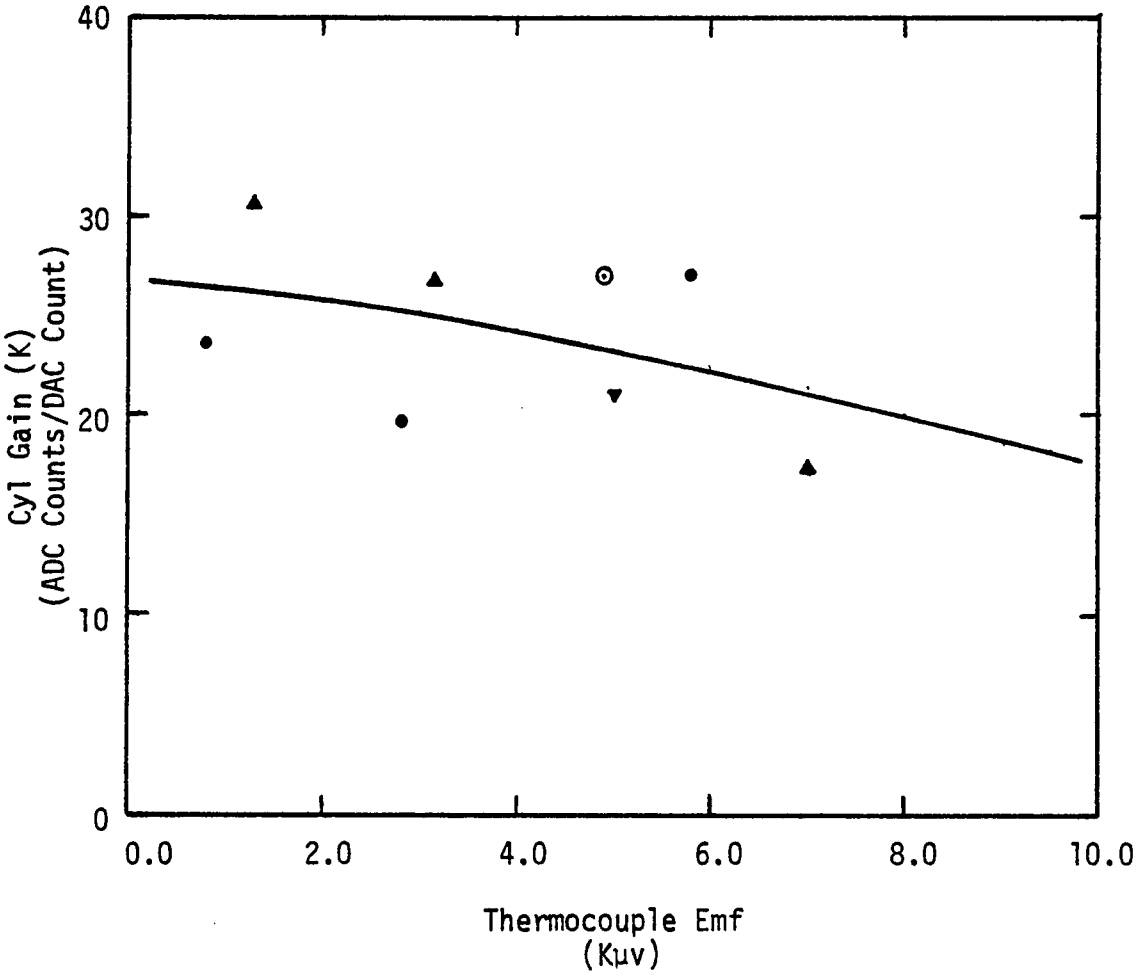


Figure III-22. Cyl Gain vs Thermocouple Emf.

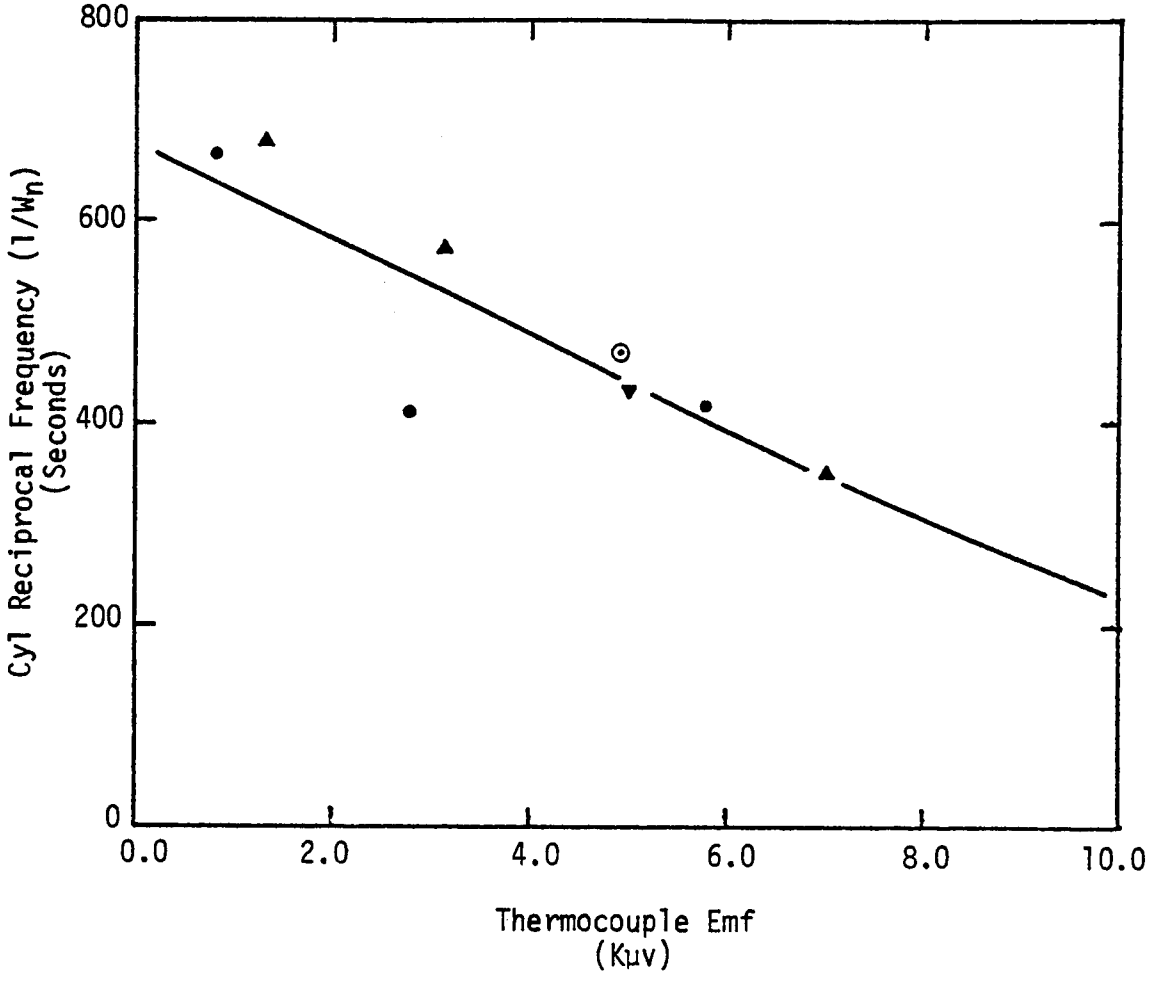


Figure III-23. Cyl Reciprocal Frequency vs Thermocouple Emf.

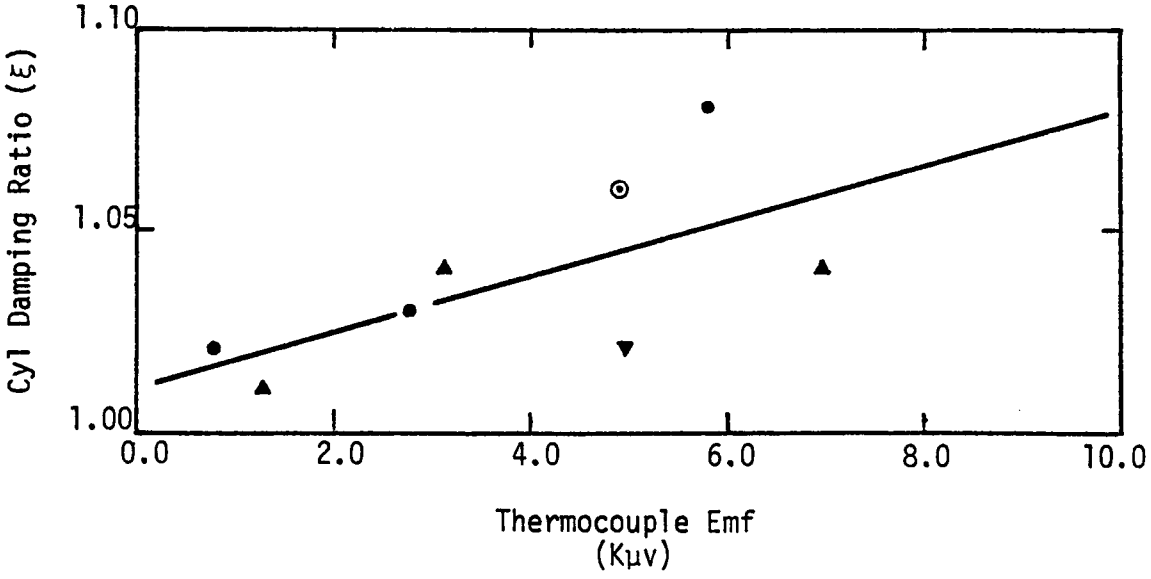


Figure III-24. Cyl Damping Ratio vs Thermocouple Emf.

TABLE III

CONSTANTS IN DAMPING RATIO DEPENDENCY EQUATION
FOR SHIELDS CYL AND G2

Shield	A_o	A_1
Cyl	1.012	6.67×10^{-6}
G2	1.202	0.0

step was issued, this delay was taken into account by the initial part of the "S" shaped curve.

Figures III-25, III-26, and III-27 show the second order parameters for shield G2. Once again the gain (Figure III-25) and frequency (Figure III-26) were curve-fitted to equations 3-14 and 3-15. The resulting values appear in Table II and are plotted along with the data in the appropriate figures. The damping ratio (ξ), which appears in Figure III-27, was fitted to equation 3-17; and its values appear in Table III. The resulting line is also plotted with the data in Figure III-27. This shield also exhibited no dead time; and, therefore, it is unnecessary to plot it. It should be noted here that the shield G2 behaves in much the same way as the Cyl but with lower gains and time constants.

The interaction between the Cyl and G1 consists solely of a gain term as previously stated. In order to determine the value of this gain, the overall interaction must be examined. Figure III-10 displays the data obtained for the shield G1 during a typical step test of the Cyl (Figure III-8). When the Cyl input is stepped, the output begins to increase as governed by its second order dynamics. But as it increases in temperature, it radiates heat to G1; and, therefore, the controller output for G1 drops as more and more free heat is provided by the Cyl. It is very important to note here that the response of G1 does not deviate appreciably from set point conditions. Therefore, the hand-tuned PID controller used in modeling was sufficient to keep G1 in good control. (This fact will later be very important in the control strategy.) Due to the fact that the

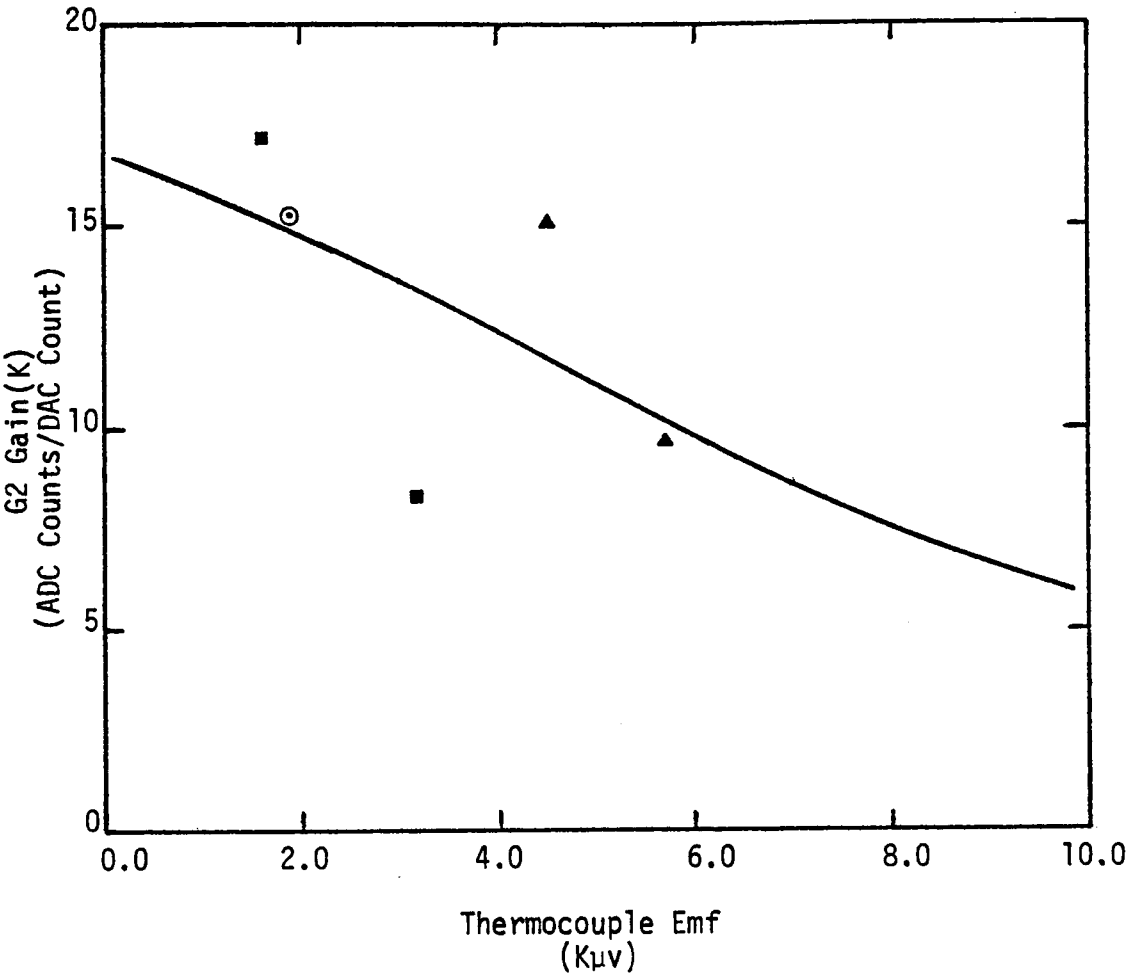


Figure III-25. G2 Gain vs Thermocouple Emf.

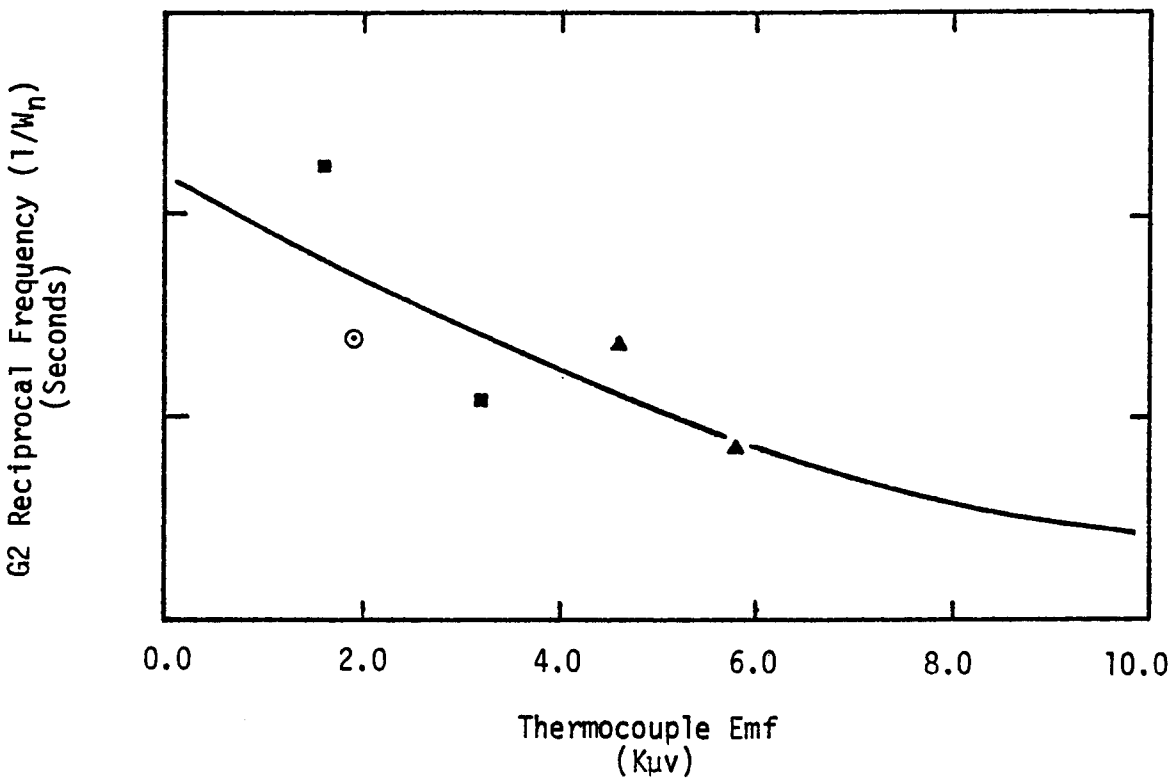


Figure III-26. G2 Reciprocal Frequency vs Thermocouple Emf.

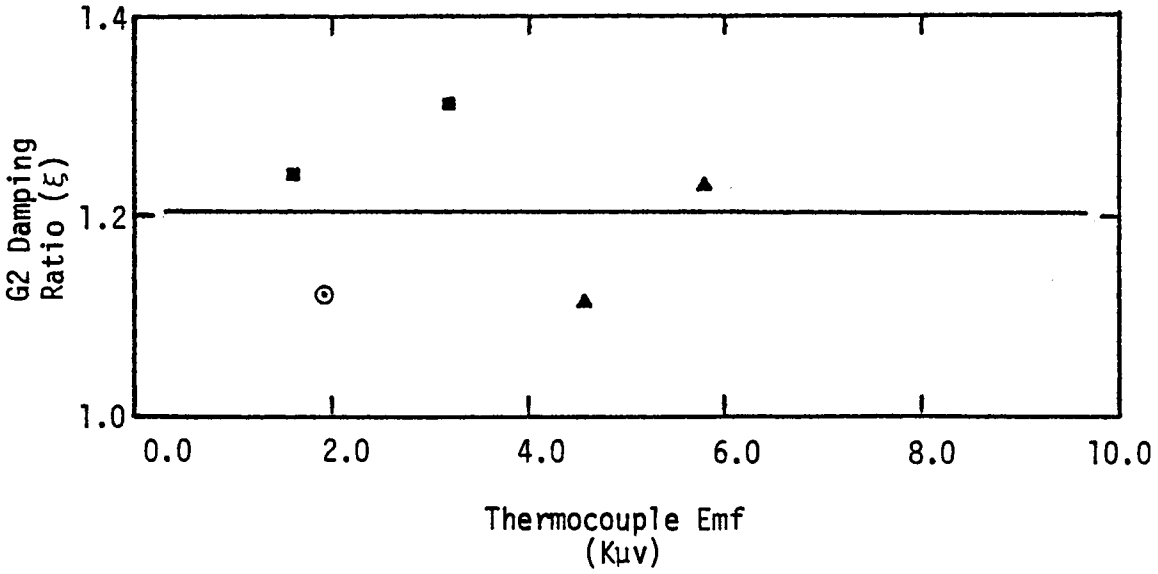


Figure III-27. G3 Damping Ratio vs Thermocouple Emf.

response in G1 does not deviate from steady state conditions, the interaction gain can easily be determined by plotting the controller output of shield G1 vs the temperature response of the Cyl. (This is demonstrated in Appendix C.) Figure III-28 depicts that the facts stated in Appendix C are indeed correct and that the gain is simply the negative of the slope of the aforementioned plot. In this case that gain is 0.1883 DAC counts/ADC count. The other data sets were analyzed in the same manner. The entire set of gains appears in Figure III-29 plotted against thermocouple emf.

Appendix C also demonstrates that the form of the functionality of this gain is:

$$K = AT^3 \qquad 3\text{-}18$$

where T = temperature (°K)

A = constant

which can also be fitted to the thermocouple emf by:

$$K = A'(E + B')^3 \qquad 3\text{-}19$$

where E = thermocouple emf

A', B' = constants.

For the Cyl-G1 interaction, these constants were found to be:

$$A = 3.66 \times 10^{-8}$$

$$A' = 2.31 \times 10^{-12}$$

$$B' = 4500$$

The curve obtained using thermocouple emfs to predict the interaction gain is plotted along with the data in Figure III-29. It predicts the functional relationship for this interaction term very well.

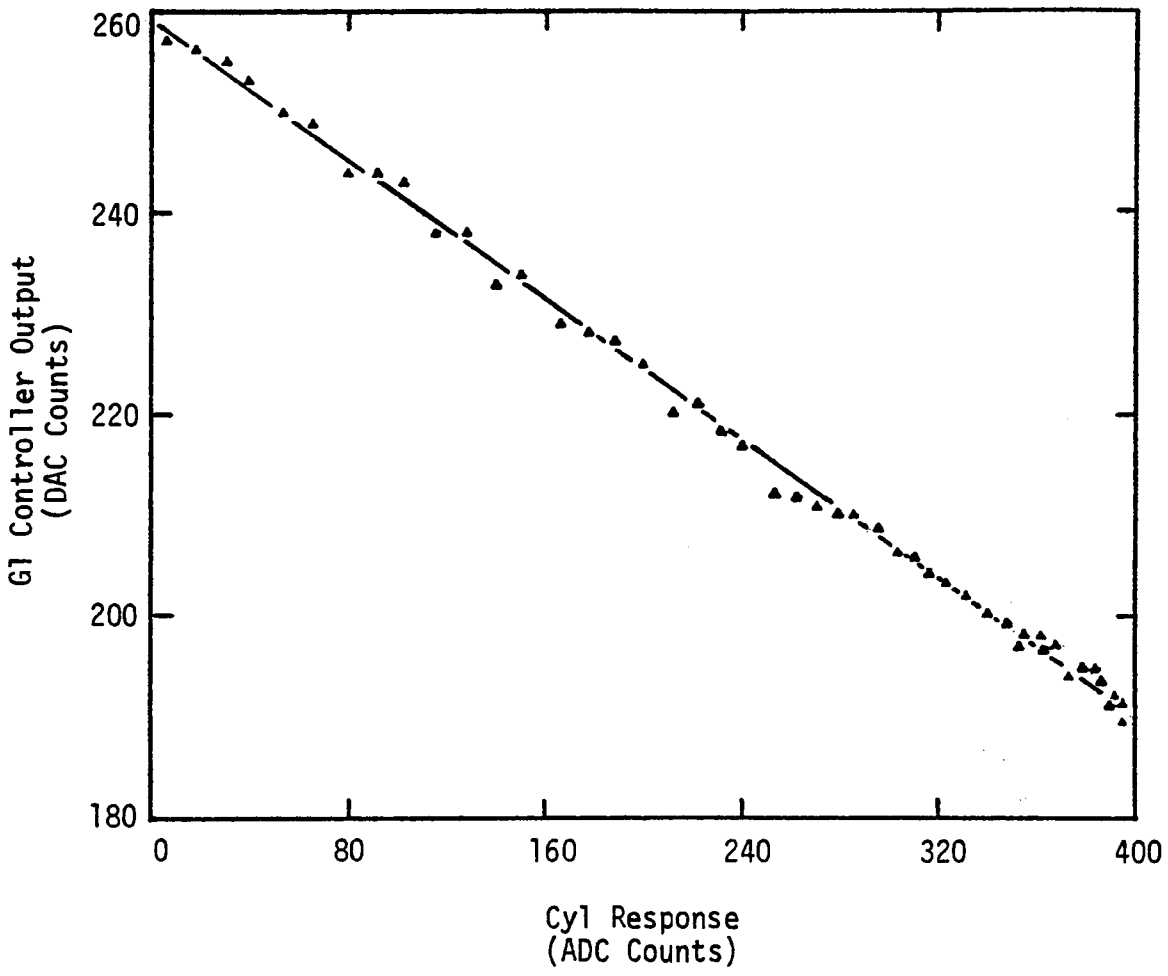


Figure III-28. Typical Plot of G1 Controller Output vs Cyl Response.

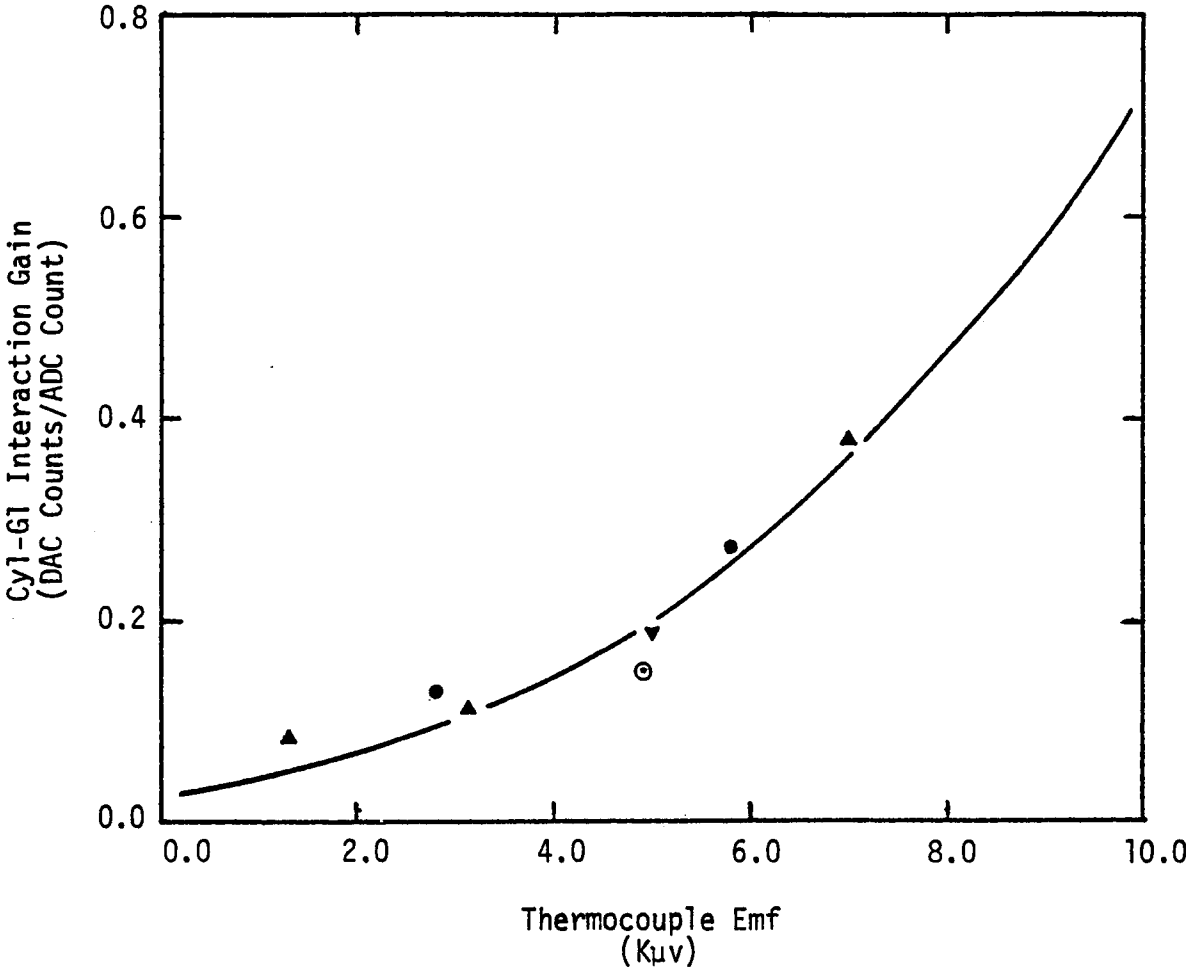


Figure III-29. Cyl-Gl Interaction Gain vs Thermocouple Emf.

Critical Shield Identification

One important factor in the process modeling which has not been examined is that of critical shield identification. In other words, which shields are most important in the control of adiabatic conditions? Once this is determined, great care must be taken to insure very precise control of that shield's temperature deviation from the sample temperature.

The method used to determine the most critical shields arises from equation 3-1 with no power input. By letting the shield i be the sample (denoted by S) and by considering heat loss by radiation to only one shield, the following relationship exists:

$$M_S C_{P_S} \frac{dT_S}{d\Theta} = -K_i (T_S^4 - T_j^4) \qquad 3\text{-}20$$

where j = shield j.

By using the linearized form of this equation (developed in Appendix C), the following relationship between the drift rate of the sample and the deviation of the shield temperature from the sample temperature exists:

$$M_S C_{P_S} D_S = 4T_a^3 K_j (t_j - t_s) \qquad 3\text{-}21$$

where D_S = sample drift rate (°K/hr)

T_a = absolute temperature (°K) at which drift is measured

This equation suggests that by varying the difference between the shield and sample temperatures, the change in sample drift rate can be determined. Figure III-30 displays the result when this procedure was implemented. (The temperature deviation was actually a thermocouple μv deviation and is plotted in that manner.) It dramatically shows that the shield G1 is by far the most important shield in sample drift rates. Therefore, it will be the shield most

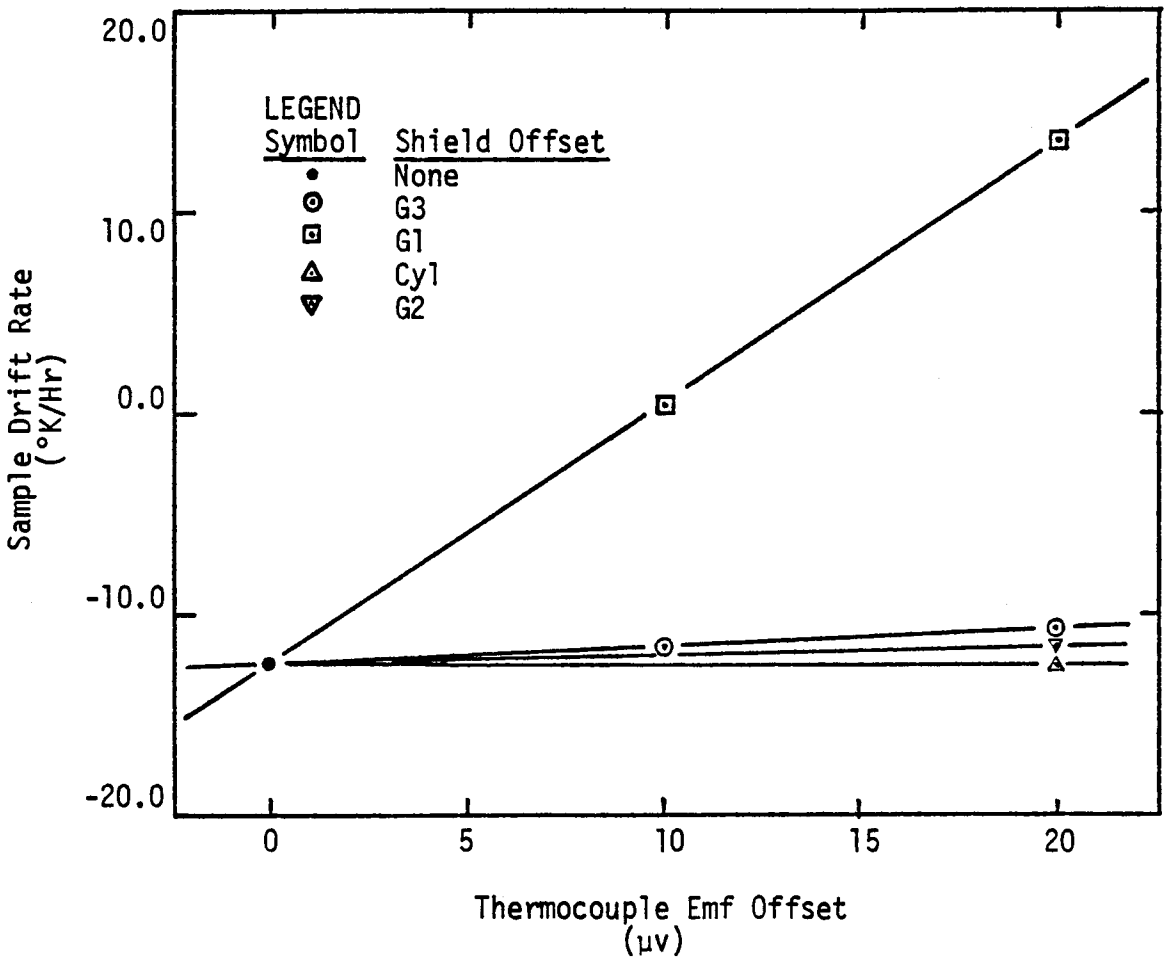


Figure III-30. Sample Drift Rate vs Offset in Thermocouple Emf for Shields G3, G1, Cyl, and G2 at 1020 °K.

critical in control. The shields G3 and G2 have only small effects on the drift rate with the Cyl having almost none. It was noted during testing that the only effect of varying the offset between the Cyl and the sample was to lower the power input to G1. This is exactly as predicted by the interaction between the Cyl and G1 found in the step test modeling.

Also, this figure serves to explain some questions which may have arisen about the data taken in the main model. Many data points were collected for shields G3 and G1, but only a few were collected for the Cyl and G2. This was not only due to long settling times for the second order shields but also due to the fact that extensive modeling in light of their effect on control was not necessary. A general idea of the model was all that was necessary due to the lack of their effect on adiabatic conditions.

This critical shield identification led to quite an extensive analysis of sample drift rates. In fact, it produces a completely new and unique approach to modeling the sample drift rate. This model (developed in detail in Appendix D) allows calculations of drift rates for any sample at any temperature using pseudo-heat transfer coefficients obtained from Figure III-30. It also allows calculation of a zero drift rate condition using the offset in G1.

Summary

The experimental model of the process consists of LaPlace domain transfer functions which relate the process inputs in DAC counts to the process outputs in ADC counts. These transfer functions are in the form of first order lags plus dead time for the shields G3 and G1.

They are in the form of second order lags plus dead time for the shields Cyl and G2. The parameters in these transfer functions were found to be functions of temperature as predicted by the theoretical model. Their values have been fitted against temperature and thermocouple emf using the functional form predicted by the theoretical model. These general forms and parameter values (as functions of temperature) are now suitable for use in the design and tuning of a digital control strategy.

Also, only one interaction was observable in the process. It existed between the Cyl and Gl and was characterized by a gain term. The value of this gain was also fitted to temperature and thermocouple emf.

To complete this model, a critical shield identification was made. It dramatically demonstrated that Gl was the most critical by far in governing adiabatic conditions. Thus, it is the most crucial in control.

CHAPTER IV

CONTROL STRATEGY

Basically, the overall control strategy can be thought of in four distinct phases:

1) Development of the overall control scheme,
2) Selection of the appropriate type of controllers used to implement that scheme,
3) Tuning the parameters in each of the controllers, keeping non-linearities in mind, and
4) Implementation of the overall control package and fine tuning of the controller parameters by hand.

The overall control scheme must be based upon the overall flow diagram of the model, such as in Figure I-4. It must take into account all the variables to be sampled (outputs) and the variables to be controlled (inputs). Appropriate matching of the sampled variables to the controlled variables must be made. It must also take into account any interactions that are of significant strength and seek to break those interactions as well as possible.

The selection of the controller type must be based upon the form of the process model, such as first order lag, second order lag, etc.

Once these general controller types have been selected, they must be tuned (since they all have constants in them) based upon the values of the model parameters. In the case of a non-linear process, such as the one under study, this must be done at several values of

temperature and the dependency of the tuning parameters to temperature must be established. Until this is done, the tuning procedure is not complete.

After completing the above steps, a very necessary step to complete the final DDC package is to implement the overall strategy with its model-determined tuning parameters and to fine tune these parameters by hand. This is a very important and often overlooked step in the procedure since it allows the designer the privilege of correcting for any modeling errors or process characteristics not taken into account by the model of the process.

Control Scheme

In the course of process modeling, it was found that the process only had one significant interaction. Its response (depicted in Figure III-10) demonstrates that even the hand-tuned PID controller used to collect modeling data adequately removed any disturbance in the process response of G1. Due to this fact, there is no justifiable basis to incorporate this interaction into the final control strategy. Therefore, no additional control signals need be added to compensate for this interaction, especially when the fact that large upsets in the Cyl response (such as occurred in modeling) should never happen under actual operating conditions. If they do, it will be caused by errors in controlling the Cyl response, and means should then be taken to solve this problem.

Thus, the final control strategy should consist of four individual and independently operating control loops, one for each of the shields

G3, G1, Cyl, and G2. The loops consist of the scheme shown in Figure IV-1. It should be noted here that the process model includes both the process block and sensor block shown separately in this figure.

Controller Selection

The task now at hand is to select the appropriate controllers for each of the shields.

The first order lag plus dead time process exhibited by shields G3 and G1 can be easily controlled using a PID controller. In fact, after the dead time has dropped to zero, all that is necessary is a PI controller. Here the derivative mode was used to control the dead time process since the dead time was small. If larger dead times had existed relative to the process time constants, more elaborate methods (1,2,24,25) would have been utilized.

The PID algorithm is characterized by the following time domain equation:

$$M(t) = K_c \left[e(t) + \frac{1}{T_i} \int e(t)dt + Td \frac{d(e(t))}{dt}\right] \qquad 4\text{-}1$$

It can be written in the LaPlace domain as:

$$M(S) = K_c \left[1 + \frac{1}{T_i} \cdot \frac{1}{S} + T_d \cdot S\right] e(S) \qquad 4\text{-}2$$

and in the discrete domain as:

$$M_i = K_c \left[e_i + \frac{ST}{Ti} \Sigma e_i + \frac{Td}{ST} (e_i - e_{i-1})\right] \qquad 4\text{-}3$$

where K_c = proportional gain

T_i = integral constant

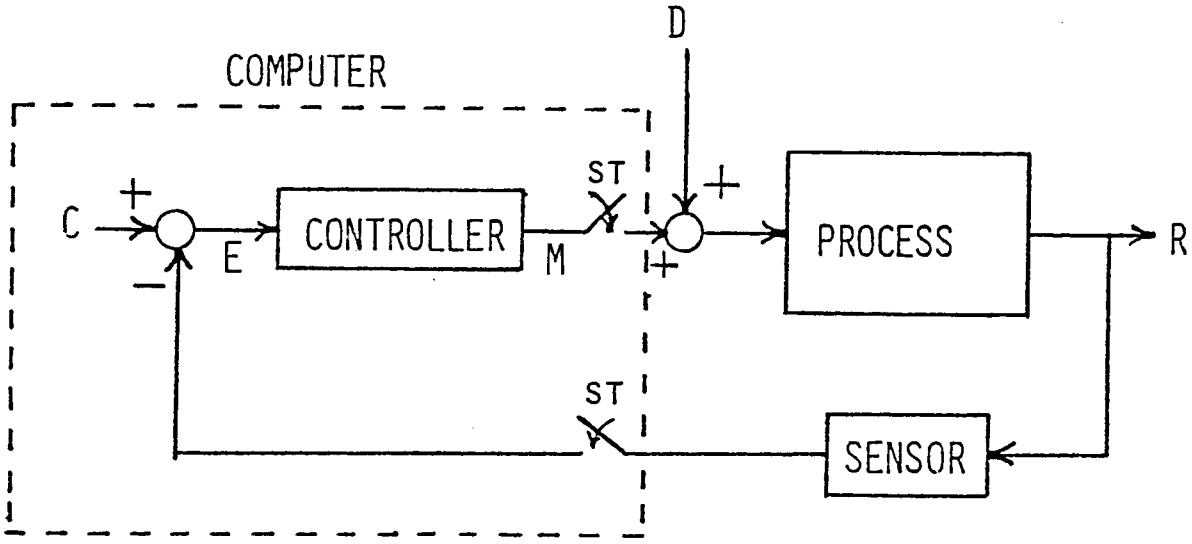


Figure IV-1. Final Control Strategy Consisting of Single Loop Control for Each Shield.

Td = derivative constant

ST = digital sampling time

e = error (set point-process value)

M = controller output.

This algorithm reduces to a PI algorithm by simply letting Td be equal to zero. By subtracting equation 4-3 evaluated at sampling time (i-1) from equation 4-3 itself, the bumpless discrete PID equation becomes:

$$M_i = M_{i-1} + K_c \left[(e_i - e_{i-1}) + \frac{ST}{T_i} e_i + \frac{Td}{ST} (e_i - 2e_{i-1} + e_{i-2})\right] \quad 4\text{-}4$$

which can be grouped on the error values as:

$$M_i = M_{i-1} + K_c \left[C_2 e_i - C_3 e_{i-1} + C_4 e_{i-2}\right] \quad 4\text{-}5$$

where $C_2 = 1 + \frac{ST}{T_i} + \frac{Td}{ST}$

$C_3 = 1 + 2\frac{Td}{ST}$

$C_4 = Td/ST$.

This is the form in which the final algorithm was actually implemented. Tuning is accomplished by optimizing the parameters K_c, T_i, and Td.

The second order lag plus dead time process exhibited by the shields Cyl and G2 was eventually controlled using a PI controller in series with a lead compensator. Normally, a second order lag plus dead time process is controlled by using a PID controller. But in preliminary tuning studies, very large derivative constants were obtained in the tuning procedure. It is known that the derivative mode of a digital controller is very noise sensitive and should be avoided if possible. Due to this fact, the author resorted to compensation as conducted in analog systems especially using Bode plots.

It can be determined, using the standard approach by Eveleigh (34), that a lead compensator is necessary to achieve good control. The lead compensator (sometimes called a lead/lag compensator) is of the form:

$$\frac{M(S)}{e(S)} = K_1 \frac{1 + \tau_1 S}{1 + \tau_2 S} \qquad 4\text{-}6$$

where K_1 = gain

τ_1, τ_2 = constants determined from compensation ($\tau_1 > \tau_2$).

For comparison purposes, Figure IV-2 shows the response to a set point change for a PI, PID, and lead compensator. The lead compensator was developed on the Bode plot to have a span of 10 and a gain of 22. This yields a crossover frequency of 0.013 which converts to a rise time (T_r) of 80 seconds. This compares to a simulated time of 90 seconds since a straight line Bode was used. The phase margin was 74 degrees which causes the small overshoot of 12%. (These values for the process are comparable to those encountered for the shield Cyl.) The PI and PID algorithms were tuned using the PATTERN Search technique with a controller velocity (as will be described later) the same as the lead compensator. It is easily recognized that the PI controller produces the poorest response, with the PID next, and the lead compensator the best by far. But Figure IV-3 displays the load response of the lead compensator used in Figure IV-2. This is unacceptable, but it can easily be remedied by replacing the gain term K_1 (actually a proportional controller) by a PI controller. Therefore, the overall controller becomes a PI controller in series with a lead compensator for which the LaPlace equation is:

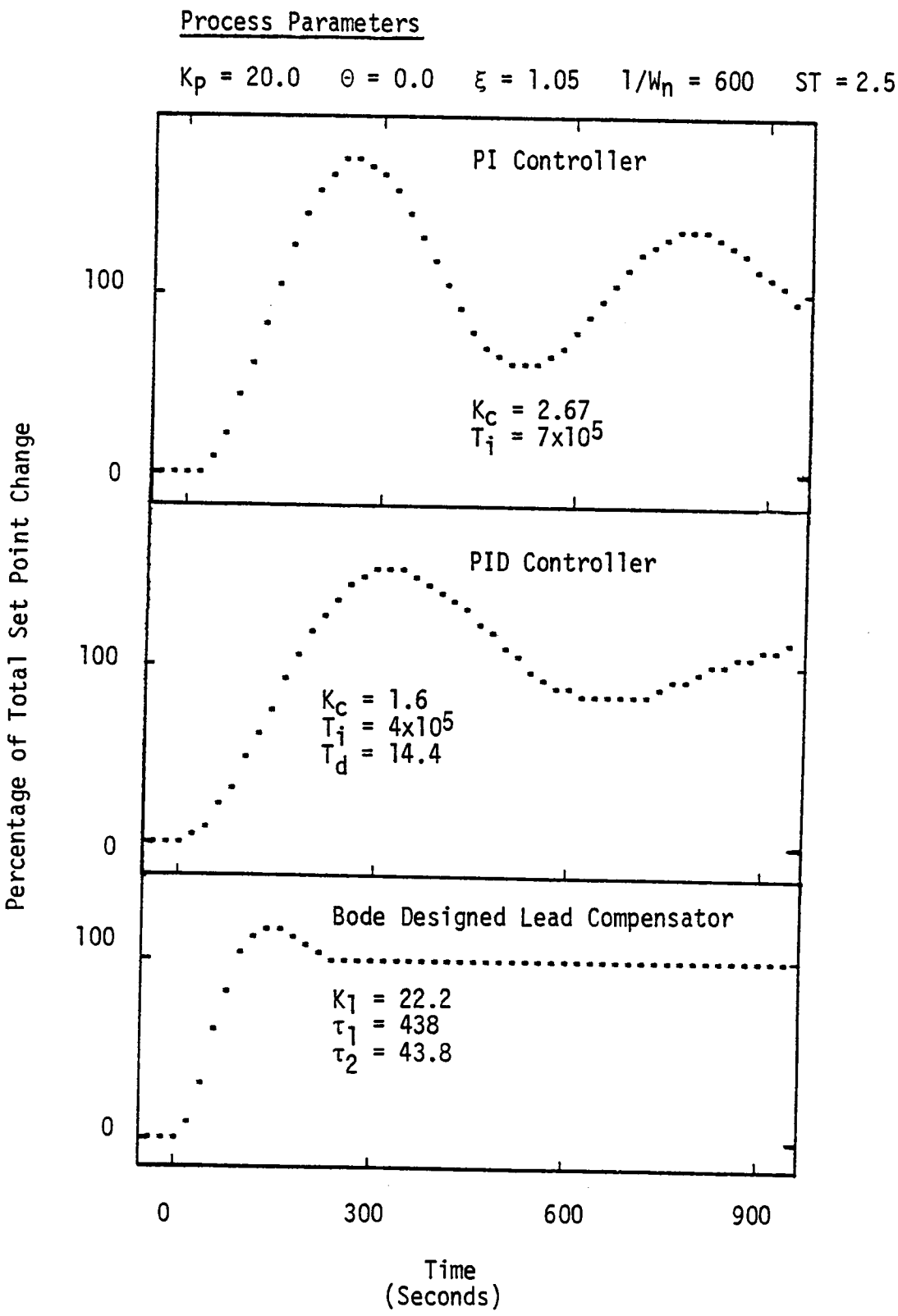


Figure IV-2. Comparison of Controller Types for a Set Point Change to a Second Order Lag Process.

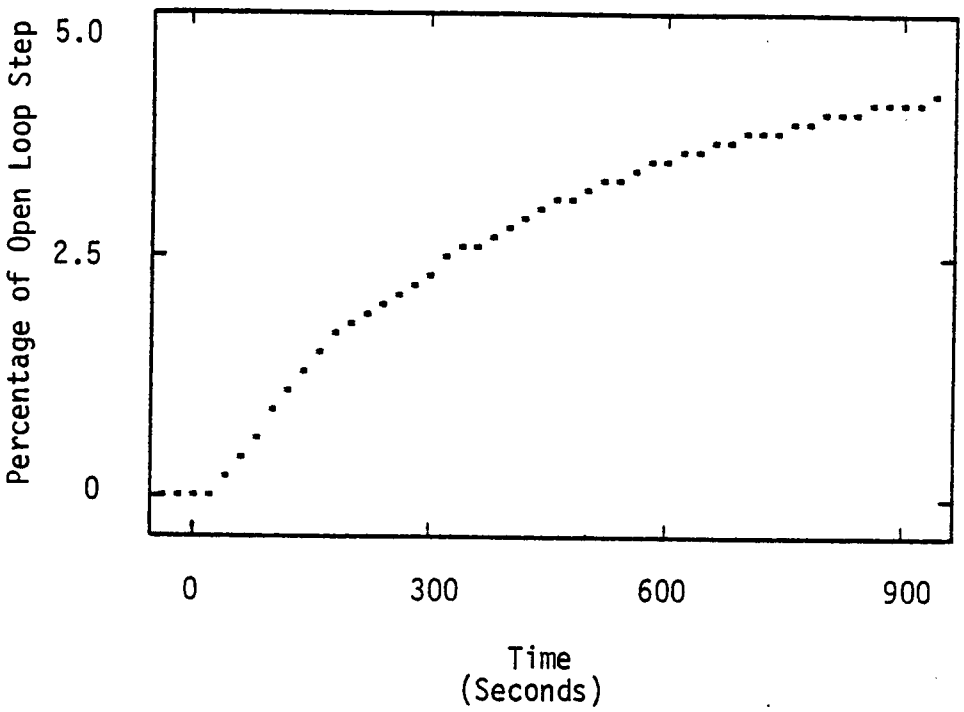


Figure IV-3. Load Response for the Bode Designed Lead Compensator in Figure IV-2.

$$\frac{M(S)}{e(S)} = K_c \left[1 + \frac{1}{T_i} \cdot \frac{1}{S}\right] \left[\frac{1 + \tau_1 S}{1 + \tau_2 S}\right] \qquad 4\text{-}7$$

This controller is tuned by optimizing the parameters K_c, T_i, τ_1, and the ratio τ_2 / τ_1.

The digital form of this algorithm was developed from a totally different approach to the problem. Since the final controller implemented used this approach, it must be developed here. As for now, let it be said that the above equation (4-7) can be made discrete by using two methods. They are formed by using the discrete version of the PI algorithm in series with the discrete lead compensator formed by:

1) Replacing xs by dx/dt and approximating this derivative as $\Delta x/\Delta t$, or
2) Taking the z-transform of a zero order hold in series with the lead compensator

These two methods are used to develop the final control algorithm in Appendix E. It should be noted here that these two methods and the approach about to be discussed all lead to the same result.

A second design was also pursued based upon a cascaded type controller. The ideal solution would be to put a proportional cascade controller around one part of the second order lag as shown in Figure IV-4. This would be put around the lower of the two time constants since the purpose of the entire exercise is to speed up the response of the inner (slave) loop. The master controller, a PI algorithm, can be used in the outside loop to complete the control and eliminate steady state offset. This method is impossible since a sensor cannot be located physically at this location; but, by knowing

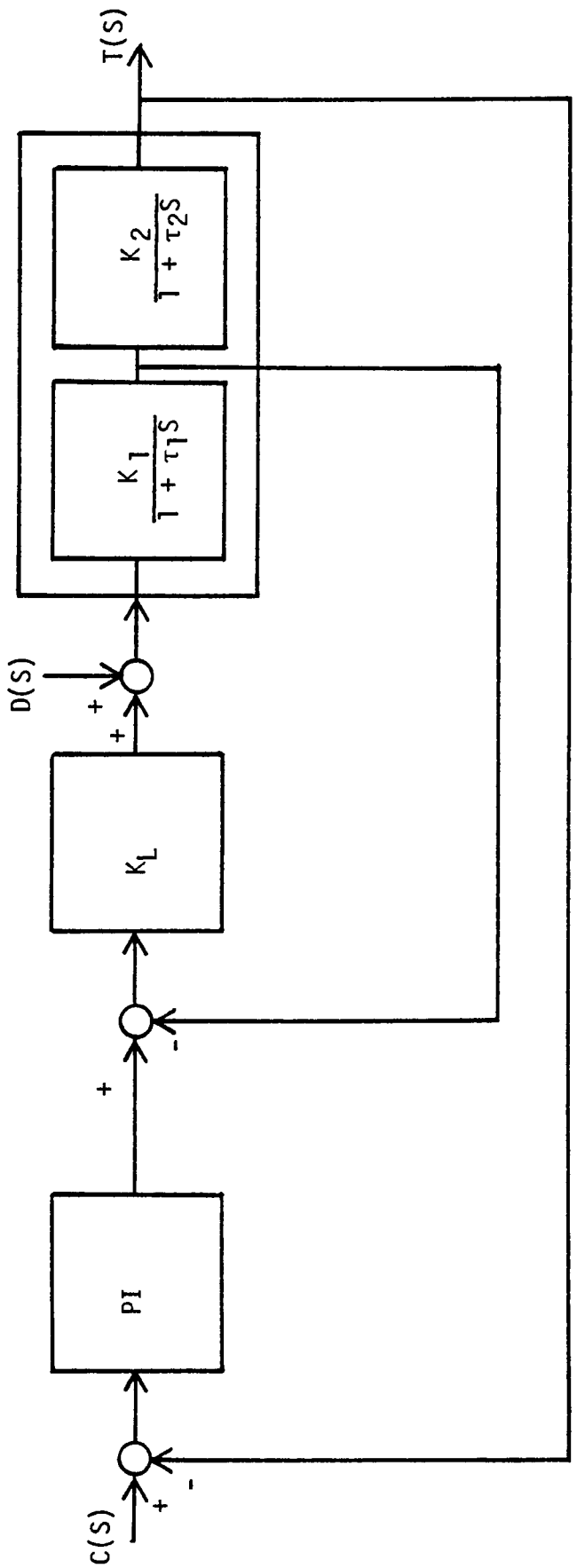


Figure IV-4. Cascade Approach to the Control of a Second Order Lag Process.

the model of the process, a feedback loop can be formed around the model of the first order part of the second order process. (Once again, the lower time constant should be used here.) This control loop is shown in Figure IV-5. But why not collapse this feedback loop into a single block? When this is done, the result is a lead compensator of the form:

$$\frac{K_L}{1 + K_L K_P} \cdot \frac{1 + \tau_P S}{1 + (\tau_P/(1 + K_L K_P))S} \qquad 4\text{-}8$$

where K_L = proportional gain in slave controller

K_P = process gain

τ_P = lower of 2 process time constants.

Comparing this to equation 4-6, we find:

$$K_1 = \frac{K_L}{1 + K_L K_P} \qquad 4\text{-}9$$

$$\tau_1 = \tau_P \qquad 4\text{-}10$$

$$\tau_2 = \frac{\tau_P}{1 + K_L K_P} \qquad 4\text{-}11$$

and the result is the same qualitatively as developed on the Bode. But why should this compensation be done in the LaPlace domain and transformed to the z-domain? By treating the model as the z-domain version of the first order lag in the feedback loop in the slave part of the cascade-type system (as in Figure IV-6), the loop can be collapsed to the following:

$$K_L \frac{1 - \alpha Z^{-1}}{1 - (\alpha - K_L K_P (1 - \alpha)) Z^{-1}} \qquad 4\text{-}12$$

where $\alpha = e^{-ST/\tau_P}$

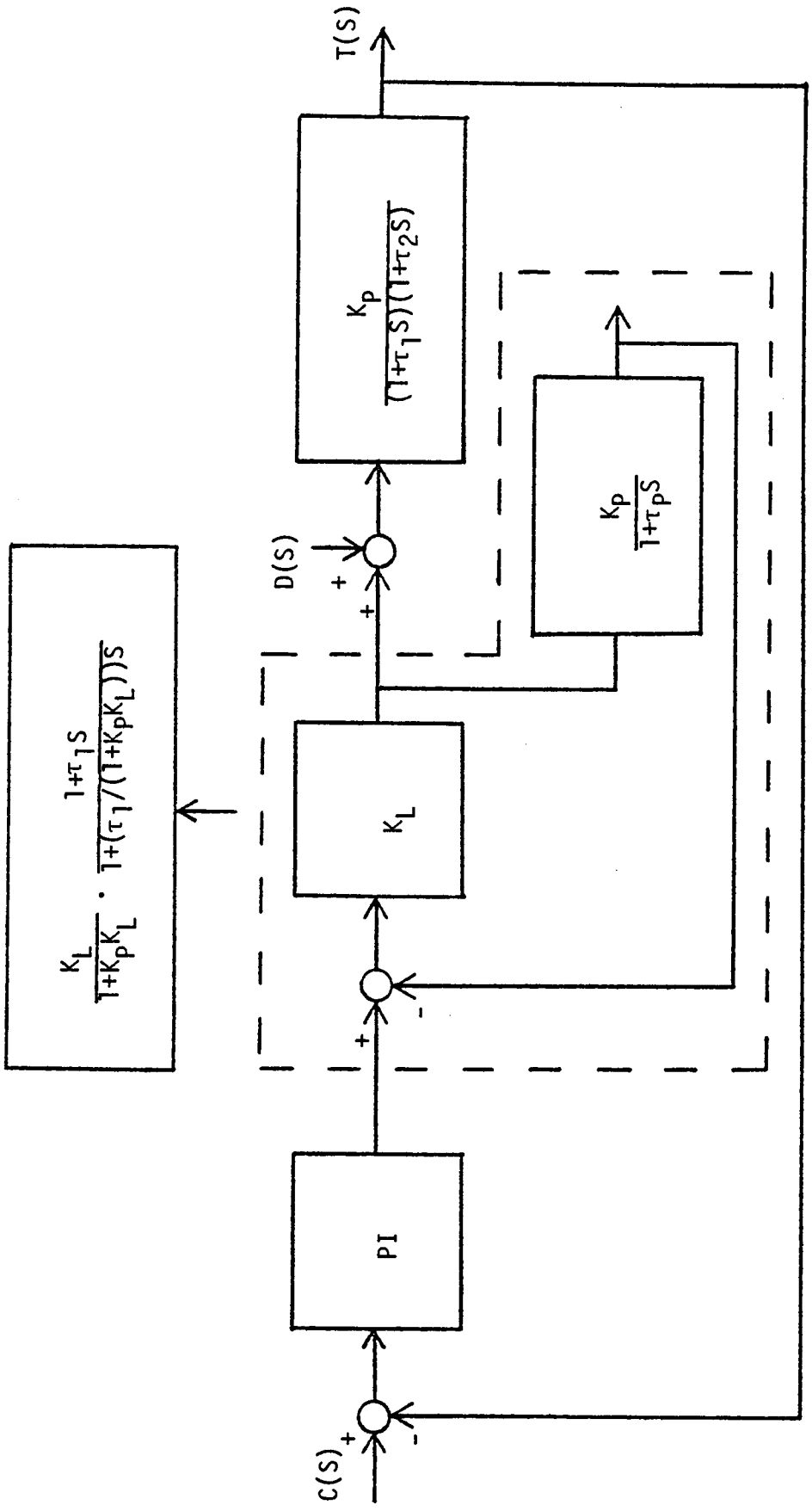


Figure IV-5. Cascade Approach to the Control of a Second Order Lag Process Using Feedback Around Part of the Process Model.

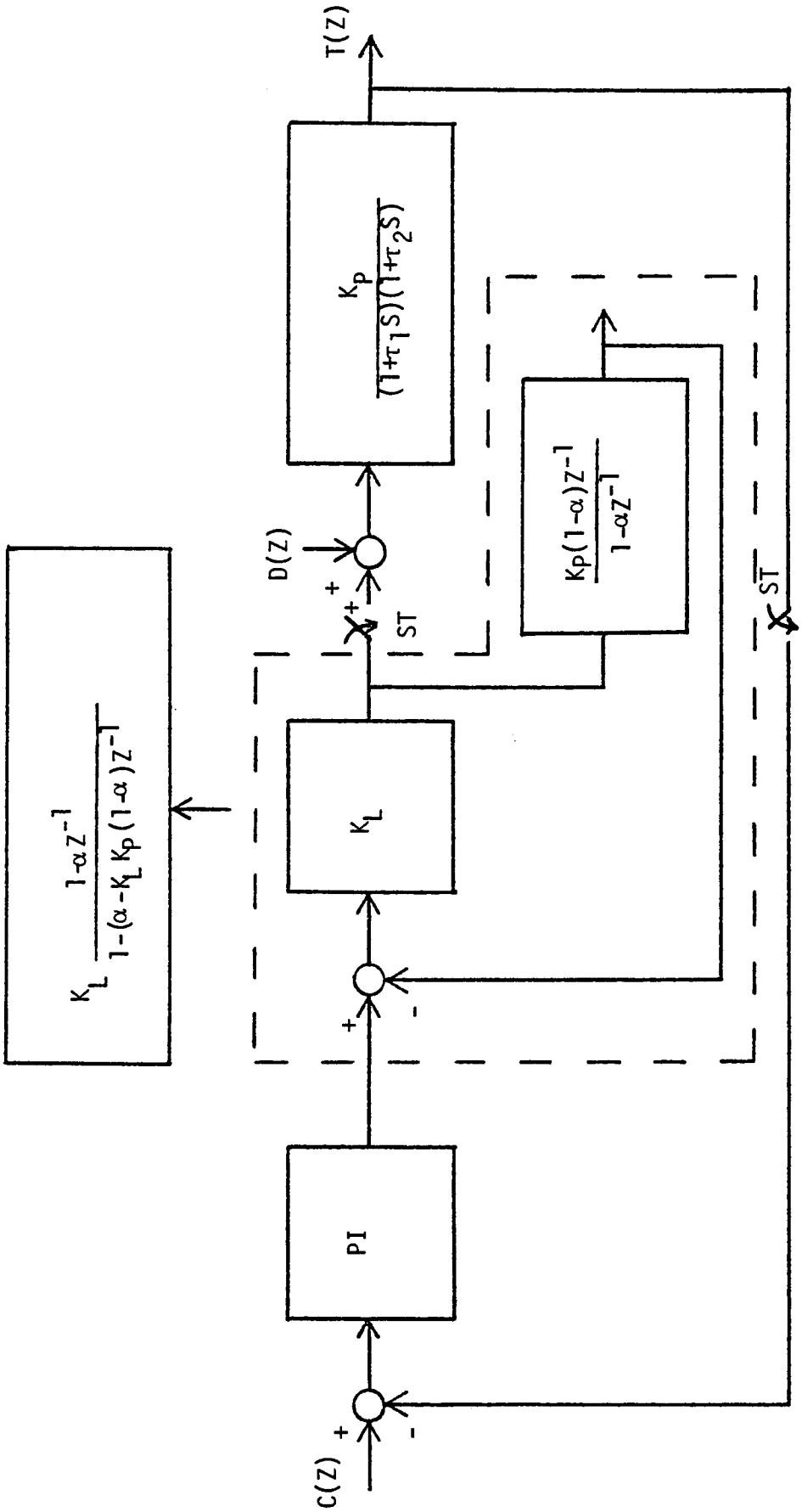


Figure IV-6. Cascade Approach to the Control of a Second Order Lag Process Using Feedback Around Part of the Process Model Using Z-Transforms.

which is a z-domain lead compensator (one zero and one pole).

Having addressed the lead compensation, the PI controller must be added. From equation 4-4, with the derivative constant set to zero, the following z-domain equation arises:

$$\frac{M(Z)}{e(Z)} = K_c \frac{(1 + \frac{ST}{T_i}) - Z^{-1}}{1-Z^{-1}} = K_c \left(1 + \frac{ST}{T_i} \frac{1}{1-Z^{-1}}\right) \qquad 4\text{-}13$$

This can be rewritten as:

$$\frac{M(Z)}{e(Z)} = K_c \left(1 + \frac{ST}{T_i}\right) \left(\frac{1 - \beta Z^{-1}}{1 - Z^{-1}}\right) \qquad 4\text{-}14$$

$$\text{where } \beta = \frac{1}{1 + \frac{ST}{T_i}} \qquad 4\text{-}15$$

By letting

$$\gamma = \alpha - K_L K_p (1 - \alpha) \qquad 4\text{-}16$$

in equation 4-12 and by multiplying the result by equation 4-14, the PI plus lead compensator z-domain equation becomes:

$$\frac{M(Z)}{e(Z)} = K_c \left(1 + \frac{ST}{T_i}\right) K_L \frac{1 - (\alpha + \beta) Z^{-1} + \alpha\beta Z^{-2}}{1 - (\gamma + 1) Z^{-1} + \gamma Z^{-2}} \qquad 4\text{-}17$$

The final algorithm can then be arranged into the form:

$$M_i = C_1 M_{i-1} - (C_1 - 1) M_{i-2} + K_c (C_2 e_i - C_3 e_{i-1} + C_4 e_{i-2}) \qquad 4\text{-}18$$

$$\text{where } C_1 = \gamma + 1 \qquad 4\text{-}19$$

$$C_2 = K_L \left(1 + \frac{ST}{T_i}\right) = K_L/\beta \qquad 4\text{-}20$$

$$C_3 = C_2 (\alpha + \beta) \qquad 4\text{-}21$$

$$C_4 = C_2 \alpha\beta \qquad 4\text{-}22$$

This controller (PI plus lead compensation) has now been developed from two separate views, classical and practical. Both achieve the same

endpoint. The algorithm now contains the important aspects of lead compensation for maximum effectiveness on a second order response and an integral mode for offset elimination and improved load response. It can be tailored to the process by determining the parameters K_C, T_i, and K_L; therefore, it is still a three parameter controller just as the PID controller is.

Controller Tuning

Controller tuning is accomplished by simulating the digital controller and process model as depicted in Figure IV-7. The controller tuning parameters are then optimized by minimizing the error entering the controller using some standard error criterion, such as IAE, ISE, ITAE (18,20). This should be done for both a set point change (point C) and a disturbance change (point D). For each shield, this must be done using the values of the process model at several different temperatures. The tuning parameters obtained at these temperatures can then be used to determine the temperature (actually thermocouple emf) dependency of the tuning parameters for each shield.

The first order lag plus dead time models were simulated using the selected controller (PID) described previously. This was done by keeping in mind the initial steady state value of the controller output obtained when modeling data were collected and the minimum and maximum controller output in DAC counts. It should be noted here that these constraints had no effect on the tuning of the first order shields. When this simulation was carried out, responses obtained for an optimized set point change and an optimized load change are shown in Figure IV-8 for a typical set of model parameters. These responses are known as

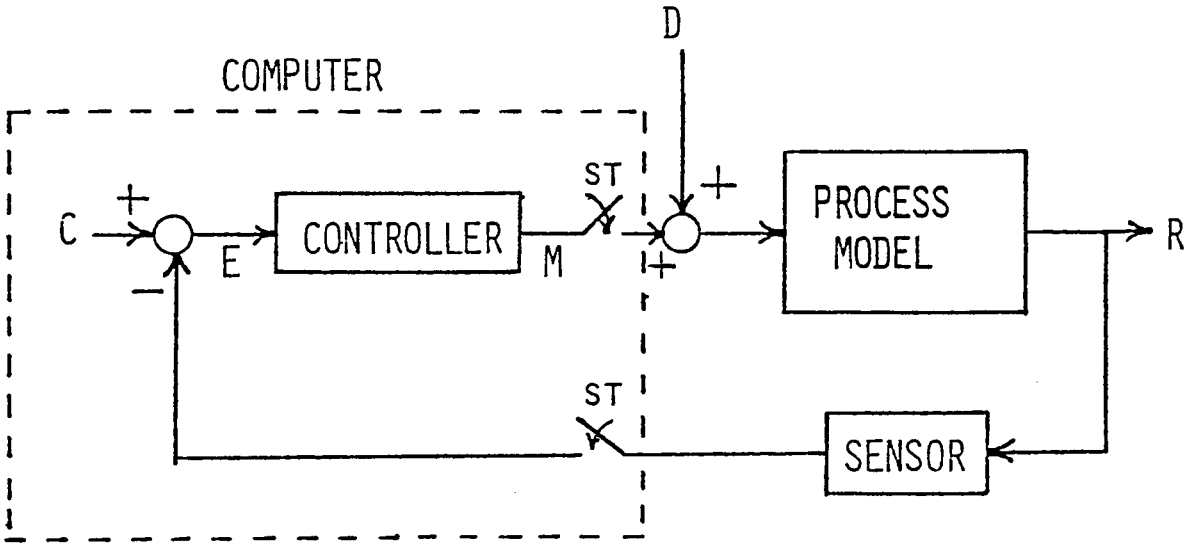


Figure IV-7. Simulation Loop for Controller Tuning.

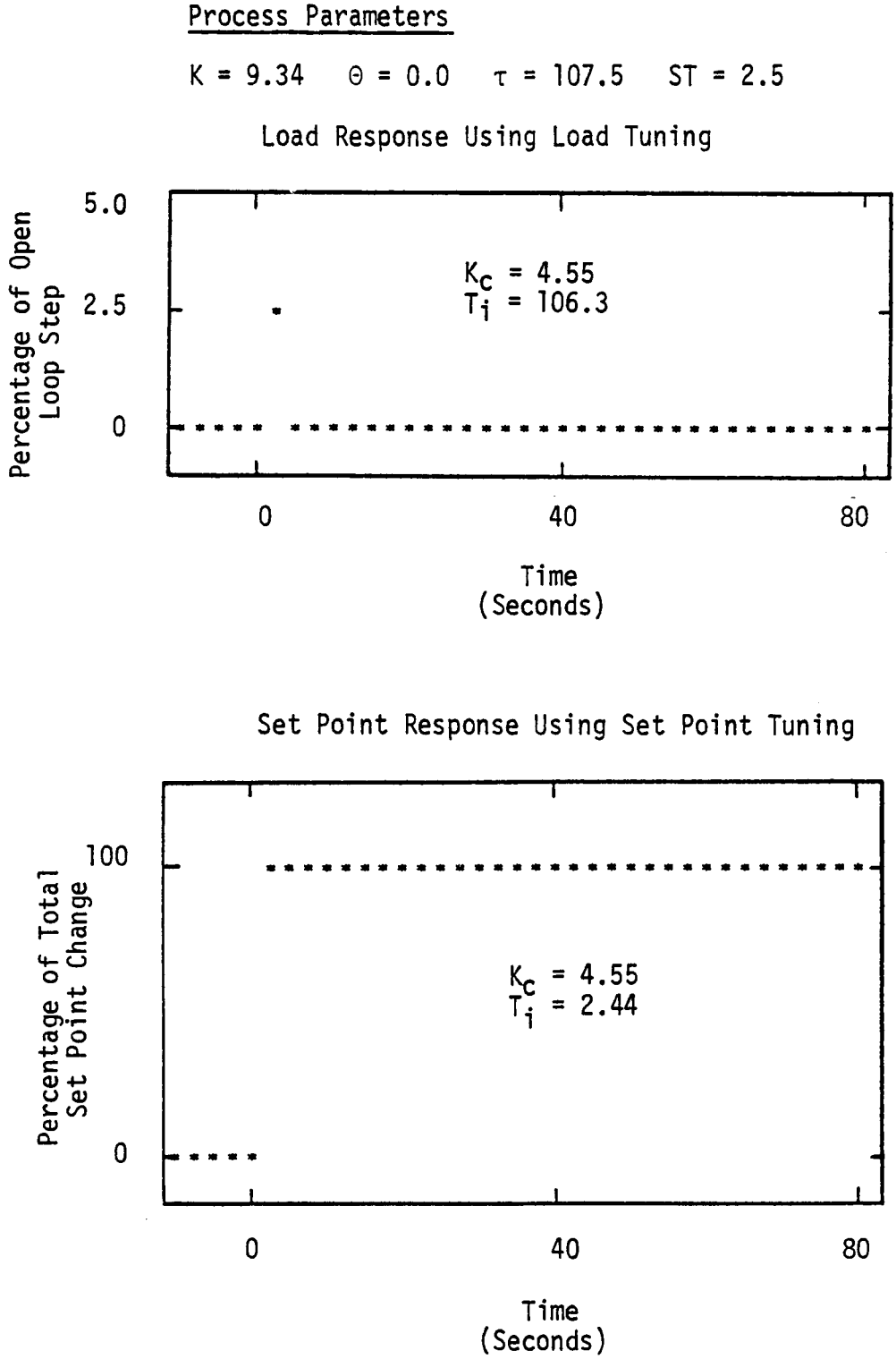


Figure IV-8. First Order Lag Deadbeat Response Using Unrestrained Tuning.

deadbeat responses since they respond in one sample time for the zero dead time case. It can be shown (Appendix F) that the tuning parameters for this type response can be directly calculated without using simulation for both a set point change and a load change. The deadbeat parameters for the shields G3 and G1 are listed in Tables IV and V along with the model parameters (K, Θ, τ). (Deadbeat parameters for the dead time cases were not listed. This is not to say that they do not exist for the dead time case. They can be calculated just as those without dead time, but the controller used to implement these parameters is not a standard PI controller. It involves substituting M_{i-N-1} for M_{i-1} in equation 4-5 where N is the integer dead time.)

When these deadbeat parameters were implemented, the control of the actual process was terrible. Both shields became very erratic, as was expected. This apparent instability was due to the high gains obtained from the deadbeat tuning. The tuning is done in a noise-free environment, but the process operates in a discrete environment. Even under ideal control of $\pm$ 1 ADC count, the controller output will be responding to this as a deviation from the set point of zero. The gains obtained using the deadbeat tuning result in the maximum values obtainable; therefore, they are the most sensitive to noise. Several approaches were considered, such as averaging the error over the last N sampling times, filtering the error using a first order filter, etc. But all these methods add additional dynamics to the process and slow the overall response of the method plus process. In the case of the first order filter, it is easily seen that the overall process is now second order and the controller tuning resulted in the addition of

TABLE IV

PROCESS PARAMETERS AND DEADBEAT PARAMETERS FOR SHIELD G3

Emf (μv)	Process Parameters			Set Point Tuned		Load Tuned	
	K	Θ	τ	K_C	T_i	K_C	T_i
500	11.84	2.5	123.7	-	-	-	-
1500	9.34	0.0	107.5	4.55	106.3	4.55	2.44
2500	7.29	0.0	91.6	4.96	90.4	4.96	2.43
3500	5.67	0.0	77.0	5.34	75.8	5.34	2.42
4500	4.44	0.0	64.2	5.67	62.9	5.67	2.40
5500	3.51	0.0	53.4	5.94	52.2	5.94	2.38
6500	2.80	0.0	44.4	6.17	43.2	6.17	2.36
7500	2.26	0.0	37.0	6.33	35.4	6.33	2.34
8500	1.84	0.0	31.0	6.47	29.8	6.47	2.31
9500	1.52	0.0	26.2	6.57	25.0	6.57	2.27

TABLE V

PROCESS PARAMETERS AND DEADBEAT PARAMETERS FOR SHIELD G1

Emf (μv)	Process Parameters			Set Point Tuned		Load Tuned	
	K	Θ	τ	K_C	T_i	K_C	T_i
500	21.90	15.0	2100.0	-	-	-	-
1500	9.44	5.0	523.0	-	-	-	-
2500	4.91	0.0	203.0	16.50	202.0	16.50	2.47
3500	2.87	0.0	100.0	13.80	99.0	13.80	2.44
4500	1.82	0.0	55.6	11.90	54.4	11.90	2.39
5500	1.22	0.0	34.3	10.80	33.1	10.80	2.32
6500	0.86	0.0	22.6	9.94	21.4	9.94	2.24
7500	0.63	0.0	15.6	9.13	14.4	9.13	2.13
8500	0.48	0.0	11.3	8.41	10.1	8.41	2.00
9500	0.37	0.0	8.4	7.80	7.2	7.80	1.86

derivative mood to compensate. Since the derivative mode is highly noise sensitive (as will become apparent later), this resulted in a "chasing of the tail" type approach. But why should all attention be focused on massaging the process signal? It was obvious that the deadbeat controller parameters are too high; therefore, why not develop a method to logically throttle the controller in order to minimize the noise sensitivity?

It should be strongly noted here that this problem is not due to modeling errors or any process characteristics. It is due solely to the discrete nature of the ADC and DAC signals. The controller tuning parameters are completely independent of this consideration. One possible solution to aid in smoothing the discrete nature of the ADC input (which is the source of the problem) involves running the data input at a much faster sampling rate than the controller. By using averaging or a first order filter with a settling time less than the controller sampling time, the ADC input is no longer limited to an integer change. This was impossible here since the minimum time (due to the nature of the multiplexer) at which the system can be sampled is the rate at which the controllers are run, namely 2.5 seconds.

The method chosen to alleviate this problem was to throttle the controller algorithm in the following manner. Using an error range of $\pm$ 1 counts centered about zero (i.e., +1, -1, +1, -1), the swing in controller action was determined by using equation 4-5 for the PID algorithm. If any point in the error sequence above is chosen, the following result is obtained:

$$M_i = M_{i-1} + K_c [C_2 (+1) - C_3 (-1) + C_4 (+1)] \qquad 4\text{-}23$$

so

$$\Delta M_i = K_c\ (C_2 + C_3 + C_4) \qquad 4\text{-}24$$

At the next sample time, it can be seen that

$$\Delta M_{i+1} = K_c\ (-C_2\ -C_3\ -C_4) = -\ \Delta M_i \qquad 4\text{-}25$$

Just as the error is centered within $\pm$ 1 count, about zero, the controller output will be centered within $\pm\ \Delta M_i/2$ about the steady state controller output. Since it is desirable to control the controller swing within a reasonable and specified limit (M_v), the controller tuning parameters should be subjected to the constraint (using equation 4-24):

$$K_c\ (C_2 + C_3 + C_4)/2 \leq M_v \qquad 4\text{-}26$$

By using equation 4-5, this yields:

$$K_c\ \left(1 + 0.5\ \frac{ST}{T_i} + 2\ \frac{Td}{ST}\right) \leq M_v \qquad 4\text{-}27$$

This equation was used to throttle the controller constants utilizing the PATTERN Search technique. It results in a tradeoff between speed of response and noise sensitivity.

One final comment is essential. Equation 4-27 demonstrates dramatically the noise sensitivity of the derivative mode. It is twice as sensitive as the proportional mode and four times as sensitive as the integral mode.

The final tuning parameters were determined by generating tables of tuning parameters using several values of M_v for each of the shields G3 and G1 and implementing these in succession on the actual process until stable process responses were obtained. This value of M_v was used for all temperatures since it is based upon noise considerations and not process characteristics.

Before the final parameters can be implemented, consideration must be given to the type of tuning necessary. In the case of the calorimeter, the primary concern is the load response. This is due to the fact that no set point changes are issued during a run. The controller is designed to control the difference between the sample temperature and shield temperature under changing heat loss conditions; therefore, the load tuning parameters are the ones to be implemented.

Table VI shows the final parameters for the shield G3 along with the process parameters. The value of M_V necessary was 2.5. This is approximately 25% of the deadbeat value. The tabular data are presented as a function of the suppression parameter (integer multiples of 1000 μvs). This means when the suppression value is zero, the thermocouple emf is between zero and 999 μvs, etc. This method was used to key the change in controller values in the computer; therefore, it was used to tabulate the data. Figure IV-9 shows a typical first order response for a set point change and load change using one set of final controller values determined using load tuning.

Table VII shows the final parameters for the shield G1 presented in the same manner as for G3. The value for M_V necessary was 7.5. This is approximately 25% of the value calculated from the deadbeat parameters.

The second order shields were tuned in the same manner as the first order shields but using the PI plus lead compensator as described. Also, throttling was employed here. It was even more important here than with the first order shields because they were much, much slower. Since dead time did not exist for either of these shields, the tendency

TABLE VI

FINAL CONTROLLER PARAMETERS FOR SHIELD G3

Emf (Kμv)	Process Parameters			Load Tuned		
	K	Θ	τ	K_C	T_i	Td
0	11.84	2.5	123.7	2.3	25.0	0.0
1	9.34	0.0	107.5	2.2	15.0	0.0
2	7.29	0.0	91.6	2.2	15.0	0.0
3	5.67	0.0	77.0	2.2	15.0	0.0
4	4.44	0.0	64.2	2.2	15.0	0.0
5	3.51	0.0	53.4	2.2	15.0	0.0
6	2.80	0.0	44.4	2.2	15.0	0.0
7	2.26	0.0	37.0	2.2	15.0	0.0
8	1.84	0.0	31.0	2.2	15.0	0.0
9	1.52	0.0	26.2	2.2	15.0	0.0

Process Parameters

$K = 9.34 \quad \Theta = 0 \quad \tau = 107.5$

Controller Parameters

$K_c = 2.2 \quad T_i = 15.0$

$ST = 2.5 \quad M_V = 2.5$

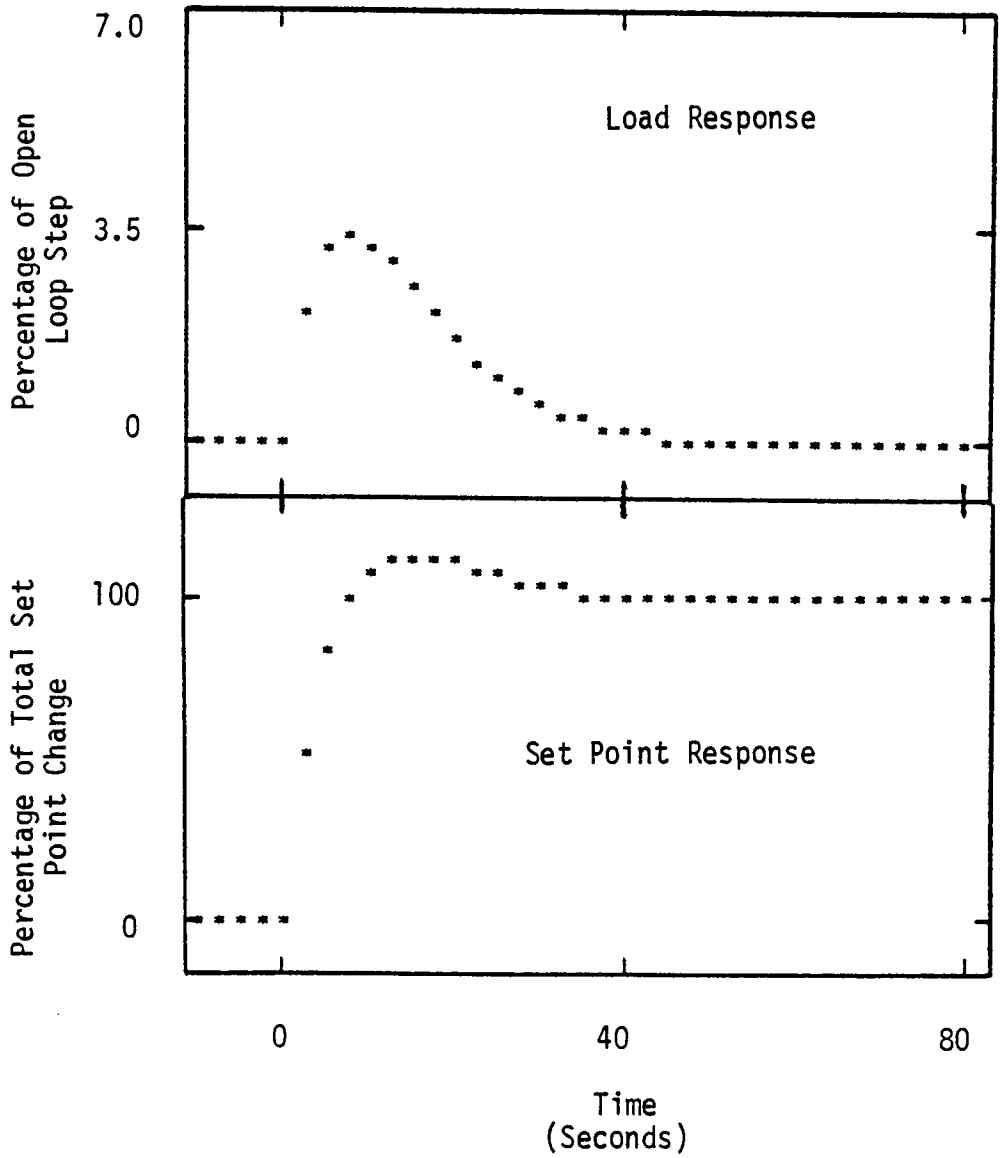


Figure IV-9. Typical First Order Response to a Set Point and Load Change Using Load Tuning and a Throttled Controller.

TABLE VII

FINAL CONTROLLER PARAMETERS FOR SHIELD G1

Emf (Kμv)	Process Parameters			Load Tuned		
	K	Θ	τ	K_C	T_i	Td
0	21.90	15.0	2100.0	8.0	100.0	0.0
1	9.44	5.0	523.0	7.0	25.0	0.0
2	4.91	0.0	203.0	6.5	10.0	0.0
3	2.87	0.0	100.0	6.5	10.0	0.0
4	1.82	0.0	55.6	6.5	10.0	0.0
5	1.22	0.0	34.3	6.5	10.0	0.0
6	0.86	0.0	22.6	6.5	10.0	0.0
7	0.63	0.0	15.6	6.5	10.0	0.0
8	0.48	0.0	11.3	6.5	10.0	0.0
9	0.37	0.0	8.4	6.5	10.0	0.0

of the optimization technique to drive toward the deadbeat response produced extremely high gains. Due to the controller limits placed on the controller during tuning, the deadbeat response could not be achieved, but the controller had such high gains that it banged back and forth between the upper and lower limits much like a bang-bang controller. The parameters obtained were so noise sensitive that any degree of control on the actual process was impossible.

The velocity limit for the PI plus lead compensator was determined due to error swings in exactly the same way as previously discussed. The controller swings due to the error string (+1, -1, +1, -1) produces a change (from equation 4-18) of:

$$\Delta M = K_c [C_2 (+1) - C_3 (-1) + C_4 (+1)] \qquad 4\text{-}28$$

just as in equation 4-23. Therefore, the controller constraint is again:

$$K_c (C_2 + C_3 + C_4)/2 \leq M_v \qquad 4\text{-}29$$

From equations 4-20 to 4-22, this becomes:

$$\frac{K_c K_L [1 + (\alpha + \beta) + \alpha\beta]}{\beta 2} \leq M_v \qquad 4\text{-}30$$

which yields

$$\frac{K_c K_L (1 + \alpha) (1 + \beta)}{2 \beta} \leq M_v \qquad 4\text{-}31$$

By eliminating β using equation 4-15, this equation in terms of the controller parameters becomes:

$$K_c K_L \left(2 + \frac{ST}{T_i}\right) (1 + \alpha)/2 \leq M_v \qquad 4\text{-}32$$

This equation was used to throttle the second order process.

As in the case of the first order process, tables of parameters were generated for several values of M_V and implemented until a stable process response was obtained. Also, the parameters implemented were obtained from load tuning.

Table VIII shows the final parameters for the shield Cyl along with the process parameters. The value of τ_p is also displayed (lower of the two process time constants), although it was not optimized. The optimized parameters are K_C, T_i, and K_L. The value of M_V necessary was 10.0. Once again the parameters are tabulated against the suppression parameter. Figure IV-10 shows a typical second order response for a load and set point change using one set of the final controller values obtained using load tuning.

Table IX shows the final parameters for the shield G2 along with the process parameters. The value of M_V necessary to achieve good control was 10.0 for this shield also.

Final Adjustment of Controller Parameters During Operation

Tables VI, VII, VIII, and IX contain the final parameters actually implemented on the process after fine tuning by hand. These differ only slightly from those obtained by simulation using throttling of the controllers.

In the case of the first order shields at low temperatures where dead time existed in the model, the tuning introduced derivative mode. But under actual operation, this mode actually reduced the quality of the response because of its noise sensitive nature. Therefore, straight PI control was used in both the dead time and no dead time case.

TABLE VIII

FINAL CONTROLLER PARAMETERS FOR SHIELD CYL

Emf (Kμv)	Process Parameters				Load Tuned			
	K	Θ	ξ	$1/W_n$	K_C	T_i	K_L	τ_P*
0	26.7	0.0	1.015	660	10.7	20.0	0.46	554
1	26.2	0.0	1.022	620	10.7	20.0	0.46	503
2	25.5	0.0	1.029	570	10.7	20.0	0.46	449
3	24.8	0.0	1.035	520	10.7	20.0	0.46	400
4	23.9	0.0	1.042	468	10.7	20.0	0.46	350
5	22.9	0.0	1.049	417	10.7	20.0	0.46	305
6	21.8	0.0	1.055	368	10.7	20.0	0.46	265
7	20.6	0.0	1.062	324	10.7	20.0	0.46	228
8	19.4	0.0	1.069	284	10.7	20.0	0.46	196
9	18.2	0.0	1.075	249	10.7	20.0	0.46	170

*Not optimized, equal to lower process time constant.

Process Parameters

$K = 22.9 \quad \Theta = 0.0 \quad \xi = 1.049 \quad 1/W_n = 417$

Controller Parameters

$K_c = 10.65 \quad T_i = 35.0 \quad K_L = 0.455 \quad \tau p = 305$

$ST = 2.5 \quad M_V = 10$

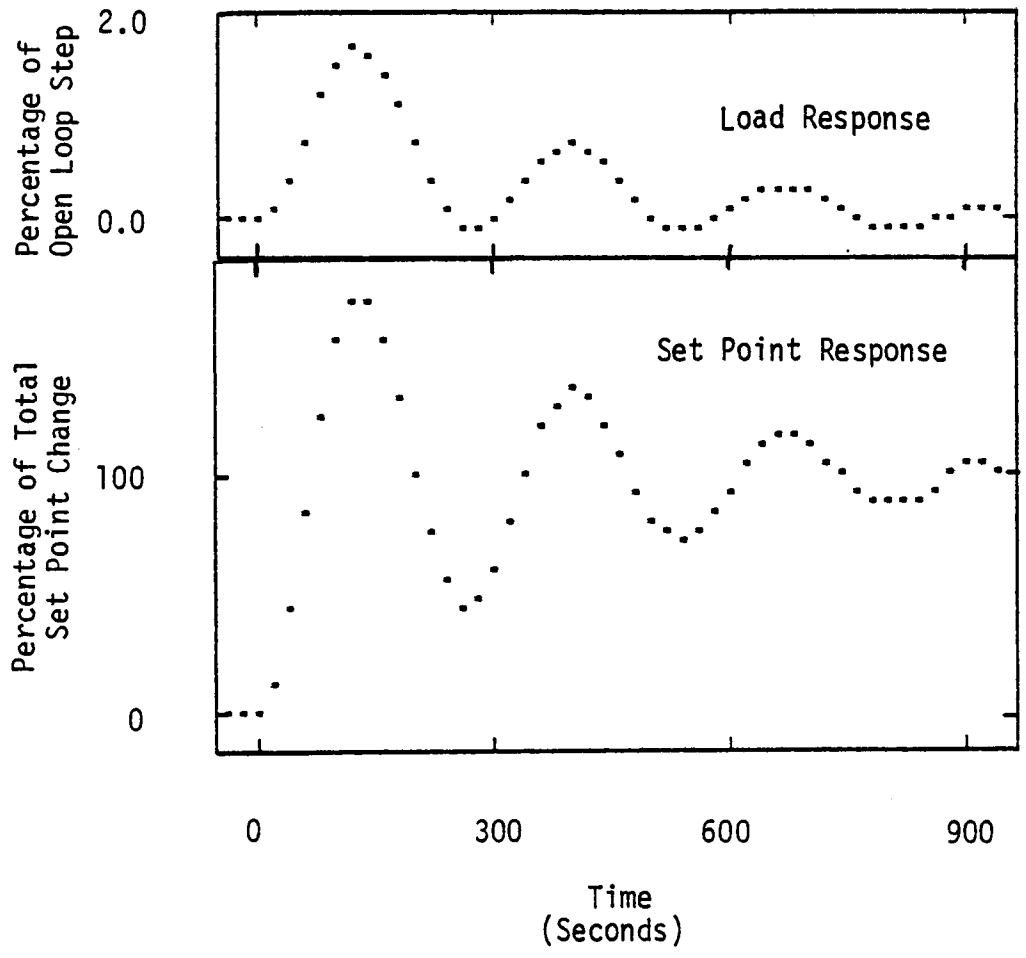


Figure IV-10. Typical Second Order Response to a Set Point and Load Change Using Load Tuning and a Throttled Controller.

TABLE IX

FINAL CONTROLLER PARAMETERS FOR SHIELD G2

Emf (Kμv)	Process Parameters				Load Tuned			
	K	Θ	ξ	$1/W_n$	K_C	T_i	K_L	τp^*
0	16.3	0.0	1.202	411	10.6	30.0	0.46	220
1	15.4	0.0	1.202	361	10.6	30.0	0.46	193
2	14.2	0.0	1.202	311	10.6	30.0	0.46	166
3	13.0	0.0	1.202	263	10.6	30.0	0.46	141
4	11.7	0.0	1.202	221	10.6	30.0	0.46	119
5	10.4	0.0	1.202	184	10.6	30.0	0.46	95
6	9.2	0.0	1.202	154	10.6	30.0	0.46	83
7	8.0	0.0	1.202	128	10.6	30.0	0.46	65
8	7.0	0.0	1.202	108	10.6	30.0	0.46	58
9	6.1	0.0	1.202	91	10.6	30.0	0.46	47

*Not optimized, equal to lower process time constant.

The parameters predicted using simulation for the second order shields behaved as expected, but slightly more integral action was employed than predicted by simulation (implies T_i is smaller). An integral constant of 40 was predicted from simulation as opposed to the values in the tables.

One final note is necessary regarding the appearance of the tabled controller constants. These values are almost constant over all temperature values for all the controllers. Only slight variations occur in the first order shields and none in the second order. This is due to the throttling of the controllers and similar dependence of the parameter on temperature. As the throttling velocity dropped below the deadbeat values, the controller tuning parameters became identical within experimental limits (variations of less than 5%). Therefore, the tuning parameters for the controllers seem much more dependent on noise considerations (which remain constant with time) and the relative values of the process parameters to each other rather than on the absolute values of the process parameters. This is also borne out by examining Tables IV and V with the unthrottled deadbeat parameters for the first order shields. Their variation is due to changes in the ratio of the process parameters rather than the absolute values.

Summary

The final control strategy implemented on the process consisted of four separate control loops. The process interactions were not of sufficient importance to warrant special control attention. The controllers implemented were PI for the first order lag plus dead time processes, and PI plus lead compensator for the second order lag

plus dead time processes. The velocity of these algorithms was throttled in order to achieve stable control due to the tendency of the tuning method to drive to a deadbeat response. The final tuning parameters for each of the shields were tabulated against the suppression parameter which is a function of thermocouple emf. This parameter was also used to key the change in the controller parameters during actual operating conditions. Control levels of $\pm$ 4 ADC counts (which is equivalent to $\pm$ 1.0 thermocouple μv and $\pm$ 0.1° C in temperature) were achieved by using this method.

CHAPTER V

SYSTEM SOFTWARE AND OPERATION

Two main computer systems were used to gather and analyze data from the calorimeter. These were the departmental computer (a PDP-15 with an RSX real-time monitor system and a KM15-V5A off-line monitor system) and the main computer for The University of Tennessee, Knoxville (a DEC-10).

The PDP-15 was used to gather all the data for heat capacity calculations, modeling, and control. The RSX real-time monitor was used to gather all these data. It was also used to do all the heat capacity analysis except fitting the final data to the standard heat capacity equation. This fit was made using the KM15-V5A off-line monitor due to the fact that the corrected heat capacity was printed on a teletype with paper punch capability; therefore, it was easier to use the KM15 monitor than to transfer these data to Dectape and read it into the RSX system. All other data calculations were accomplished inside the RSX system directly, either on an on-line or off-line basis. The programs used to complete these tasks are listed in Table X with their functions. Appendix G contains a complete listing of all these programs while Appendix H lists all the variables and their definitions which appear in the common blocks. All of these programs have been placed on Dectapes along with the RSX system tapes and have been given to Dr. Charles F. Moore.

Also, a complete operating manual for the calorimeter appears in

TABLE X

PDP-15 PROGRAM NAMES AND FUNCTIONS
RSX MONITOR

Program	Function
	Initialization
STUPP	- Turn on clock - Turn on A/D converter channels - Start scan of A/D converter channels - Set relays to proper positions
RSTUP	- Set K, MM, ITK, ISEC, INEC, IVS, and ICS to zero - Set controller errors to zero and controller outputs to minimum values - Turn on D/A converter channels - Reset relays to proper positions
HTCAP	- Starts heat capacity run by asking questions concerning heat capacity information and type control to be used (analog or digital) - Fixes and requests appropriate programs based upon type control desired
	Common Variable Assignment and Display
CALØØ	- Displays line assignments, SMASS, NTICK, IPCR, ITCR, NTCCP, GAIN, and SUP values - Displays controller parameters OF, FC, MI, MA, and IRNG
CALØ1	- Used to change line assignments, SMASS, SUP, and GAIN
CAAØ1	- Used to change variables NTICK, K, NTCCP, ITCR, IPCR, and MM
GPFAC	- Used to change variables OF, FC, MI, MA, and IRNG
FOCCO	- Used to change first order shield controller parameters GC, TI, and TD
SOCCO	- Used to change second order shield parameters GC, TI, and GL
CURCV	- Used to list current controller parameters for all four controllers and the corresponding algorithm constants

TABLE X (Continued)

Program	Function
ALLCV	- Used to list all controller parameters for all controllers and the corresponding algorithm constants from 0°C to 1000°C. It cannot be run during any data collection.
	Modeling and Control
DDCAL	- Takes all thermocouple data during digital control - Calculates errors between shield and sample temperatures in ADC counts - Executes controller algorithm and sets controller outputs on DAC - Reads sample voltage and current - Requests ERROR, resumes STORE, and requests SUPRS if necessary
ERROR	- Sets error values on DAC
APUFO	- Updates controller parameters for first order shields G3 and G1
APUSO	- Updates controller parameters for second order shields Cy1 and G2
CPCFO	- Calculates algorithm constants from controller parameters for first order shields
CPCSO	- Calculates algorithm constants from controller parameters for second order shields
STORE	- Writes all eight shield temperatures, K, four controller outputs, and sample voltage and current on LUN 19.
STEP	- Used to issue step to any of the controllers - At end, resumes normal control for the tested shield and closes data file.
RESP	- Used off-line to set up data for modeling programs from data files created by STORE
	Heat Capacity Calculations
CALØ4	- Runs during analog control - Reads 2 sample thermocouples, sample voltage and current

TABLE X (Continued)

Program	Function
	- Increments T/C table to save last 20 values - If NTCCP have been collected . Reads time since last heat capacity calculation . Stores sample voltage and current . Calculates sums for least squares fit . Initialize counter for NTCCP to zero . Request CALØ6, CALØ8, and CALØ9. - If suppression necessary, request SUPRS
CALØ5	- Same as CALØ4 but runs under digital control - Obtains T/C values from DDCAL rather than collecting data directly
CALØ6	- Calculates temperature at begin and end of 20 point interval from least squares sums - Calculates sample voltage and current from integer values summed over NTCCP values - Calculates heat capacity
CALØ8	- Calculates sigma on least squares fit on 20 T/C values - Writes K, IVSA, ICSA, IDTK, IDSEC, and 20 T/C values on LUN 6 - Writes sigma on LUN 8
CALØ9	- Writes temperatures at begin and end of 20 point interval on LUN 8 - Writes average and delta temperature for interval on LUN 8 - Writes Cp on LUN 8
RADFR	- Runs off-line - Read raw data put on LUN 6 by CALØ8 after transfer to Dectape - Filters raw T/C data if necessary - Does all functions of CALØ4 after reading data
CAAØ8	- Similar to CALØ8 but writes data on LUN 4 in a more presentable form
CAAØ9	- Similar to CALØ9 - Corrects raw Cp data for heater mass and drift rate - Outputs same data as CALØ9 on LUN 4 and then corrected Cp and average temperature on LUN 6

TABLE X (Continued)

Program	Function
	General
CAA1Ø	- Reads all eight thermocouple values and calculates the corresponding temperatures
CAB1Ø	- Reads T/C values out of common from DDCAL and calculates the corresponding temperatures
WRA1Ø	- Writes temperatures generated by CAA1Ø and CAB1Ø on LUN 4
CAL1Ø	- Reads all eight thermocouple values - Calculates corresponding emfs - Writes results on LUN 4
SUPRS	- Sets variable suppression
CAL1B	- Used to calibrate data acquisition system gain and suppression
	KM15-V5A Monitor
CPFIT	- Fits corrected heat capacity data to standard equation using least squares

Appendix I. It includes instructions for heat capacity runs and subsequent analysis as well as instructions for collecting modeling data and its subsequent analysis. Provisions are also made for completing runs under analog control if necessary.

The DEC-10 computer was utilized to do all the analysis of modeling data since the PDP-15 was too small and too slow to accomplish this. The programs used in the analysis are listed in Table XI along with their functions. These programs are too lengthy (> 500 lines each) to present here, but listings of these programs have also been left with Dr. Charles F. Moore.

TABLE XI

DEC-10 PROGRAM NAMES AND FUNCTIONS

Program Name	Function
	Modeling
MOD1.F10	- Used to fit step test data to a first order lag plus dead time model
MOD2.F10	- Used to fit step test data to a second order lag plus dead time model
GRAPH.F10	- Used to plot all data points in a step test data file to check for bad points
CPLST.F10	- Prints differences between shield and sample temperatures and controller outputs - Plots these values for all four shields
	Controller Tuning
TUN.F10	- Tunes a PI (NP=2) or PID (NP=3) controller for a first order lag plus dead time process
TUN2.F10	- Tunes a PI (NP=2) or PID (NP=3) controller for a second order lag plus dead time process
LDLG.F10	- Tunes a PI + lead compensator (NP=3) controller for a second order lag plus dead time process

CHAPTER VI

HEAT CAPACITY OF TITANIUM

The digital control system developed for the calorimeter can only be evaluated through the heat capacity data. This is due to that fact that comparison of errors between the analog system and digital cannot be made because interactions between the analog control system and digital control system inject large amounts of noise into the digital signal. This means the comparison can only be made using the error in heat capacity as an indicator.

From the beginning, two problems had to be solved in order to digitally operate the calorimeter from a heat capacity standpoint. These two problems were due to:

1) Data collection and

2) Control.

The data had to be sampled accurately enough to allow precise measurements of the heat capacity. From a control standpoint, poor control would result in violations of the zero heat loss assumption which governs the heat capacity calculation. Therefore, if a good job has been done in both areas, the accuracy in the heat capacity should be within the $\pm$ 1.0% goal originally sought.

Figure VI-1 shows the heat capacity of titanium using the digital control and data acquisition system developed in this work. One run was made under conventional analog control for comparison purposes. The heat capacity data shown here have been corrected for

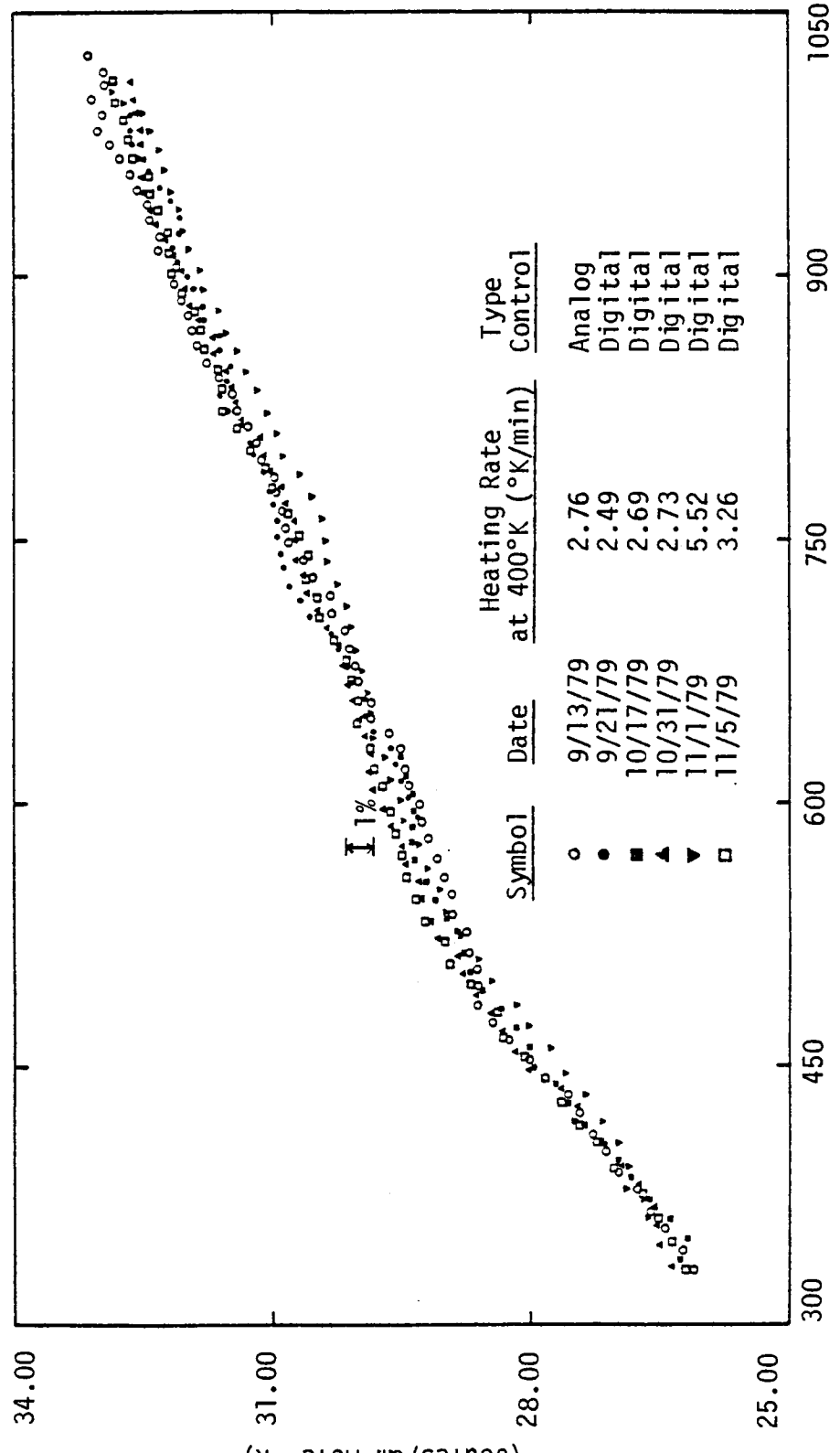


Figure VI-1. Heat Capacity of Titanium from 320 to 1020K.

drift rate and heater mass using the appropriate programs in Table X. Figure VI-1 shows that the data does indeed lie within the desired accuracy of ± 1.0% for both the digital and analog control.

These runs were made using a titanium sample with a mass of 144.012 grams at the various heating rates listed on Figure VI-1. The data for all runs was fitted to the standard form of the heat capacity equation:

$$C_p = a + bT + cT^{-2} \qquad 6\text{-}1$$

where Cp = heat capacity (joules/gm-mole °K)

T = absolute temperature (°K)

a, b, c = constants.

The constants in this equation were determined using least squares fitting (31) and the appropriate programs in Table X. They were determined to be:

a = 25.979

b = 0.007015

c = -248134

The sum of squares of deviations for all 325 points was equal to 9.33. This corresponds to a 95% probability limit of ± 0.330 joules/gm-mole °K. This is approximately ± 1.1% based upon the average value of 30.00 joules/gm-mole °K. All calculations were done on as collected data with no smoothing as was done in previous work by the author (21).

This demonstrates the digital data acquisition and control system is fully capable of replacing the analog based system and, in fact, offers superior data analyzing capabilities under one single system with only minimal hand calculations being necessary.

One final comment regarding the absolute accuracy of the heat capacity data is made. The accuracy of ± 1.1% referred to here is actually a relative accuracy, or scatter. On an absolute scale, these data lie within a ± 2% range of other data which exist for titanium. Historically, this calorimeter has proven to produce heat capacity data with an absolute accuracy of ± 1.0% (44).

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

At the start of this project to place the Dynamic Adiabatic Calorimeter under digital data acquisition and control, one goal was sought. This goal was to determine the heat capacity of materials within an accuracy of ± 1.0% from room temperature to 1000 °C.

This problem was addressed both from a purely data collection standpoint and a control standpoint. Control considerations were of great concern due to the non-linear and multivariable nature of the process.

The data collection problem was solved by developing and characterizing a data acquisition system capable of collecting heat capacity data with an accuracy of less than ± 0.5% theoretically and ± 1.0% practically. This was proven by collection of data of sufficient accuracy for publication (26).

The control problem was attacked by developing a theoretical process model. Using it as a guideline, data were collected and an experimental process model was determined from these data. The experimental model consisted of transfer functions for which the type response and parameters (as functions of temperature due to the non-linear nature of the process) were determined for each shield. This experimental modeling also demonstrated that the multivariable interactions were either non-observable or not of sufficient strength to deteriorate the closed-loop control.

The final control system consisted of single loops for each of the system components. The controllers were selected based upon the type of response and tuned based upon the parameters in these types of responses. These controller tuning parameters were changed in accordance with the non-linear aspects of the process during actual operation.

This system was implemented and used to determine the heat capacity of titanium from 320 to 1020 K. These data fell within the accuracy limits of $\pm$ 1.1%. This demonstrates the success of the data acquisition system and control system.

From the start of this project, a much more complex control effort was envisioned. But as modeling proceeded, understanding of the process increased, and the complexity of the control strategy decreased considerably. It is the opinion of this author that, in general, this axiom will hold. It could also be proposed that seemingly complex systems are really governed by very simple principles. The reduction of the current project to a very simple direct control strategy is by far the major accomplishment of this work.

The recommendations proceeding from this work are:

1) Adopt the digital data acquisition and control package developed here as a standard operating procedure for the Dynamic Adiabatic Calorimeter under study.

2) Devote further study to the transfer of this system to a microprocessor based system.

3) Devote further study to the suggested drift rate model developed in the critical shield analysis.

REFERENCES

REFERENCES

1. Doss, J. E., Master's Thesis, The University of Tennessee, Knoxville, 1973.

2. Doss, J. E., Doctoral Dissertation, The University of Tennessee, Knoxville, 1974.

3. Sayers, H. D., Master's Thesis, The University of Tennessee, Knoxville, 1972.

4. Sayers, H. D., Doctoral Dissertation, The University of Tennessee, Knoxville, 1975.

5. Jones, D. K., Master's Thesis, The University of Tennessee, Knoxville, 1974.

6. Stoots, R. C., Master's Thesis, The University of Tennessee, Knoxville, 1972.

7. Elder, G. E., Master's Thesis, The University of Tennessee, Knoxville, 1952.

8. Pawel, R. E., Doctoral Dissertation, The University of Tennessee, Knoxville, 1956.

9. Nauman, E. B., Master's Thesis, The University of Tennessee, Knoxville, 1960.

10. Brooks, C. R., Doctoral Dissertation, The University of Tennessee, Knoxville, 1962.

11. Yeh, C. C., Master's Thesis, The University of Tennessee, Knoxville, 1972.

12. Peng, Y. C., Master's Thesis, The University of Tennessee, Knoxville, 1976.

13. Stout, K., Master's Thesis, The University of Tennessee, Knoxville, 1976.

14. Kreith, F., Principles of Heat Transfer, International Textbook Co., Scranton, Penn., 1969.

15. Love, T. J., Radiative Heat Transfers, C. E. Merrill Publishing Co., Columbus, Ohio, 1968.

16. Siegel, R., Thermal Radiation Heat Transfer, McGraw-Hill, New York, 1972.

17. Spurrow, E. M., and R. D. Cess, Radiation Heat Transfer, Brooks-Cole Publishing Co., Belmont, Calif., 1970.

18. Murrill, P. W., Automatic Control of Processes, International Textbook Co., Scranton, Penn., 1967.

19. Weber, T. W., An Introduction to Process Dynamics and Control, Wiley-Interscience, New York, 1973.

20. Smith, C. L., Digital Computer Process Control, International Textbook Co., Scranton, Penn., 1972.

21. Cash, W. M., Master's Thesis, The University of Tennessee, Knoxville, 1976.

22. Miller, W. E., Digital Computer Applications to Process Control, Plenum Press, New York, 1965.

23. Savas, E. S., Computer Control of Industrial Processes, McGraw-Hill, New York, 1965.

24. Smith, O.J.M., "Closer Control of Loops with Dead Time," Chemical Engineering Progress, Vol. 53, No. 5, May 1957, p. 217.

25. Smith, O.J.M., "A Controller to Overcome Dead Time," ISA Journal, Vol. 6, No. 2, February 1959, p. 28.

26. Brooks, C. R., W. M. Cash, and A. Garcia, "The Heat Capacity of Inconel 718 from 313 to 1053 K," Journal of Nuclear Materials, 78, 1978, pp. 419-421.

27. Lee, T. H., G. E. Adams, and W. M. Gaines, Computer Process Control: Modeling and Optimization, John Wiley & Sons, New York, 1968.

28. Luyben, W. L., Process Modeling, Simulation, and Control for Chemical Engineers, McGraw-Hill Book Co., New York, 1973.

29. Frank, R.G.E., Modeling and Simulation in Chemical Engineering, Wiley-Interscience, New York, 1972.

30. Moore, C. F., Doctoral Dissertation, Louisiana State University, 1969.

31. Graupe, D., Identification of Systems, V. N. Reinhold Co., New York, 1972.

32. Wilde, D. J., Optimum Seeking Methods, Prentice-Hall, Englewood Cliffs, N. J., 1964.

33. Gupta, S. C., and L. Hasdorff, Fundamentals of Automatic Control, John Wiley & Sons, New York, 1974.

34. Eveleigh, V. W., An Introduction to Control Systems Designs, McGraw-Hill, New York, 1972.

35. McCausland, I., Introduction to Optimal Control, John Wiley & Sons, Inc., New York, 1969.

36. Freeman, H., Discrete-Time Systems, John Wiley & Sons, New York, 1965.

37. Steiglitz, K., An Introduction to Discrete Systems, John Wiley & Sons, New York, 1974.

38. Cadzow, J. A., Discrete Time Systems, Prentice-Hall, Inc., Engelwood Cliffs, N. J., 1973.

39. Morgan, B. S., "The Synthesis of Linear Multivariable Systems by State-Variable Feedback," IEEE Trans. Auto. Control, October 1964, pp. 405-411.

40. Kavanagh, R. J., "Multivariable Control System Synthesis," Trans. IEEE, 77, Part II, 1958, pp. 425-429.

41. Kavanagh, R. J., "The Application of Matrix Methods to Multivariable Control Systems," J. Franklin Inst., 262, No. 5, 1956, pp. 349-367.

42. Mesarovic, M. D., The Control of Multivariable Systems, John Wiley & Sons, New York, 1960.

43. Wonham, W. M., Linear Multivariable Control, Springer-Verlag, New York, 1974.

44. Stansbury, E. E., and C. R. Brooks, "Dynamic Adiabatic Calorimetry," High Temperatures-High Pressures, volume 1, 1969, pp. 289-307.

APPENDICES

APPENDIX A

RATIONALE FOR MODELING THE DIFFERENCE IN THE SHIELD TEMPERATURE AND SAMPLE TEMPERATURE

This appendix serves as an explanation for the rationale behind modeling the difference between the shield temperature and the sample temperature ($T_H - T_S$) as suggested in Chapter III.

In deriving the LaPlace domain model, linearization of the T^4 term necessitated the subtraction of the steady state conditions from the time domain equation (18,19). Normally, steady state implies variables that do not change with time, but in this case the steady state equation was of the form:

$$M_i C_{P_i} \frac{dTa}{d\Theta} = P_a - \sum_{j=1}^{r} K_{ij} (T_{i_a} - T_{j_a}) - \sum_{h=1}^{s} K_{ih} (T_{i_a}^4 - T_{h_a}^4) \qquad A\text{-}1$$

where a indicates the temperature at the instant of linearization. Since the heat loss terms are very small compared to P_a, they can be incorporated into an overall heat loss term Q_{L_a}. Then the difference between P_a and Q_{L_a} can be thought of as a net heat input term Q_{N_a}. Therefore, equation A-1 becomes:

$$M_i C_{P_i} \frac{dTa}{d\Theta} = Q_{N_a} \qquad A\text{-}2$$

to which the LaPlace domain solution is:

$$T_a(S) = \frac{Q_{N_a}(S)}{M_i C_{P_i}} \cdot \frac{1}{S} = \frac{C_1}{S} \qquad A\text{-}3$$

This is the LaPlace equation for a ramp, which indeed it should be.

The solution derived by the author in the Master's thesis (21)

was the transient solution. The overall solution is the sum of the transient solution and the steady state solution. This sum is the observed temperature response for T_H given in Figure III-3. In order to get the transient response only, the steady state solution given by equation A-3 must be subtracted from T_H. This can easily be accomplished by letting the temperature about which the process is linearized be equal to the sample temperature. This leaves the modeled variable to be the difference between the shield temperature and the sample temperature.

Although both temperatures increase with time, their difference under step testing will reach a steady state difference. This behavior can be physically explained by examining the time domain equation for the shield with only one heat loss term:

$$M_H C_{P_H} \frac{dT_H}{d\Theta} = P_H - K (T_H - T_S) \qquad \text{A-4}$$

Before stepping P_H, the term $K (T_H - T_S)$ will have a constant value close to zero. As P_H is stepped, this excess power will result in a sudden jump in $dT_H/d\Theta$ since all the other values are constant. But this jump will result in an increase in $T_H - T_S$. Finally, at steady state, the increase in the term $K (T_H - T_S)$ caused by an increase in $(T_H - T_S)$ will exactly negate the excess power supplied by the step, and the heating rate will again return to its original value.

This argument demonstrates the reasons for modeling $T_H - T_S$ rather than T_H directly.

APPENDIX B

SPECIAL METHOD FOR MODELING PROCESS INTERACTIONS WITH THE NON-TESTED SHIELDS REMAINING IN AUTOMATIC

Normally, step testing is conducted with all the process controllers in manual except the one being tested as shown in Figure III-4. But for reasons explained in Chapter III, this was impossible in the case of the calorimeter.

In order to implement this method, Figure III-5 has been modified to show only the lower control loop and a disturbance input from R_1 (Figure B-1). By knowing the response R_1, the model of $HG_1(Z)$, and the controller algorithm D(Z) (used when the testing was done), this control loop can be simulated with R_1 appearing as a disturbance through the interaction transfer function $HG_2(Z)$.

The response in the controller output $M_2(Z)$ can then be predicted with one value for $HG_2(Z)$. By varying the form of $HG_2(Z)$ and its parameters until the simulated response of $M_2(Z)$ matches the actual response of $M_2(Z)$, the interaction transfer function $HG_2(Z)$ can be determined.

The response in $M_2(Z)$ is used as the key rather than the response in R_2 because it is a more accurate indicator of the interaction. For instance, say $M_2(Z)$ dropped by 20% during a test of R_1, but the lower control loop behaved so well that no response was seen in R_2. If R_2 were used as the key, the obvious answer would be that $HG_2(Z)$ was zero. But, indeed, this would not be the case since the controller output

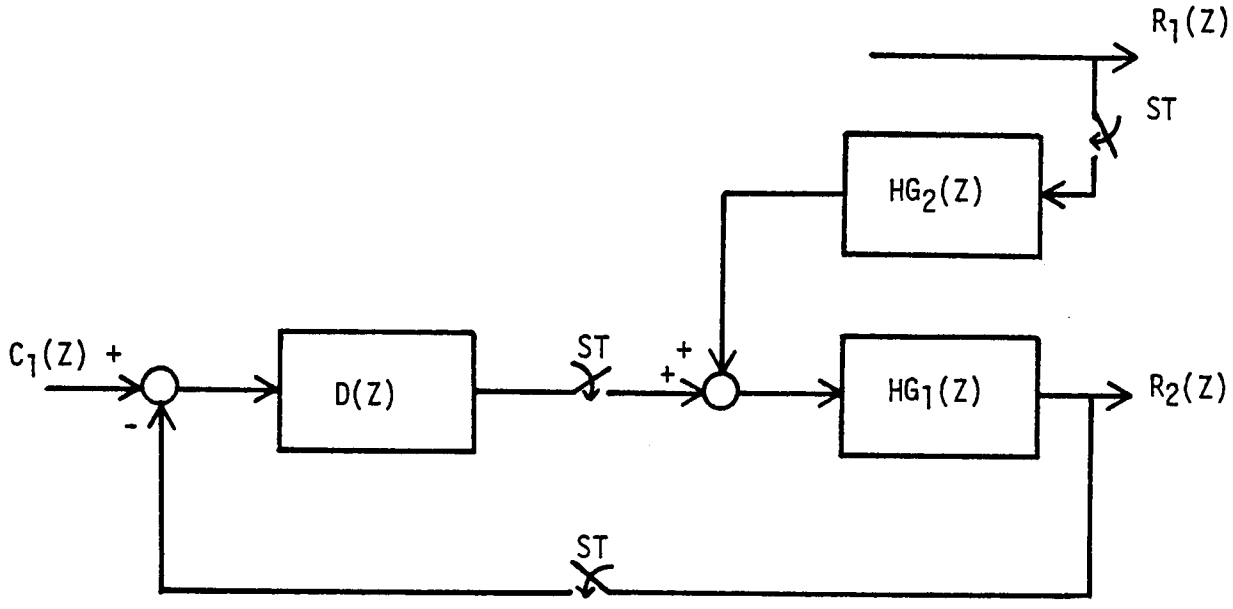


Figure B-1. Control Loop with an Interaction Entering Loop as a Disturbance.

dropped. Therefore, care must be taken not to overlook interactions that may not apparently exist.

APPENDIX C

CHARACTERIZATION OF THE CYL-G1 INTERACTION

The interaction between G1 and the Cyl can be characterized by taking the two time domain equations to the final LaPlace equations. By letting subscript 1 denote the Cyl and subscript 2 denote G1, the following time domain equations can be written:

Cyl

$$M_1 C_{p_1} \frac{dT_1}{d\Theta} = P_1 - K_{12} (T_1^4 - T_2^4) \qquad C\text{-}1$$

G1

$$M_2 C_{p_2} \frac{dT_2}{d\Theta} = P_2 - K_{21} (T_2^4 - T_1^4) \qquad C\text{-}2$$

The steady state equation for C-1 is:

$$M_1 C_{p_1} \frac{dT_{1_a}}{d\Theta} = P_1 - K_{12} (T_{1_a}^4 - T_{2_a}^4) \qquad C\text{-}3$$

In order to linearize the T^4 terms, the standard method of partials can be used (18). Therefore,

$$T_i^4 = T_{i_a}^4 + \frac{\partial T_i^4}{\partial T}\Big|_a (T_i - T_{ia}) \qquad C\text{-}4$$

and

$$T_i^4 - T_{i_a}^4 = 4T_{ia}^3 (T_i - T_{ia}) \qquad C\text{-}5$$

When equation C-3 is subtracted from C-1, the result is:

$$M_1 C_{p_1} \frac{d (T_1 - T_{1a})}{d\Theta} = (P_1 - P_{1a}) - K_{12} [(T_1^4 - T_{1_a}^4) - (T_2^4 - T_{2_a}^4)] \qquad C\text{-}6$$

By substituting equation C-5 with i being 1 and 2 into equation C-6, and

by denoting deviations from steady state by small letters, equation C-6 becomes:

$$M_1 C_{P_1} \frac{dt_1}{d\Theta} = P_1 - K_{12} (4T_{1_a}^3 t_1 - 4T_{2_a}^3 t_2) \qquad C\text{-}7$$

Since the temperature about which linearization occurred was the sample temperature, $T_{1_a} = T_{2_a}$, and equation C-7 becomes:

$$M_1 C_{P_1} \frac{dt_1}{d\Theta} = P_1 - 4T_a^3 K_{12} (t_1 - t_2) \qquad C\text{-}8$$

Also the steady state equation C-3 becomes:

$$M C_P \frac{dT_a}{d\Theta} = P_a \qquad C\text{-}9$$

to which the LaPlace solution is:

$$T_a(S) = \frac{P_a(S)}{M C_P} \cdot \frac{1}{S} = \frac{K_R}{S} \qquad C\text{-}10$$

Therefore, from C-8, the following is obtained:

$$t_1(S) = \frac{P_1(S) + 4T_a^3 K_{12} t_2}{M_1 C_{P_1} S + 4T_a^3 K_{12}} \qquad C\text{-}11$$

and

$$t_1(S) = \frac{K_1}{1 + \tau_1 S} (P_1(S) + C_{12} t_2(S)) \qquad C\text{-}12$$

where $K_1 = 1 / 4T_a^3 K_{12}$

$\tau_1 = M_1 C_{P_1} / 4T_a^3 K_{12}$

$C_{12} = 4T_a^3 K_{12}$

In a similar manner, equation C-2 becomes:

$$t_2(S) = \frac{K_2}{1 + \tau_2 S} (P_2(S) + C_{21} t_1(S)) \qquad C\text{-}13$$

These equations are displayed in block diagram form in Figure C-1. This is the V-Canonical form of the model. It also demonstrates that the interaction terms are pure gains. These gains are of the form:

$$C_{ij} = 4T_a^3 K_{ij} = AT_a^3 \qquad \text{C-14}$$

where T = absolute temperature (°K)

A = constant

C_{21} can be found very easily if t_2 does not deviate from its controlled value during the step testing of t_1 by letting equation C-13 be equal to zero. This can only occur if:

$$P_2(S) + C_{21}\, t_1(S) = 0 \qquad \text{C-15}$$

Therefore,

$$C_{21} = - \frac{P_2(S)}{t_1(S)} \qquad \text{C-16}$$

Thus, a plot of the process input, i.e., controller output ($P_2(S)$) vs the temperature response of $t_1(S)$ would have a slope of $- C_{21}$. This is a direct means of finding the interaction gain as opposed to a complex simulation as suggested in Appendix B.

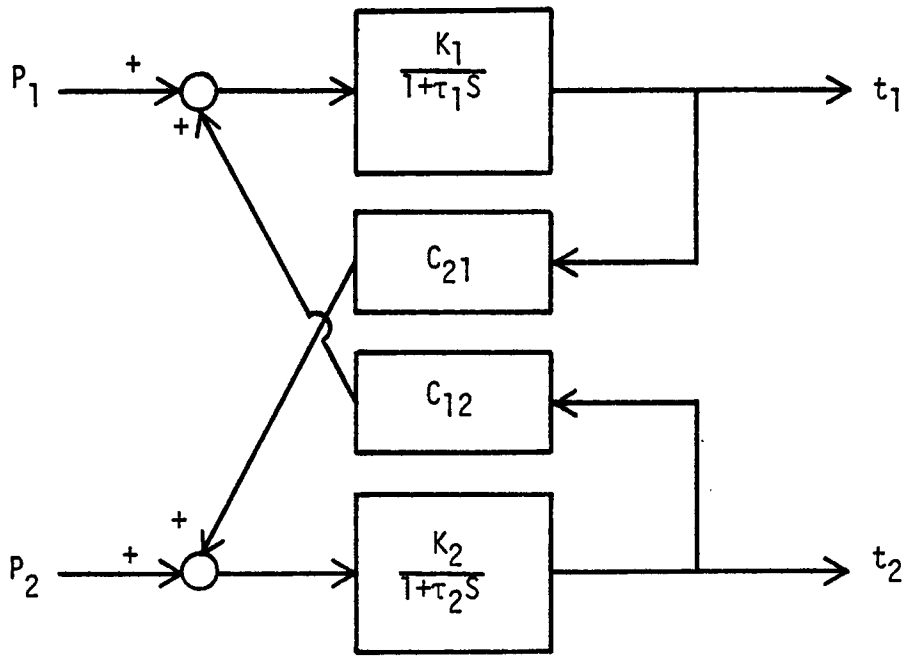


Figure C-1. Block Diagram for Cyl-Gl Type Interaction.

APPENDIX D

SAMPLE DRIFT RATE MODEL

Based upon data collected for the critical shield identification, a completely new approach to modeling the sample drift rate was suggested. That model is developed here in its entirety.

General Drift Model

Although Figure III-30 shows the effect of shield offset on drift rates, it is also the basis for an extended drift rate model. By rearranging equation 3-21 in Chapter III, the drift rate can be calculated for one shield by:

$$D_s = \frac{4\ T_a^3\ K_j}{M_s\ C_{P_s}}\ (t_j - t_s) \qquad \text{D-1}$$

Close examination of this equation, it suggests that a plot of sample drift rate vs offset in shield temperature (Figure III-30) has a slope of $4\ T_a^3\ K_j\ /\ M_s\ C_{P_s}$. Therefore, the heat transfer coefficient K_j can be calculated to be:

$$K_j = \frac{(\text{slope})\ M_s\ C_{P_s}}{4\ T_a^3} \qquad \text{D-2}$$

The values determined in this manner appear in Table XII. By realizing that equation D-1 is determined from only one term in the overall equation 1-3 and that it has a non-zero intercept (as seen in Figure III-30), a generalized model for the sample drift rate can be written as:

TABLE XII

PSEUDO-HEAT TRANSFER COEFFICIENTS FROM DRIFT RATE ANALYSIS

Shield	K_j (Joules/°K^4/Hr)
G3	2.40×10^{-8}
G1	3.39×10^{-7}
Cyl	5.20×10^{-10}
G2	1.30×10^{-8}

$$D_s = [D_o + K_{G3}(t_{G3} - t_s) + K_{G1}(t_{G1} - t_s) + K_{Cyl}(t_{Cyl} - t_s) + K_{G2}(t_{G2} - t_s)] \quad \text{D-3}$$

The value of D_o is determined when all offsets are zero, i.e., the intercept of Figure III-30, from:

$$D_o = \frac{(\text{intercept})\ M_s\ C_{P_s}}{4Ta^3} \quad \text{D-4}$$

In this case, $D_o = -2.65 \times 10^{-7}$ joules/°K^3/hr.

It should be noted here that the heat transfer coefficients are only slight functions of temperature. The drift rate at zero offset conditions (D_o) is a function of thermocouple placement; therefore, it must be redetermined if thermocouples are moved.

Table XIII contains all the raw data from which Figure III-30 was constructed and from which the subsequent constants were calculated.

The result of the drift rate model given in equation D-3 is a method by which drift rates can be predicted using any sample at any temperature after D_o is redetermined if thermocouple locations are changed.

Drift Elimination Using G1 Offset

One very interesting and very important fact remains to be examined. This is the elimination of drift by using shield offsets. Since G1 is the most responsive, it is logical to use it to accomplish this effect.

By letting all other offsets be zero, equation D-3 can be set to zero to yield:

$$t_{G1} - t_s = -\frac{D_o}{K_{G1}} \quad \text{D-5}$$

TABLE XIII

DATA USED IN THE DETERMINATION OF DRIFT RATE PARAMETERS

Shield	Absolute Values			Offset		Drift Rate	
	(μv)	(°K)	(μv/°K)	(μv)	(°K)	(μv/sec)	(°K/hr)
All	7370	1027.5	12.05	0	0.000	-0.0422	-12.61
G3	7330	1024.2	12.03	+10	0.831	-0.0390	-11.67
	7290	1020.9	12.02	+20	1.664	-0.0359	-10.75
G1	7230	1015.9	11.99	+10	0.834	+0.0013	+0.39
	7260	1018.4	12.00	+20	1.666	+0.0450	+13.5
Cyl	7260	1018.4	12.00	+20	1.666	-0.0419	-12.57
G2	7220	1015.1	11.99	+20	1.668	-0.0387	-11.62

M_S = 144.012 gms.

C_{P_S} = 0.635 joules/gm °K.

This is the offset in G1 necessary to achieve zero drift conditions. Using the values determined previously, this value becomes 0.782 °K which, using the thermocouple tables, corresponds to 9.38 μvs. This is the value obtained from Figure III-30.

Suggested Procedure for Drift Rate Model Enhancement

Following this procedure, the author suggests that an in-depth look be taken using the drift rate model developed here. This involves collecting extensive drift rate data under conditions similar to those used here but using greater care in accuracy and several temperatures to test the temperature dependence of the heat transfer coefficients. Also, different samples could be used to check the validity of the extension of the model to other samples using the same thermocouple locations. Apparent break-down in the model might occur here due to the fact that changing samples might cause a shift in sample thermocouple relative to the others. This violates the constant thermocouple location principles which makes D_o a constant. In the worst case, once a sample is mounted on the "tree," quick tests could be run to establish values of the constants.

It should be noted that this model aids in the understanding of the calorimeter more than anything else. The true solution to zero drift lies in the quick and dirty solution of heating any new sample or thermocouple location to an elevated temperature, say 800°C, and adjusting G1 offset until zero drift (or any desired value) occurs.

APPENDIX E

DETERMINATION OF THE DISCRETE VERSION OF A LEAD COMPENSATOR USING THREE METHODS

As stated in Chapter IV, the lead compensator can be put into discrete form in several ways. The first method is to let the LaPlace operator s revert to a derivative and then be approximated by a difference. Using the lead compensator in the form of equation 4-6 in Chapter IV,

$$\frac{M(s)}{e(s)} = K_1 \cdot \frac{1 + \tau_1 s}{1 + \tau_2 s} \qquad \text{E-1}$$

the time domain differential becomes:

$$m(t) + \tau_2 \frac{dm(t)}{dt} = K_1 \left[e(t) + \tau_1 \frac{de(t)}{dt}\right] \qquad \text{E-2}$$

which can be written in the discrete domain as:

$$M_i + \frac{\tau_2}{ST}(M_i - M_{i-1}) = K_1 \left[e_i + \frac{\tau_1}{ST}(e_i - e_{i-1})\right] \qquad \text{E-3}$$

In the z-domain, this becomes:

$$(1 + A - AZ^{-1})\, M(Z) = K_1 (1 + B - BZ^{-1})\, E(Z) \qquad \text{E-4}$$

Therefore,

$$\frac{M(Z)}{e(Z)} = K_1 \cdot \frac{1 + B}{1 + A} \cdot \frac{1 - \alpha Z^{-1}}{1 - \gamma Z^{-1}} \qquad \text{E-5}$$

where $A = 1 + \frac{\tau_2}{ST}$ E-6

$B = 1 + \frac{\tau_1}{ST}$ E-7

$\alpha = B / (B + 1)$ E-8

$$\gamma = A / (A + 1) \qquad \text{E-9}$$

The second method involves taking the z-transform of the lead compensator in series with a zero order hold. (During this derivation the sample time will be denoted by t.) This is written as:

$$\frac{M(Z)}{e(Z)} = Z\left[\frac{1 - e^{-st}}{s} \cdot K_1 \cdot \frac{1 + \tau_1 s}{1 + \tau_2 s}\right] \qquad \text{E-10}$$

which leads to:

$$\frac{M(Z)}{e(Z)} = K_1 \frac{\tau_1}{\tau_2}(1 - Z^{-1})\, Z\left[\frac{s + a}{s\,(s + b)}\right] \qquad \text{E-11}$$

$$\text{where } a = 1/\tau_1 \qquad \text{E-12}$$

$$b = 1/\tau_2 \qquad \text{E-13}$$

By use of the fact that

$$F(Z) = \sum_{\text{poles}} \left[\frac{F_{(P)}}{1 - Z^{-1} e^{pt}}\right] \qquad \text{E-14}$$

it can be shown that the z-transform of the group in brackets in equation E-11 is:

$$\frac{1}{b}\frac{b + (a - b - ae^{-bt})\, Z^{-1}}{(1 - Z^{-1})(1 - e^{-bt}\, Z^{-1})} \qquad \text{E-15}$$

By adding the terms from equation E-11 to E-13 and performing the necessary algebra, the final form is:

$$\frac{M(Z)}{e(Z)} = K_1 \frac{\tau_1}{\tau_2}\frac{1 - \alpha Z^{-1}}{1 - \gamma Z^{-1}} \qquad \text{E-16}$$

$$\text{where } \alpha = \frac{b + ae^{-bt} - a}{b} \qquad \text{E-17}$$

$$\gamma = e^{-bt} \qquad \text{E-18}$$

The third method of developing the z-domain lead compensator was developed in Chapter IV and will not be rederived here.

These methods can be compared by picking a set of actual values and inserting them in the equations using:

$$K_p = 20 \qquad \tau_p = 400$$

$$K_L = 0.45 \qquad ST = 2.5$$

The lead compensator using the LaPlace feedback system yields from equations 4-9, 4-10, and 4-11:

$$K_1 = 0.045 \qquad \tau_1 = 400 \qquad \tau_2 = 40$$

Inserting these in equations E-6 through E-9, the following values are obtained:

$$A = 17 \qquad B = 161$$

$$\alpha = 0.9938 \qquad \gamma = 0.9444$$

which can be placed into the final equation E-5 as:

$$\frac{M(Z)}{e(Z)} = 0.405 \frac{1 - 0.9938\ Z^{-1}}{1 - 0.9444\ Z^{-1}} \qquad \text{E-14}$$

This is the z-domain equation obtained using method 1 of discrete differences. Using these same values in equations E-12, E-13, E-17, and E-18, the following values are obtained:

$$a = 0.0025 \qquad b = 0.025$$

$$\alpha = 0.9939 \qquad \gamma = 0.9394$$

When placed in the final equation E-16, the following is obtained:

$$\frac{M(Z)}{e(Z)} = 0.45 \frac{1 - 0.9939\ Z^{-1}}{1 - 0.9394\ Z^{-1}} \qquad \text{E-15}$$

This is the z-domain equation obtained using method 2 of a zero-order hold in series with the LaPlace lead compensator.

Method 3 described in Chapter IV yields the values (from equations 4-12 and 4-16):

$$\alpha = 0.9938 \qquad \gamma = 0.9377$$

These values, when placed in equation 4-12, yield:

$$\frac{M(Z)}{e(Z)} = 0.45 \frac{1 - 0.9938\ Z^{-1}}{1 - 0.9377\ Z^{-1}} \qquad \text{E-16}$$

This is the z-domain equation obtained using method 3 of a feedback system around the z-domain first order lag as described in Chapter IV.

In summary, all these methods yield the same result. Equations E-15 and E-16 are identical while equation E-14 has a slightly lower gain and a slightly larger value in the pole. But equation E-14 has a steady state gain of 0.0451 (found by evaluating equation E-14 with Z^{-1} = 1) which is the same as that obtained from equation E-15 and E-16.

All the final control algorithms are formed by placing the PI z-domain equation in front of them. This would be done exactly as described in Chapter IV.

APPENDIX F

FIRST ORDER DEADBEAT ALGORITHM

Figure F-1 shows a typical digital control loop. The output can be written as:

$$C(Z) = \frac{D(Z)\ HG(Z)\ R(Z) + HG(Z)\ N(Z)}{1 + D(Z)\ HG(Z)} \qquad \text{F-1}$$

If the set point response is to be studied, N(Z) is set to zero, and equation F-1 can be solved for D(Z) as follows:

$$D(Z) = \frac{C(Z)}{HG(Z)\ [R(Z) - C(Z)]} = \frac{C(Z)/R(Z)}{HG(Z)\ [1 - C(Z)/R(Z)]} \qquad \text{F-2}$$

For a deadbeat set point change with no dead time, the desired value of $C(Z)/R(Z)$ is Z^{-1}. This causes the response to be delayed by one sample time to the set point input. Therefore,

$$D(Z) = \frac{Z^{-1}}{HG(Z)\ (1 - Z^{-1})} \qquad \text{F-3}$$

If the process is a first order lag with no dead time,

$$HG(Z) = \frac{K\ (1 - \alpha)\ Z^{-1}}{1 - \alpha Z^{-1}} \qquad \text{F-4}$$

where $\alpha = e^{-ST/\tau}$

Placing this in equation F-3, the following is obtained:

$$D(Z) = \frac{1}{K\ (1 - \alpha)} \cdot \frac{1 - \alpha Z^{-1}}{1 - Z^{-1}} \qquad \text{F-5}$$

Comparing this to equations 4-14 and 4-15 in Chapter IV, it is found that this is the z-domain form of a PI controller where:

$$\alpha = \beta = \frac{1}{1 + \frac{ST}{T_i}} \qquad \text{F-6}$$

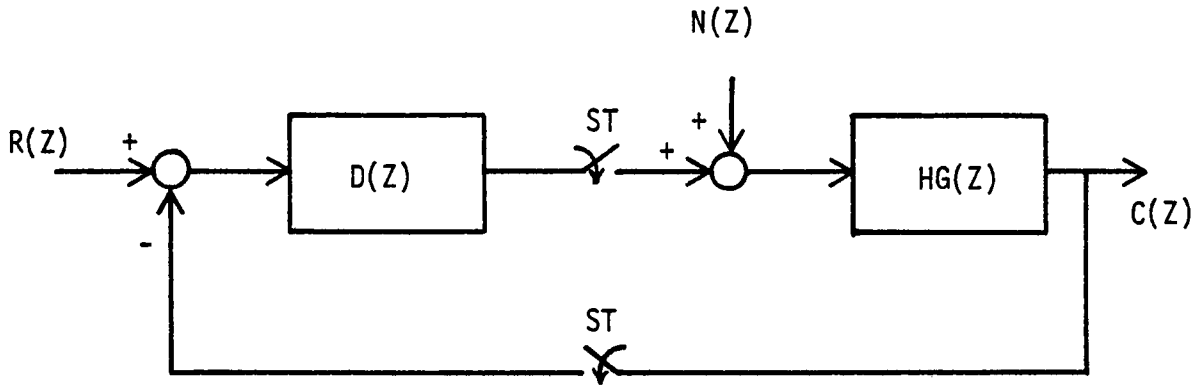


Figure F-1. Digital Control Loop with Noise.

and

$$K_c \left(1 + \frac{ST}{T_i}\right) = \frac{1}{K\,(1 - \alpha)} \qquad \text{F-7}$$

From F-6, the integral constant for the PI controller can be calculated to be:

$$T_i = \frac{\alpha}{1 - \alpha}\, ST \qquad \text{F-8}$$

Using F-6 in F-7, the result is:

$$\frac{K_c}{\alpha} = \frac{1}{K\,(1 - \alpha)} \qquad \text{F-9}$$

which yields

$$K_c = \frac{\alpha}{1 - \alpha} \cdot \frac{1}{K} \qquad \text{F-10}$$

Therefore, the deadbeat algorithm for a first order lag process is a PI controller with parameters which can be calculated from equations F-10 and F-8.

In the case of load tuning, R(Z) in equation F-1 is set to zero, and D(Z) can be calculated as:

$$D(Z) = \frac{HG(Z)N(Z) - C(Z)}{HG(Z)\,C(Z)} = \frac{N(Z)}{C(Z)} - \frac{1}{HG(Z)} \qquad \text{F-11}$$

All that needs to be realized is that for a deadbeat load change to a step change in N(Z), there will be only one non-zero value at a time of Z^{-1}. The magnitude of this value will be $K(1 - \alpha)$ times the magnitude of the step, as seen in equation F-4. Therefore,

$$C(Z) = K\,(1 - \alpha)\,\Delta N Z^{-1} \qquad \text{F-12}$$

where ΔN = magnitude of step of N(Z)

and

$$N(Z) = \frac{\Delta N}{1 - Z^{-1}} \qquad \text{F-13}$$

Using the same process equation in F-4, equation F-11 can be written as:

$$D(Z) = \frac{\Delta N}{(1 - Z^{-1})\, K\, (1 - \alpha)\, \Delta N Z^{-1}} - \frac{1 - \alpha Z^{-1}}{K\, (1 - \alpha)\, Z^{-1}} \qquad \text{F-14}$$

This can be rearranged to:

$$D(Z) = \frac{1 + \alpha}{K\, (1 - \alpha)} \frac{1 - \alpha\, /\, (1 + \alpha)\, Z^{-1}}{1 - Z^{-1}} \qquad \text{F-15}$$

Comparing this to equations 4-14 and 4-15 in Chapter IV, the following results:

$$\frac{\alpha}{1 + \alpha} = \beta = \frac{1}{1 + \frac{ST}{T_i}} \qquad \text{F-16}$$

and

$$\frac{1 + \alpha}{K\, (1 - \alpha)} = K_c\, (1 + \frac{ST}{T_i}) = K_c (\frac{1 + \alpha}{\alpha}) \qquad \text{F-17}$$

The right portion of equation F-17 was obtained from F-16. Equation F-16 leads to an integral constant of:

$$T_i = \alpha \cdot ST \qquad \text{F-18}$$

Equation F-17 leads to:

$$K_c = \frac{\alpha}{1 - \alpha} \cdot \frac{1}{K} \qquad \text{F-19}$$

which is the same as equation F-10. Therefore, the deadbeat algorithm for a load change using a first order lag process is a PI controller with parameters which can be calculated from equations F-19 and F-18.

For the dead time case, it can be shown that all that is necessary to obtain a deadbeat response is to use M_{i-N-1} rather than M_{i-1} in the deadbeat algorithm (where N = integer dead time Θ/ST). However, this is not possible for the load change since future controller values must be predicted.

APPENDIX G

COMPLETE LISTING OF ALL RSX PROGRAMS

```
C       PROGRAM NAME--RST2
        SUBROUTINE RST(ITICKS)
C       SUBROUTINE TO SET RELAYS AND WAIT ITICKS
C       BEFORE RESUMING PROGRAM EXECUTION
        DIMENSION IT(2)
        IT(1)=ITICKS
        IT(2)=1
        CALL RLYST
        CALL MARK(IT,IEV)
        CALL WAITFR(IEV)
        RETURN
        END
```

```
PIP V13A

>T TT_DT0 STUPP SRC
```

```
C       TASK NAME STUPP(START UP PROGRAM)
        DIMENSION IT(4)
        COMMON/COMB/L(23)
C
C       TURN ON CLOCK USING DTCLK
        IT(1)=0
        IT(2)=?
        IT(3)=1
        IT(4)=1
        IP=1
        CALL FIX(5HDTCLK,IEV)
        CALL RUN(5HDTCLK,IT,IP,IEV)
C
C       START SCAN OF ADC CHANNELS
        CALL REQST(4HSCAN,0,IEV)
C
C       TURN ON ADC CHANNELS
        CALL ADCON(L(13),1,1,IX,IST)
        CALL ADCON(L(14),1,1,IX,IST)
        CALL ADCON(L(15),1,10,IX,IST)
C
C       SET RELAYS TO PROPER POSITION
        DO 10 I=1,12
10      CALL RLYOF(L(I),1,IEV)
        CALL RLYON(L(6),1,IEV)
        CALL RLYST
        CALL EXIT
        STOP
        END
```

```
C       TASK NAME RSTUP(RE-START UP PROGRAM)
C
        COMMON/COMA/NTICK,ITCR,
        1 JTSIG(2,21),IVS,ICS,INEC,ITK,ISEC,IVSA,ICSA,IDTK,
        1 IDSEC,JSUM(2),MSUM(2),NSUM(2),NTCCP,VO,CU,GRIN,K,
        1 SUP,SMASS,CP(2),TEMP(12),MM,ISPAN
        COMMON/COMB/L(23),ISPEC,IPCR,ITSIG(8),FE(4),FC(4),
        1 E(2,4),U(2,4),JOF(4),IRNG(4),MI(4),MA(4),M(2),C(4,4),
        1 GC(4),TI(4),TD(2),GL(2),LP(2),TP(2),IFC(4)
C
C       INITIALIZE VARIABLES FOR CAL04
        K = 0
        MM = 0
        ITK = 0
        ISEC = 0
        INEC = 0
        IVS = 0
        ICS = 0
C
C       INITIALIZE VARIABLES FOR DDCAL
        DO 5 I=1,4
        U(1,I)=MI(I)
        U(2,I)=MI(I)
        E(1,I)=0.
        E(2,I)=0.
        CALL DACST(L(I+15),1,MI(I),IEV)
5       CALL DACST(L(I+19),1,0,IEV)
C
C       THIS PORTION SETS RELAYS TO PROPER POSITIONS.
        DO 10 I=1,12
10      CALL RLYOF(L(I),1,IEV)
        CALL RLYON(L(6),1,IEV)
        CALL RLYST
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME--HTCAP
C       USED TO START A HEAT CAPACITY RUN
        COMMON/COMA/NTICK,ITCR,IBL(67),SMASS
        COMMON/COMB/L(23),ISPEC,IPCR
        DATA YES/1HY/
C
C       ENTER HEAT CAPACITY RUN DATA
100     WRITE(4,1000)
        READ(4,1100) SMASS
        WRITE(4,1200)
        READ(4,1100) OHM
        WRITE(4,1300)
        READ(4,1100) CAP
        WRITE(4,1400)
        READ(4,1100) HR
        ECAL=SQRT(OHM*SMASS*CAP*HR/60.)
        IF (ECAL.LE.9.8) GO TO 200
        ECAL =9.8
        HR=(ECAL**2*60.)/(OHM*SMASS*CAP)
        WRITE(4,1500) HR
200     WRITE(4,1600) SMASS,OHM,CAP,HR,ECAL
        READ(4,1700) ANS
        IF (ANS.NE.YES) GO TO 100
        WRITE(4,1800)
        READ(4,1700) ANS
        IF (ANS.NE.YES) GO TO 300
C
C       DIGITAL CONTROL SECTION
        WRITE(4,1900)
        IRATE=3000./HR
        ITCR=((IRATE/IPCR)+1)*IPCR
        WRITE(4,2000) ITCR
        CALL FIX(5HDDCAL,IEV)
        CALL FIX(5HCAL05,IEV)
        CALL RST(100)
        CALL REQST(5HDDCAL,0,IEV)
        CALL RST(100)
        CALL REQST(5HCAL05,0,IEV)
        GO TO 999
C
C       ANALOG CONTROL SECTION
300     WRITE(4,2100)
        IRATE=3000./HR
        ITCR=((IRATE/50)+1)*50
        WRITE(4,2000) ITCR
        CALL FIX(5HCAL04,IEV)
        CALL RST(100)
        CALL REQST(5HCAL04,0,IEV)
999     CALL REQST(5HCAL00,0,IEV)
1000    FORMAT(/1X,27H ENTER SAMPLE MASS IN GRAMS)
1100    FORMAT(F10.5)
1200    FORMAT(/1X,41H ENTER SPECIMEN HEATER RESISTANCE IN OHMS)
1300    FORMAT(/1X,51H ENTER HEAT CAPACITY AT 400 DEG C (J
        1OULES/GM-DEG K))
1400    FORMAT(/1X,42H ENTER DESIRED HEATING RATE (DEG C/
        1MINUTE))
1500    FORMAT(/1X,36H VOLTAGE GREATER THAN 10 VOLTS/
        1 25H SET VOLTAGE TO 9.8 VOLTS/
        1 20H NEW HEATING RATE = ,F10.2)
1600    FORMAT(/1X,8H MASS = ,F10.5/1X,14H RESISTANCE = ,F10.2
        1/1X,17H HEAT CAPACITY = ,F10.4/1X,16H HEATIN
        1G RATE = ,F10.2//1X,25H SET SPECIMEN VOLTAGE TO ,F10.2,
        1 8H   VOLTS//13H CHECK Y OR N)
1700    FORMAT(1A1)
1800    FORMAT(/1X,29H DO YOU WANT DIGITAL CONTROL?/1X,
        1 12H CHECK Y OR N)
1900    FORMAT(/1X,24H DIGITAL CONTROL SECTION)
2000    FORMAT(/1X,8H ITCR = ,I7)
2100    FORMAT(/1X,23H ANALOG CONTROL SECTION)
        CALL EXIT
        STOP
        END
```

```
C       TASK NAME - CAL00
C       TASK LISTS CURRENT VALUES OF VARIABLES IN COMMON
        COMMON/COMA/NTICK, ITCR,
        1 JTSIG(2,21), IVS, ICS, INEC, ITK, ISEC, IVSA, ICSA, IDTK,
        1 IDSEC, JSUM(2), MSUM(2), NSUM(2), NTCCP, VO, CU, GAIN, K,
        1 SUP, SMASS, CP(2), TEMP(12), MM, ISPAN
        COMMON/COMB/L(23), ISPEC, IPCR, ITSIG(8), FE(4), FC(4),
        1 F(2,4), U(2,4), JOF(4), IRNG(4), MI(4), MA(4), M(2), C(4,4),
        1 GC(4), TI(4), TD(2), GL(2), GP(2), TP(2), IFC(4)
        WRITE(4,1500)
        WRITE(4,2000)(L(I), I=1,8)
        WRITE(4,2500)
        WRITE(4,3000)(L(I), I=9,15)
        WRITE(4,3200)
        WRITE(4,3400)(L(I), I=16,23)
        WRITE(4,3500)SMASS, NTICK, IPCR
        WRITE(4,5000)ITCR, NTCCP, GAIN, SUP
        WRITE(4,5500)
        WRITE(4,6000)K, MM, ITK, ISEC, INEC, VO, CU
        WRITE(4,7500)
        DO 100 I=1,4
        OF=JOF(I)/(2.047E-4*GAIN)
100     WRITE(4,8000) I, OF, FC(I), MI(I), MA(I), IRNG(I)
1500    FORMAT(/1X40H L(1) L(2) L(3) L(4) L(5) L(6) L(7) L(8))
2000    FORMAT(8I5)
2500    FORMAT(/1X42H  L(9) L(10) L(11) L(12) L(13) L(14) L(15))
3000    FORMAT(7I6)
3200    FORMAT(/1X48H L(16) L(17) L(18) L(19) L(20) L(21) L(22) L(23))
3400    FORMAT(8I6)
3500    FORMAT(/1X13HSPEC.  MASS = ,F10.5,5X,7HNTICK= ,I5,5X,6HIPCR =,I5)
5000    FORMAT(/1X7HITCR = ,I5,5X,8HNTCCP = ,I5,3X,6HGAIN= ,F8.0,
        1    2X,6HSUP = ,F6.1)
5500    FORMAT(/1X50H  K        MM     ITK      ISEC     INEC     VO       CU   )
6000    FORMAT(/1X,I3,I8,I7,I9,I8,2X,F8.4,1X,F8.4)
7500    FORMAT(/1X,12HCONTROLLER #,3X,6HOFFSET,4X,5HALPHA,6X,
        1 2HMI,4X,2HMA,4X,4HIRNG)
8000    FORMAT(1X,I7,7X,F6.1,F10.4,I9,2I6)
        CALL EXIT
        STOP
        END



C       TASK NAME - CAL01
C       TASK SETS OPERATING VARIABLES
        DIMENSION V(7)
        COMMON/COMA/NTICK, ITCR,
        1 JTSIG(2,21), IVS, ICS, INEC, ITK, ISEC, IVSA, ICSA, IDTK,
        1 IDSEC, JSUM(2), MSUM(2), NSUM(2), NTCCP, VO, CU, GAIN, K,
        1 SUP, SMASS, CP(2), TEMP(12), MM, ISPAN
        COMMON/COMB/L(23)
        DATA YES/1HY/, V(1), V(2), V(3), V(4), V(5), V(6), V(7)
        1/5HLJOTE, 5HLJNTF, 5HLJSTT, 5HSMASS, 3HSUP, 4HGAIN
        1, 4HNONE/
10      WRITE(4,500)
        READ(4,1000)VAR
        DO 20 I = 1,7
        J = I
        IF(VAR.EQ.V(I)) GOTO 30
20      CONTINUE
        WRITE(4,1100)
        WRITE(4,1150)
        GOTO 10
30      GOTO(50,70,80,90,100,110,140), J
50      WRITE(4,1500)
        READ(4,2000)(L(I), I = 1,8)
        WRITE(4,2500)(L(I), I = 1,8)
65      READ(4,3000)ANS
        IF(ANS.NE.YES) GOTO(50,70,80,90,100,110), J
        GOTO 10
```

```
70      WRITE(4,3500)
        READ(4,4000)(L(I), I =9,15)
        WRITE(4,4500)(L(I), I =9,15)
        GOTO 65
80      WRITE(4,4600)
        READ(4,4700)(L(I), I=16,23)
        WRITE(4,4800)(L(I), I=16,23)
        GOTO 65
90      WRITE(4,5000)
        READ(4,5500)SMASS
        WRITE(4,6000)SMASS
        GOTO 65
100     WRITE(4,190)
        READ(4,200)SUP
        WRITE(4,250)SUP
        GOTO 65
110     WRITE(4,9000)
        READ(4,9500)GAIN
        WRITE(4,9600)GAIN
        ISPAN = (GAIN/10000.)*2047.
        GOTO 65
500     FORMAT(//1X4HVAR:)
190     FORMAT(/1X11H SUPRESSION)
200     FORMAT(F8.3)
250     FORMAT(/1X8H  SUP = ,F8.3,14H  CHECK Y OR N)
1000    FORMAT(1A5)
1100    FORMAT(/1X23HINCORRECT VARIABLE NAME)
1150    FORMAT(1X40HTRY AGAIN OR TYPE  NONE  TO EXIT PROGRAM)
1500    FORMAT(/1X40H L(1) L(2) L(3) L(4) L(5) L(6) L(7) L(8))
2000    FORMAT(8I5)
2500    FORMAT(1X,8I5,15H   CHECK Y OR N)
3000    FORMAT(1A1)
3500    FORMAT(/1X42H  L(9) L(10) L(11) L(12) L(13) L(14) L(15))
4000    FORMAT(7I6)
4500    FORMAT(1X,7I6,15H   CHECK Y OR N)
4600    FORMAT(/1X48H L(16) L(17) L(18) L(19) L(20) L(21) L(22) L(23))
4700    FORMAT(8I6)
4800    FORMAT(1X,8I6/1X,15H   CHECK Y OR N)
5000    FORMAT(/1X18HSPEC. MASS, GRAMS:)
5500    FORMAT(F10.5)
6000    FORMAT(1X7HMASS = , F10.5, 15H   CHECK Y OR N)
9000    FORMAT(//1X23HCALIBRATED SYSTEM GAIN:)
9500    FORMAT(F8.0)
9600    FORMAT(/1X8H GAIN = ,F8.0,15H   CHECK Y OR N)
140     CALL EXIT
        STOP
        END
```

```
C       CAAA1
C       TASK TO SET VARIABLES IN COMMON NOT SET BY CAL01
        DIMENSION V(7)
        COMMON/COMA/NTICK,ITCR,
        1 JTSIG(2,21),IVS,ICS,INFC,ITK,ISEC,IVSA,ICSA,IDTK,
        1 IDSEC,JSUM(2),MSUM(2),NSUM(2),NTCCP,VO,CU,GAIN,K,
        1 SUP,SMASS,CP(2),TEMP(12),MM,ISPAN
        COMMON/COMB/L(23),ISPEC,IPCR
        DATA YES/1HY/,V(1),V(2),V(3),V(4),V(5),V(6),V(7)
        1 /5HNTICK,1HK,5HNTCCP,4HITCR,4HIPCR,2HMM,4HNONE/
10      WRITE(4,150)
        READ(4,200) VAR
        DO 20 I=1,7
        J=I
        IF(VAR.EQ.V(I)) GO TO 30
20      CONTINUE
```

```
      WRITE(4,300)
      WRITE(4,400)
      GO TO 10
30    GO TO (50,60,70,80,90,100,999), J
65    READ(4,800) ANS
      IF(ANS .NE. YES) GO TO (50,60,70,80,90,100), J
      GO TO 10
50    WRITE(4,500)
      READ(4,600)NTICK
      WRITE(4,700) NTICK
      GO TO 65
60    WRITE(4,900)
      READ(4,1000)K
      WRITE(4,1100) K
      GO TO 65
70    WRITE(4,1200)
      READ(4,600) NTCCP
      WRITE(4,1400) NTCCP
      GO TO 65
80    WRITE(4,1500)
      READ(4,600) ITCR
      WRITE(4,1600)ITCR
      GO TO 65
90    WRITE(4,1700)
      READ(4,600) IPCR
      WRITE(4,1800) IPCR
      GO TO 65
100   WRITE(4,1900)
      READ(4,600) MM
      WRITE(4,2000) MM
      GO TO 65
150   FORMAT(//1X4HVAR:)
200   FORMAT(1A5)
300   FORMAT(/1X23HINCORRECT VARIABLE NAME)
400   FORMAT(1X38HTRY AGAIN OR TYPE NONE TO EXIT PROGRAM)
500   FORMAT(/1X19HNO. OF TICKS DELAY:)
600   FORMAT(I5)
700   FORMAT(1X7HNTICK= ,I5,5X,12HCHECK Y OR N)
800   FORMAT(1A1)
900   FORMAT(/1X30HUNITS OF 1000 MICROVOLTS SUPP:)
1000  FORMAT(I3)
1100  FORMAT(1X3HK= ,I3,5X,12HCHECK Y OR N)
1200  FORMAT(/1X47H NO. OF SETS OF TC READINGS BETWEEN SP HT CALC:)
1400  FORMAT(/1X9H NTCCP = ,I5,15H   CHECK Y OR N)
1500  FORMAT(/1X,46H TICKS BETWEEN SETS OF TC READINGS-XX00:)
1600  FORMAT(/1X8H ITCR = ,I5,15H   CHECK Y OR N)
1700  FORMAT(/1X40H NO. TICKS BETWEEN CONTROLLER READINGS :)
1800  FORMAT(/1X8H IPCR = ,I5,15H   CHECK Y OR N)
1900  FORMAT(/1X9H ENTER MM)
2000  FORMAT(/1X6H MM = ,I5,13H CHECK Y OR N)
999   CALL EXIT
      STOP
      END



C     PROGRAM NAME--GPFRC
C     TASK TO SET VARIABLES IN COMMON USED IN DIGITAL CONTROLLERS
      DIMENSION V(6)
      COMMON/COMA/NTICK,ITCR,
     1 JTSIG(2,21),IVS,ICS,INEC,ITK,ISEC,IVSA,ICSA,IDTK,
     1 IDSEC,JSUM(2),MSUM(2),NSUM(2),NTCCP,VO,CU,GAIN,K,
     1 SUP,SMASS,CP(2),TEMP(12),MM,ISPAN
      COMMON/COMB/L(23),ISPEC,IPCR,ITSIG(8),FE(4),FC(4),
     1 E(2,4),U(2,4),JOF(4),IRNG(4),MI(4),MA(4),M(2),C(4,4),
     1 GC(4),TI(4),TD(2),GL(2),GP(2),TP(2)
      DATA YES/1HY/,V(1),V(2),V(3),V(4),V(5),V(6)
     1 /2HOF,2HFC,2HMI,2HMA,4HIRNG,4HNONE/
```

```
10      WRITE(4,1000)
        READ(4,1100) N
        WRITE(4,1200) N
        READ(4,1300) ANS
        IF(ANS.NE.YES) GO TO 10
20      WRITE(4,1400)
        READ(4,1300) VAR
        DO 30 I=1,6
        J=I
        IF(VAR.EQ.V(I)) GO TO 40
30      CONTINUE
        WRITE(4,1500)
        GO TO 20
40      GO TO (60,70,80,90,100,999), J
50      READ(4,1300) ANS
        IF (ANS .NE. YES) GO TO (60,70,80,90,100), J
        WRITE(4,1550)
        READ(4,1300) ANS
        IF (ANS .EQ. YES) GO TO 10
        GO TO 20
60      WRITE(4,1600)
        READ(4,1700) OF
        WRITE(4,1800) OF
        JOF(N)=2.047E-4*GAIN*OF+0.5
        GO TO 50
70      WRITE(4,1900)
        READ(4,1700) FC(N)
        WRITE(4,2000) FC(N)
        GO TO 50
80      WRITE(4,2100)
        READ(4,1100) MI(N)
        WRITE(4,2200) MI(N)
        GO TO 50
90      WRITE(4,2300)
        READ(4,1100) MA(N)
        WRITE(4,2400) MA(N)
        GO TO 50
100     WRITE(4,2500)
        READ(4,1100) IRNG(N)
        WRITE(4,2600) IRNG(N)
        GO TO 50
1000    FORMAT(/1X,15H CONTROLLER # ?)
1100    FORMAT(I5)
1200    FORMAT(/1X,15H CONTROLLER # =,I5,13H CHECK Y OR N)
1300    FORMAT(A5)
1400    FORMAT(/1X,4HVAR:)
1500    FORMAT(/1X,24H INCORRECT VARIABLE NAME,//1X,
        1 39H TRY AGAIN OR TYPE NONE TO EXIT PROGRAM)
1550    FORMAT(/1X29H NEW CONTROLLER?  TYPE Y OR N)
1600    FORMAT(/1X38H ENTER CONTROLLER OFFSET IN MICROVOLTS)
1700    FORMAT(F10.0)
1800    FORMAT(1X,9H OFFSET =,F10.2,13H CHECK Y OR N)
1900    FORMAT(/1X,53H ENTER ALPHA FOR FIRST ORDER LAG USED I
        1N ERROR FILTER)
2000    FORMAT(1X,24H ERROR FILTER CONSTANT =,F10.5,13H CHECK Y OR N)
2100    FORMAT(/1X,45H ENTER MINIMUM CONTROLLER VALUE IN DAC COUNTS)
2200    FORMAT(1X,5H MI =,I5,13H CHECK Y OR N)
2300    FORMAT(/1X,45H,ENTER MAXIMUM CONTROLLER VALUE IN DAC COUNTS)
2400    FORMAT(1X,5H MA =,I5,13H CHECK Y OR N)
2500    FORMAT(/1X47H ENTER +/- RANGE FOR ERROR OUTPUT IN MICROVOLTS)
2600    FORMAT(/1X22H ERROR OUTPUT RANGE = ,I5,13H CHECK Y OR N)
999     CALL EXIT
        STOP
        END
```

```
C       FOCCO
C       TASK TO SET VARIABLES IN COMMON USED IN PID CONTROLLER
        DIMENSION V(4)
        COMMON/COMB/IBL(24),IPCR,JBL(74),C(4,4),
       1 GC(4),TI(4),TD(3)
        DATA YES/1HY/,V(1),V(2),V(3),V(4)
       1 /2HGC,2HTI,2HTD,4HNONE/
10      WRITE(4,1000)
        READ(4,1100) N
        WRITE(4,1200) N
        READ(4,1300) ANS
        IF(ANS.NE.YES) GO TO 10
        IF(N.GT.0 .AND. N.LE.2) GO TO 20
        WRITE(4,1350)
        GO TO 10
20      WRITE(4,1400)
        READ(4,1300) VAR
        DO 30 I=1,4
        J=I
        IF(VAR.EQ.V(I)) GO TO 40
30      CONTINUE
        WRITE(4,1500)
        GO TO 20
40      GO TO (60,70,80,999), J
50      READ(4,1300) ANS
        IF (ANS .NE. YES) GO TO (60,70,80), J
        WRITE(4,1550)
        READ(4,1300) ANS
        IF (ANS .EQ. YES) GO TO 10
        GO TO 20
60      WRITE(4,1600)
        READ(4,1700) GC(N)
        WRITE(4,1800) GC(N)
        GO TO 50
70      WRITE(4,1900)
        READ(4,1700) TI(N)
        WRITE(4,2000) TI(N)
        GO TO 90
80      WRITE(4,2100)
        READ(4,1700) TD(N)
        WRITE(4,2200) TD(N)
90      T=IPCR
        T=T/100.
        C(4,N)=TD(N)/T
        C(2,N)=1. +T/TI(N)+C(4,N)
        C(3,N)=1. +2.*C(4,N)
        GO TO 50
1000    FORMAT(/1X,15H CONTROLLER # ?)
1100    FORMAT(I5)
1200    FORMAT(/1X,15H CONTROLLER # =,I5,13H CHECK Y OR N)
1300    FORMAT(A5)
1350    FORMAT(/1X,51H THIS PROGRAM IS ONLY VALID FOR CON
        1TROLLERS 1 AND 2)
1400    FORMAT(/1X,4HVAR:)
1500    FORMAT(/1X,24H INCORRECT VARIABLE NAME,//1X,
       1 39H TRY AGAIN OR TYPE NONE TO EXIT PROGRAM)
1550    FORMAT(/1X29H NEW CONTROLLER?  TYPE Y OR N)
1600    FORMAT(/1X22H ENTER CONTROLLER GAIN)
1700    FORMAT(F10.0)
1800    FORMAT(/1X18H CONTROLLER GAIN =,F10.4,13H CHECK Y OR N)
1900    FORMAT(/1X,40H ENTER INTEGRAL CONSTANT-TI (IN SECONDS))
2000    FORMAT(1X5H TI =,F10.4,13H CHECK Y OR N)
2100    FORMAT(/1X,42H ENTER DERIVATIVE CONSTANT-TD (IN SECONDS))
2200    FORMAT(1X5H TD =,F10.4,13H CHECK Y OR N)
999     CALL EXIT
        STOP
        END
```

```
C       SOCCO
C       TASK TO SET VARIABLES IN COMMON USED IN PID CONTROLLER
        DIMENSION V(4)
        COMMON/COMB/IBL(24),IPCR,JBL(74),C(4,4),
        1 GC(4),TI(4),TD(2),GL(2),GP(2),TP(2)
        DATA YES/1HY/,V(1),V(2),V(3),V(4)
        1 /2HGC,2HTI,2HGL,4HNONE/
10      WRITE(4,1000)
        READ(4,1100) N
        WRITE(4,1200) N
        READ(4,1300) ANS
        IF(ANS.NE.YES) GO TO 10
        IF(N.GE.3 .AND. N.LE.4) GO TO 20
        WRITE(4,1350)
        GO TO 10
20      WRITE(4,1400)
        READ(4,1300) VAR
        DO 30 I=1,4
        J=I
        IF(VAR.EQ.V(I)) GO TO 40
30      CONTINUE
        WRITE(4,1500)
        GO TO 20
40      GO TO (60,70,80,999), J
50      READ(4,1300) ANS
        IF (ANS .NE. YES) GO TO (60,70,80), J
        WRITE(4,1550)
        READ(4,1300) ANS
        IF (ANS .EQ. YES) GO TO 10
        GO TO 20
60      WRITE(4,1600)
        READ(4,1700) GC(N)
        WRITE(4,1800) GC(N)
        GO TO 50
70      WRITE(4,1900)
        READ(4,1700) TI(N)
        WRITE(4,2000) TI(N)
        GO TO 110
80      WRITE(4,2100)
        READ(4,1700) GL(N-2)
        WRITE(4,2200) GL(N-2)
110     CALL REQST(5HCPCSO,0,IEV)
        GO TO 50
1000    FORMAT(/1X,15H CONTROLLER # ?)
1100    FORMAT(I5)
1200    FORMAT(/1X,15H CONTROLLER # =,I5,13H CHECK Y OR N)
1300    FORMAT(A5)
1350    FORMAT(/1X,51H THIS PROGRAM IS ONLY VALID FOR CONT
        1ROLLERS 3 AND 4)
1400    FORMAT(/1X,4HVAR:)
1500    FORMAT(/1X,24H INCORRECT VARIABLE NAME,//1X,
        1 39H TRY AGAIN OR TYPE NONE TO EXIT PROGRAM)
1550    FORMAT(/1X29H NEW CONTROLLER?  TYPE Y OR N)
1600    FORMAT(/1X22H ENTER CONTROLLER GAIN)
1700    FORMAT(F10.0)
1800    FORMAT(/1X18H CONTROLLER GAIN =,F10.4,13H CHECK Y OR N)
1900    FORMAT(/1X,46H ENTER INTEGRAL CONSTANT-TI (IN SECONDS))
2000    FORMAT(1X5H TI =,F10.4,13H CHECK Y OR N)
2100    FORMAT(/1X,16H ENTER LEAD GAIN)
2200    FORMAT(1X5H GL =,F10.4,13H CHECK Y OR N)
999     CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME-CURCV: LISTS CURRENT CONTROLLER VALUES
        COMMON/COMA/KBL(64),GAIN,K
        COMMON/COMB/L(23),ISPEC,IPCR,ITSIG(8),FE(4),FC(4),
       1 E(2,4),U(2,4),JOF(4),IRNG(4),MI(4),MA(4),M(2),C(4,4),
       1 GC(4),TI(4),TD(2),GL(2),GP(2),TP(2)
        WRITE(4,1000)
        DO 10 I=1,4
        OF=JOF(I)
        OF=OF/(2.047E-4*GAIN)
10      WRITE(4,1100) I,OF,FC(I),MI(I),MA(I),IRNG(I)
15      WRITE(4,1200) K
        DO 20 I=1,2
20      WRITE(4,1300) I,GC(I),TI(I),TD(I)
        DO 30 I=3,4
30      WRITE(4,1350) I,GC(I),TI(I),GL(I-2),GP(I-2),TP(I-2)
        WRITE(4,1400) K
        DO 40 I=1,4
40      WRITE(4,1500) I,(C(J,I),J=1,4)
C
1000    FORMAT(////1X,35H TABLE OF CURRENT CONTROLLER VALUES//
       1 15X,29H GENERAL CONTROLLER VARIABLES//1X,12HCONTROLLE
       1R #,3X,6HOFFSET,4X,5HALPHA,6X,2HMI,4X,2HMA,4X,4HIRNG/)
1100    FORMAT(1X,I7,7X,F6.1,F10.4,I9,2I6)
1200    FORMAT(//10X,42H CONTROLLER CONSTANTS FOR A SUPPRESSIO
       1N OF,I3,14H000 MICROVOLTS//1X,12HCONTROLLER #,4X,2HKC,
       18X,2HTI,8X,2HTD,7X,2HGL,9X,2HGP,8X,2HTP/)
1300    FORMAT(1X,I7,F13.2,2F10.2)
1350    FORMAT(1X,I7,F13.2,F10.2,6X,3HN/A,1X,F10.4,2F10.2)
1400    FORMAT(//10X,41H ALGORITHM CONSTANTS FOR A SUPPRESSIO
       1N OF,I3,14H000 MICROVOLTS//1X,12HCONTROLLER #,3X,2HC1,
       18X,2HC2,8X,2HC3,8X,2HC4/)
1500    FORMAT(1X,I7,F13.4,3F10.4)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME-ALLCV: LISTS CURRENT CONTROLLER VALUES
        COMMON/COMA/KRL(64),GAIN,K
        COMMON/COMB/L(23),ISPEC,IPCR,ITSIG(8),FE(4),FC(4),
       1 E(2,4),U(2,4),JOF(4),IRNG(4),MI(4),MA(4),M(2),C(4,4),
       1 GC(4),TI(4),TD(2),GL(2),GP(2),TP(2)
        DATA YES/3HYES/
        WRITE(4,900)
        READ(4,950) ANS
        IF (ANS.NE.YES) GO TO 9999
        WRITE(4,1000)
        DO 10 I=1,4
        OF=JOF(I)
        OF=OF/(2.047E-4*GAIN)
10      WRITE(4,1100) I,OF,FC(I),MI(I),MA(I),IRNG(I)
        DO 40 KK=1,11
        K=KK-1
        CALL RQST(5HAPUF0,0,IEV)
        CALL RQST(5HAPUS0,0,IEV)
        CALL RST(200)
15      WRITE(4,1200) K
        DO 20 I=1,2
20      WRITE(4,1300) I,GC(I),TI(I),TD(I)
        DO 30 I=3,4
30      WRITE(4,1350) I,GC(I),TI(I),GL(I-2),GP(I-2),TP(I-2)
        WRITE(4,1400) K
        DO 40 I=1,4
40      WRITE(4,1500) I,(C(J,I),J=1,4)
C
900     FORMAT(//10X,39HCAUTION: DO NOT USE THIS PROGRAM EXCEPT/
        110X,39HOFF-LINE.  THE VALUE OF K IS CHANGED AND/
        110X,36HTHIS WILL RESULT IN A MALFUNCTION OF/
        110X,32HANY PROGRAM USING THIS VARIABLE.//
        110X,21HTO CONTINUE TYPE YES!)
950     FORMAT(A5)
1000    FORMAT(////1X,36H COMPLETE TABLE OF CONTROLLER VALUES//
       1 15X,29H GENERAL CONTROLLER VARIABLES//1X,12HCONTROLLE
       1R #,3X,6HOFFSET,4X,5HALPHA,6X,2HMI,4X,2HMA,4X,4HIRNG/)
1100    FORMAT(1X,I7,7X,F6.1,F10.4,I9,2I6)
1200    FORMAT(//10X,42H CONTROLLER CONSTANTS FOR A SUPPRESSIO
       1N OF,I3,14H000 MICROVOLTS//1X,12HCONTROLLER #,4X,2HKC,
       18X,2HTI,8X,2HTD,7X,2HGL,9X,2HGP,8X,2HTP/)
1300    FORMAT(1X,I7,F13.2,2F10.2)
1350    FORMAT(1X,I7,F13.2,F10.2,6X,3HN/A,1X,F10.4,2F10.2)
1400    FORMAT(//10X,41H ALGORITHM CONSTANTS FOR A SUPPRESSIO
       1N OF,I3,14H000 MICROVOLTS//1X,12HCONTROLLER #,3X,2HC1,
       18X,2HC2,8X,2HC3,8X,2HC4/)
1500    FORMAT(1X,I7,F13.4,3F10.4)
9999    CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME--DDCAL
C       THIS PROGRAM IS USED TO TAKE THERMOCOUPLE DATA
C       AND USING THE SPECIMEN AS A SET POINT, CONTROL THE
C       OTHER THERMOCOUPLES USING A STANDARD DISCRETE PID
C       ALGORITHM FOR CONTROLLERS 1 (G3) AND 2 (G1).
C       IT USES A PI+LEAD-LAG ALGORITHM FOR CONTROLLERS
C       3 (CYLINDER) AND 4 (G2).  THESE SHIELDS APPLY TO
C       CALORIMETER #2 ONLY.
        DIMENSION IT(4)
        COMMON/COMB/L(23),ISPEC,IPCR,ITSIG(8),FE(4),FC(4),
       1 E(2,4),U(2,4),JOF(4),IRNG(4),MI(4),MA(4),M(2),C(4,4),
       1 GC(4)
C
C       CALL PROGRAM BACK IN AT NEXT SAMPLE INTERVAL
        IT(1)=IPCR
        IT(2)=1
        IT(3)=0
        IT(4)=1
        IP=0
        CALL RUN(5HDDCAL,IT,IP,IEV)
C
C       READ ALL THERMOCOUPLES
        DO 300 J=1,8
        CALL RLYON(L(J+4),1,IEV)
        DO 100 I=1,8
        IF (I.EQ.J) GO TO 100
        CALL RLYOF(L(I+4),1,IST)
100     CONTINUE
        CALL RST(14)
        JTO=0
        DO 200 I=1,3
        CALL ADCIN(L(13),ITC,IEV)
        CALL RST(1)
200     JTO=JTO+ITC
300     ITSIG(J)=JTO/3
        CALL RLYON(L(6),1,IST)
        CALL RLYOF(L(12),1,IST)
        CALL RST(1)
C
C       CALCULATE ERRORS,CALCULATE CONTROLLER OUTPUTS, AND OUTPUT
C       THESE VALUES ON APPROPRIATE DACS
        DO 400 I=1,4
        FE(I)=FC(I)*FE(I)+(1.-FC(I))*(FLOAT(ITSIG(I+2)-ITSIG(2)))
        EI=FLOAT(JOF(I))-FE(I)
        UI=C(1,I)*U(1,I)-(C(1,I)-1.)*U(2,I)
       1 +GC(I)*(C(2,I)*EI-C(3,I)*E(1,I)+C(4,I)*E(2,I))
        IF (UI.GT.FLOAT(MA(I))) UI=MA(I)
        IF (UI.LT.FLOAT(MI(I))) UI=MI(I)
        IU=UI+0.5
        CALL DACST(L(I+15),1,IU,IEV)
        E(2,I)=E(1,I)
        E(1,I)=EI
        U(2,I)=U(1,I)
400     U(1,I)=UI
C
C       READ SPECIMEN VOLTAGE AND CURRENT
        CALL ADCIN(L(14),M(1),IST)
        CALL ADCIN(L(15),M(2),IST)
C
C       SET SUPPRESSION IF NECESSARY
        ISPEC=ITSIG(2)
        IF (IABS(ISPEC).GE.1850) CALL REQST(5HSUPRS,0,IEV)
C
C       SET ERRORS ON VOLTMETERS
        CALL REQST(5HERROR,0,IEV)
C
C       WRITE ALL DATA USING PROGRAM STORE
        CALL RESUME(5HSTORE,IEV)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME-ERROR: USED TO DISPLAY CONTROLLER ERRORS
C       ON VOLTMETERS USING DAC OUTPUTS
        COMMON/COMA/IBL(64),GAIN
        COMMON/COMB/L(23),JBL(10),FE(4),KBL(44),IRNG(4)
        DO 10 I=1,4
        ISET=FE(I)*2.5E6/(GAIN*FLOAT(IRNG(I)))
10      CALL DACST(L(I+19),1,ISET,IEV)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME-APUFO: AUTOMATIC PARAMETER UPDATE-FIRST ORDER
        COMMON/COMA/KBL(66),K
        COMMON/COMB/IBL(24),IPCR,JBL(74),C(4,4),
        1 GC(4),TI(4),TD(2),GL(2),GP(2),TP(2)
        T=IPCR
        T=T/100.
        RK=K
        TD(1)=0.
        TD(2)=0.
        IF (K.GT.0) GO TO 10
        GC(1)=2.3
        TI(1)=25.0
        GC(2)=8.0
        TI(2)=100.0
        GO TO 100
10      IF (K.GT.1) GO TO 20
        GC(2)=7.0
        TI(2)=25.0
        GO TO 90
20      GC(2)=6.5
        TI(2)=10.0
90      GC(1)=2.2
        TI(1)=15.0
100     CALL REQST(5HCPCFO,0,IEV)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME-RFUSO: AUTOMATIC PARAMETER UPDATE-SECOND ORDER
        COMMON/COMA/KBL(66),K
        COMMON/COMB/IBL(24),IPCR,JBL(74),C(4,4),
        1 GC(4),TI(4),TD(2),GL(2),GP(2),TP(2)
        T=IPCR
        T=T/100.
        X=K
        X=1000.*X
        GC(3)=10.7
        TI(3)=20.0
        GL(1)=0.455
        X=X+6398.
        GP(1)=1.95E14/(7.03E12+X*X*X)
        TP(1)=8.031E14/(1.162E12+X*X*X)
        GC(4)=10.6
        TI(4)=20.0
        GL(2)=0.455
        X=X-1096.
        GP(2)=2.73E13/(1.52E12+X*X*X)
        TP(2)=1.765E14/(6.64E11+X*X*X)
        CALL RFQST(5HCPC30,0,IEV)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME-CPCF0: USED TO CONVERT CONTROLLER PARAMETERS
        COMMON/COMB/IBL(24),IPCR,JBL(74),C(4,4),
        1 GC(4),TI(4),TD(2),GL(2),GP(2),TP(2)
        T=IPCR
        T=T/100.
        DO 10 I=1,2
        C(1,I)=1.0
        C(4,I)=TD(I)/T
        C(2,I)=1.+T/TI(I)+C(4,I)
10      C(3,I)=1.+2.*C(4,I)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME-CPCS0: USED TO CONVERT CONTROLLER PARAMETERS
        COMMON/COMB/IBL(24),IPCR,JBL(74),C(4,4),
        1 GC(4),TI(4),TD(2),GL(2),GP(2),TP(2)
        T=IPCR
        T=T/100.
        DO 10 I=1,2
        N=I+2
        X=T/TP(I)
        A=1.-X+(X*X)/2.-(X*X*X)/6.+(X*X*X*X)/24.
        B=1./(1.+T/TI(N))
        C(1,N)=1.+A-GL(I)*GP(I)*(1.-A)
        C(2,N)=GL(I)/B
        C(3,N)=C(2,N)*(A+B)
10      C(4,N)=C(2,N)*A*B
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME-STORE  USED TO STORE DATA ON DECTAPE
        COMMON/COMA/IBL(66),K,JBL(32),MM
        COMMON/COMB/L(23),ISPEC,IFCR,ITSIG(8),FE(4),FC(4),
       1 F(2,4),U(2,4),JOF(4),IRNG(4),MI(4),MA(4),M(2)
C
C       NAME DATA FILE
        WRITE(4,10)
        READ(4,20) FNAM
        WRITE(4,30) FNAM
10      FORMAT(/1X,"ENTER DATA FILE NAME (5 CHARACTERS)")
20      FORMAT(A5)
30      FORMAT(/1X,"DATA FILE NAME =",A6)
C
C       OPEN DATA FILE FOR INPUT
        CALL ENTER(19,FNAM,3HSRC,IEV)
C
C       SET UP TO WRITE DATA EVERY TIME PID EXECUTES
100     CALL SUSPND
        IF (MM.EQ.1) GOTO 200
C
C       WRITE THERMOCOUPLE DATA
        WRITE(19,1000) (ITSIG(I),I=1,8),K
C
C       WRITE CONTROLLER OUTPUT DATA
        WRITE(19,1100) (U(1,I),I=1,4),M(1),M(2)
C
        GO TO 100
C
C       IF ALL DATA HAS BEEN TAKEN CLOSE FILE
200     KK=100
        WRITE(19,1000) (ITSIG(I),I=1,8),KK
        WRITE(19,1100) (U(1,I),I=1,4),M(1),M(2)
        CALL CLOSE(19,FNAM,3HSRC,IEV)
C
1000    FORMAT(1X,9I7)
1100    FORMAT(1X,4F7.1,2I7)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME--STEP
C       USED TO ISSUE A STEP INPUT TO ANY SPECIFIED CONTROLLER
        COMMON/COMA/IBL(99),MM
        COMMON/COMB/L(23),ISPEC,IPCR,ITSIG(8),FE(4),FC(4),
       1 E(2,4),U(2,4),JOF(4),IRNG(4),MI(4),MA(4),M(2),C(4,4),
       1 GC(4)
C
C       ENTER CONTROLLER NUMBER AND STEP SIZE
        WRITE(4,1000)
        READ(4,1100) NUM
        WRITE(4,1200) NUM
        WRITE(4,1300)
        READ(4,1100) IM
        WRITE(4,1400) IM
        DU=IM
C
C       ISSUE STEP
        CK=GC(NUM)
        C1=C(1,NUM)
        A=FC(NUM)
        GC(NUM)=0.
        C(1,NUM)=1.
        FC(NUM)=0.
        U(1,NUM)=U(1,NUM)+DU
50      WRITE(4,1500)
        READ(4,1100) IKEY
        IF(IKEY.EQ.0) GO TO 100
        GO TO 50
100     MM=1
        GC(NUM)=CK
        C(1,NUM)=C1
        FC(NUM)=A
        CALL RST(500)
        MM=0
1000    FORMAT(1X,"ENTER CONTROLLER #")
1100    FORMAT(I5)
1200    FORMAT(1X,"CONTROLLER # =",I3)
1300    FORMAT(1X,"ENTER STEP SIZE IN DAC COUNTS ")
1400    FORMAT(1X,"STEP SIZE = ",I5)
1500    FORMAT(1X,"ENTER 0 WHEN AT STEADY STATE ")
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME--RESP
C       USED TO CALCULATE THE ERRORS USED IN PID FOR
C       OFF LINE ANALYSIS
        DIMENSION IE(12)
        COMMON/COMB/IBK(25),ITSIG(8)
C
C       NAME FILE
        WRITE(4,10)
        READ(4,20) FNAM
        WRITE(4,30) FNAM
10      FORMAT(/1X,"ENTER DATA FILE NAME (<6 CHARACTERS)")
20      FORMAT(A5)
30      FORMAT(/1X,"DATA FILE NAME =",A6)
C
C       OPEN DATA FILE FOR OUTPUT
        CALL SEEK(19,FNAM,3HSRC,IEV)
C
C       SELECT PROPER THERMOCOUPLE, CONTROLLER STEP SIZE, AND
C       SAMPLE TIME
        WRITE(4,1000)
        READ(4,1100) NTC
        WRITE(4,1200) NTC
        WRITE(4,1300)
        READ(4,1100) MMM
        WRITE(4,1400) MMM
        WRITE(4,1500)
        READ(4,1100) IPCR
        WRITE(4,1600) IPCR
        WRITE(6,1700) FNAM
        WRITE(6,1800) MMM,IPCR
C
C       CALCULATE ERROR AND OUTPUT ON TELETYPE
        NDAT=0
50      DO 100 J=1,12
        READ(19,1100) (ITSIG(I),I=1,8),K
        READ(19,1150) DUM
        IF(K.GT.15) GO TO 900
100     IE(J)=ITSIG(NTC+2)-ITSIG(2)
        WRITE(6,1800) (IE(J),J=1,12)
        NDAT=NDAT+12
        GO TO 50
900     NDAT=NDAT+J
        WRITE(6,1800) (IE(JJ),JJ=1,J)
        WRITE(6,1800) NDAT
        CALL CLOSE(19,FNAM,3HSRC,IEV)
1000    FORMAT(1X,"ENTER CONTROLLER NUMBER")
1100    FORMAT(9I7)
1150    FORMAT(F7.0)
1200    FORMAT(1X,"CONTROLLER # =",I5)
1300    FORMAT(1X,"ENTER CONTROLLER STEP SIZE IN DAC COUNTS")
1400    FORMAT(1X,"STEP SIZE =",I5)
1500    FORMAT(1X,"ENTER SAMPLE TIME IN TICKS")
1600    FORMAT(1X,"SAMPLE TIME =",I5)
1700    FORMAT(1X,A6)
1800    FORMAT(1X,12I5)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME--CAL04--USED TO TAKE THERMOCOUPLE DATA EVERY
C       ITCR TICKS
C       THIS PROGRAM USES AN AVERAGE OF THREE READINGS FOR T/C DATA
        DIMENSION IT(4)
        COMMON/COMA/NTICK,ITCR,
       1 JTSIG(2,21),IVS,ICS,INEC,ITK,ISEC,IVSA,ICSA,IDTK,
       1 IDSEC,JSUM(2),MSUM(2),NSUM(2),NTCCP
        COMMON/COMB/L(23),ISPEC
C
C       SET UP NEXT DATA ACQUISITION
        IT(1)=ITCR
        IT(2)=1
        IT(3)=0
        IT(4)=1
        IP=0
        CALL RUN(5HCAL04,IT,IP,IEV)
C
C       READ NEW SET OF TC EMFS
50      DO 550 J = 1,2
        CALL RLYON(L(J + 4),1,IEV)
        DO 100 IN = 1,8
        IF(IN+4.EQ.J+4) GOTO 100
        CALL RLYOF(L(IN +4),1,IST)
100     CONTINUE
        CALL RST(NTICK)
        JTO=0
        DO 160 N=1,5
        CALL ADCIN(L(13),ITC,IST)
        CALL RST(2)
160     JTO=JTO+ITC
550     JTSIG(J,21) = JTO/5
        CALL ADCIN(L(14),IVO,IEV)
        CALL ADCIN(L(15),ICO,IEV)
        IVS=IVS+IVO
        ICS=ICS+ICO
C
C       READ DELTA TIME IF HAVE COLLECTED NTCCP NEW DATA POINTS
600     INEC=INEC + 1
        IF(INEC .GE. NTCCP) GO TO 625
        GO TO 650
625     CALL DTIM(ITK,ISEC,IDTK,IDSEC)
        CALL DTIMON(ITK,ISEC)
C
C       INCREMENT THERMOCOUPLE AND CONTROLLER TABLES BY 1
650     DO 700 J=1,2
        DO 700 I=1,20
700     JTSIG(J,I)=JTSIG(J,I+1)
C
C       WRITE DATA ON TELETYPE IF HAVE COLLECTED NTCCP NEW POINTS
        IF(INEC.GE.NTCCP) GOTO 850
        GOTO 170
850     IVSA=IVS
        ICSA=ICS
        CALL REQST(5HCAL00,0,IEV)
```

```
C
C       ALSO SET UP SUMS FOR LEAST SQUARES FIT
        DO 890 J=1,2
        MSUM(J)=0
        NSUM(J)=0
        JSUM(J)=0
        DO 890 I=1,20
        MSUM(J)=MSUM(J)+JTSIG(J,I)
        NSUM(J)=NSUM(J)+I*JTSIG(J,I)
        IF(IABS(NSUM(J)).GE.100000) GOTO 660
        GOTO 890
660     IC=NSUM(J)/10000
        NSUM(J)=NSUM(J)-IC*10000
        JSUM(J)=JSUM(J)+IC
890     CONTINUE
        CALL REQST(5HCAL06,0,IEV)
        CALL REQST(5HCAL09,0,IFV)
        IVS=0
        ICS=0
        INEC=0
C
C       SET SUPRESSION IF NECESSARY
170     ISPEC=JTSIG(2,21)
        IF(IABS(ISPEC).GE.1850) CALL REQST(5HSUPRS,0,IFV)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME-CAL05: USED TO TAKE THERMOCOUPLE DATA FROM
C       CONTROLLER PROGRAM AND USE IT TO CALCULATE THE SPECIFIC HEAT
C       THE TIME INTERVAL THAT IT RUNS ON (ITCR) MUST BE AN INTEGER
C       MULTIPLE OF THE CONTROLLER SAMPLING TIME (IPCR)
        DIMENSION IT(4)
        COMMON/COMA/NTICK,ITCR,
        1 JTSIG(2,21),IVS,ICS,INEC,ITK,ISEC,IVSA,ICSA,IDTK,
        1 IDSEC,JSUM(2),MSUM(2),NSUM(2),NTCCP
        COMMON/COMB/L(23),ISPEC,IPCR,ITSIG(8)
C
C       SET UP NEXT DATA ACQUISITION
        IT(1)=ITCR
        IT(2)=1
        IT(3)=0
        IT(4)=1
        IP=0
        CALL RUN(5HCAL05,IT,IP,IEV)
C
C       READ NEW SET OF TC EMFS
50      DO 550 J = 1,2
        JTSIG(1,21)=ITSIG(1)
550     JTSIG(2,21)=ITSIG(2)
        CALL ADCIN(L(14),IVO,IEV)
        CALL ADCIN(L(15),ICU,IEV)
        IVS=IVS+IVO
        ICS=ICS+ICU
C
C       READ DELTA TIME IF HAVE COLLECTED NTCCP NEW DATA POINTS
600     INEC=INEC + 1
        IF(INEC .GE. NTCCP) GO TO 625
        GO TO 650
625     CALL DTIM(ITK,ISEC,IDTK,IDSEC)
        CALL DTIMON(ITK,ISEC)
C
C       INCREMENT THERMOCOUPLE AND CONTROLLER TABLES BY 1
650     DO 700 J=1,2
        DO 700 I=1,20
700     JTSIG(J,I)=JTSIG(J,I+1)
C
C       WRITE DATA ON TELETYPE IF HAVE COLLECTED NTCCP NEW POINTS
        IF(INEC.GE.NTCCP) GOTO 850
        GOTO 170
850     IVSA=IVS
        ICSA=ICS
        CALL RFQST(5HCAL08,0,IEV)
C
C       ALSO SET UP SUMS FOR LEAST SQUARES FIT
        DO 890 J=1,2
        MSUM(J)=0
        NSUM(J)=0
        JSUM(J)=0
        DO 890 I=1,20
        MSUM(J)=MSUM(J)+JTSIG(J,I)
        NSUM(J)=NSUM(J)+I*JTSIG(J,I)
        IF(IABS(NSUM(J)).GE.100000) GOTO 660
        GOTO 890
660     IC=NSUM(J)/100000
        NSUM(J)=NSUM(J)-IC*100000
        JSUM(J)=JSUM(J)+IC
890     CONTINUE
        CALL RFQST(5HCAL06,0,IEV)
        CALL RFQST(5HCAL09,0,IEV)
        IVS=0
        ICS=0
        INEC=0
170     CALL EXIT
        STOP
        END
```

```
C       TASK NAME - CAL06
C       TASK TO CALCULATE SPECIFIC HEAT
        DIMENSION EXVAL(4),EMF(4),JT(121)
        COMMON/COMA/IBLK(49),IVSA,ICSA,IDTK,IDSEC,JSUM(2),MSUM(2),
        1 NSUM(2),NTCCP,VO,CU,GAIN,K,SUP,SMASS,CP(2),TEMP(12)
        DATA JT(1),JT(2),JT(3),JT(4),JT(5),JT(6),JT(7)/
        1 0,1796,3466,5039,6536,7972,9357/
        DATA JT(8),JT(9),JT(10),JT(11),JT(12),JT(13),JT(14)/
        1 10697,12060,13270,14510,15725,16917,18087/
        DATA JT(15),JT(16),JT(17),JT(18),JT(19),JT(20),JT(21)/
        1 19239,20373,21492,22597,23688,24768,25836/
        DATA JT(22),JT(23),JT(24),JT(25),JT(26),JT(27),JT(28),JT(29)/
        1 26894,27943,28982,30014,31038,32054,33064,34067/
        DATA JT(30),JT(31),JT(32),JT(33),JT(34),JT(35),JT(36),JT(37)/
        1 35064,36056,37042,38022,38998,39969,40935,41897/
        DATA JT(38),JT(39),JT(40),JT(41),JT(42),JT(43),JT(44),JT(45)/
        1 42854,43806,44755,45699,46639,47575,48507,49435/
        DATA JT(46),JT(47),JT(48),JT(49),JT(50),JT(51),JT(52),JT(53)/
        1 50359,51279,52195,53107,54016,54920,55821,56718/
        DATA JT(54),JT(55),JT(56),JT(57),JT(58),JT(59),JT(60),JT(61)/
        1 57612,58502,59388,60271,61150,62026,62899,63769/
        DATA JT(62),JT(63),JT(64),JT(65),JT(66),JT(67),JT(68),JT(69)/
        1 64635,65499,66360,67217,68072,68924,69774,70620/
        DATA JT(70),JT(71),JT(72),JT(73),JT(74),JT(75),JT(76),JT(77)/
        1 71464,72305,73144,73980,74813,75644,76473,77298/
        DATA JT(78),JT(79),JT(80),JT(81),JT(82),JT(83),JT(84),JT(85)/
        1 78121,78942,79760,80576,81389,82200,83008,83814/
        DATA JT(86),JT(87),JT(88),JT(89),JT(90),JT(91),JT(92),JT(93)/
        1 84617,85418,86217,87013,87807,88599,89388,90175/
        DATA JT(94),JT(95),JT(96),JT(97),JT(98),JT(99),JT(100)/
        1 90960,91743,92523,93301,94078,94852,95624/
        DATA JT(101),JT(102),JT(103),JT(104),JT(105),JT(106),JT(107)/
        1 96394,97162,97928,98692,99454,100214,100973/
        DATA JT(108),JT(109),JT(110),JT(111),JT(112),JT(113),JT(114)/
        1 101729,102484,103238,103989,104739,105487,106234/
        DATA JT(115),JT(116),JT(117),JT(118),JT(119),JT(120),JT(121)/
        1 106979,107723,108465,109206,109945,110683,111420/
C       SET UP LEAST SQUARES FIT TO CURRENT LAST 20 VALUES
        DO 30 J=1,2
30      EXVAL(J)=-.0142857*(FLOAT(NSUM(J))+10000.*FLOAT(JSUM(J)))
        1 +.2*FLOAT(MSUM(J))
        DO 50 J = 1,2
50      EXVAL(J+2)=.0142857*(FLOAT(NSUM(J))+10000.*FLOAT(JSUM(J)))
        1  -.1*FLOAT(MSUM(J))
        VO=FLOAT(IVSA/(NTCCP))/204.7
        CU=FLOAT(ICSA/(NTCCP))/2047.
        DO 100 I=1,4
        EMF(I) = EXVAL(I)*1.E7/(2047.*GAIN) + FLOAT(K)*1000.  + SUP
        LA=AINT(EMF(I)/100.)+1.
100     TEMP(I)=(FLOAT(JT(LA))+((EMF(I)-FLOAT((LA-1)*100))/100.)
        1 *FLOAT(JT(LA+1)-JT(LA)))/100.
C       CP(1,2) = SPECIFIC HEAT CALCULATED FROM TC 1,2
        CP(1) = .95*VO*CU*(FLOAT(100*IDSEC + IDTK)/100.)/
        1         (SMASS*(TEMP(3) - TEMP(1)))
        CP(2) = .95*VO*CU*(FLOAT(100*IDSEC + IDTK)/100.)/
        1         (SMASS*(TEMP(4) - TEMP(2)))
        CALL EXIT
        STOP
        END
```

```
C       TASK NAME_CAL08
C       TASK TO WRITE JTSIG(J,I) USED IN EMF CALCULATION.
        DIMENSION KTSIG(2,20),SFE(2)
        COMMON/COMA/NTICK,ITCR,
       1 JTSIG(2,21),IVS,ICS,INEC,ITK,ISEC,IVSA,ICSA,IDTK,
       1 IDSEC,JSUM(2),MSUM(2),NSUM(2),NTCCP,VO,CU,GAIN,K
        DO 100 I = 1,20
        KTSIG(1,I) = JTSIG(1,I)
100     KTSIG(2,I) = JTSIG(2,I)
        DO 250 J=1,2
        SOS=0.
        DO 200 I=1,20
200     SOS=(FLOAT(KTSIG(J,I))-(FLOAT(I)*.0015038-.0157895)
       1 *(FLOAT(NSUM(J))+10000.*FLOAT(JSUM(J)))
       1 -(.2157895-FLOAT(I)*.0157895)*FLOAT(MSUM(J)))**2+SOS
250     SFE(J)=SQRT(SOS/18.)
        WRITE(6,1000) K,IVSA,ICSA,IDTK,IDSEC
        DO 300 I=1,2
        WRITE(6,1000) (KTSIG(I,J),J=1,10)
300     WRITE(6,1000) (KTSIG(I,J),J=11,20)
        WRITE(8,3000)(SFE(J),J=1,2)
1000    FORMAT(1X,10I6)
3000    FORMAT(/1X,10H SIGMA 1 =,F10.2,3X,10H SIGMA 2 =,F10.2)
        CALL EXIT
        STOP
        END
```

```
C       TASK NAME - CAL09
C       TASK LISTS CURRENT VALUES OF VARIABLES IN COMMON
        COMMON/COMA/NTICK,ITCR,
       1 JTSIG(2,21),IVS,ICS,INEC,ITK,ISEC,IVSA,ICSA,IDTK,
       1 IDSEC,JSUM(2),MSUM(2),NSUM(2),NTCCP,VO,CU,GAIN,K,
       1 SUP,SMASS,CP(2),TEMP(12),MM,ISPAN
        T1AVG=(TEMP(1)+TEMP(3))/2.
        T2AVG=(TEMP(2)+TEMP(4))/2.
        DT1=TEMP(3)-TEMP(1)
        DT2=TEMP(4)-TEMP(2)
        WRITE(8,1000)
        WRITE(8,2000)K,MM,ITK,ISEC,INEC,VO,CU
        WRITE(8,3000)
        WRITE(8,4000) (TEMP(I),I=1,4)
        WRITE(8,5000)
        WRITE(8,6000) T1AVG,T2AVG,DT1,DT2
        WRITE(8,7000) (CP(I),I=1,2)
1000    FORMAT(/1X50H   K        MM     ITK      ISEC     INEC     VO       CU   )
2000    FORMAT(/1X,I3,I8,I7,I9,I8,2X,F8.4,1X,F8.4)
3000    FORMAT(/42H TEMP(1,1)  TEMP(2,1)  TEMP(1,20)  TEMP(2,20))
4000    FORMAT(2F10.2,2F11.2)
5000    FORMAT(/36H     T1AVG     T2AVG       DT1       DT2)
6000    FORMAT(4F9.2)
7000    FORMAT(/1X,11HSP.  HT.  1 =,F8.4,5X,11HSP.  HT.  2 =,F8.4///)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME--RADFR
C       DELTA FILTER--USED TO FILTER THE DIFFERENCES
C       BETWEEN T/C MEASUREMENTS FOR SPECIFIC HEAT CALCULATIONS
        DIMENSION KTSIG(2,21),DY(2),DLY(2),DFY(2),TSIG(2)
        COMMON/COMA/NTICK,ITCR,
       1 JTSIG(2,21),IVS,ICS,INEC,ITK,ISEC,IVSA,ICSA,IDTK,
       1 IDSEC,JSUM(2),MSUM(2),NSUM(2),NTCCP,VO,CU,GAIN,K,
       1 SUP,SMASS,CP(2),TEMP(12),MM,ISPAN
C
C       ENTER DATA FILE NAME
        WRITE(4,1000)
        READ(4,1100) FNAM
        WRITE(4,1200) FNAM
        CALL SFFK(19,FNAM,3HSRC,IEV)
C
C       ENTER FILTER CONSTANT
        WRITE(4,1210)
        READ(4,1220) INST
        IF(INST.NE.0) GO TO 10
        ALPHA=0.
        GO TO 20
10      ALPHA=EXP(-1./FLOAT(INST))
20      WRITE(4,1230) ALPHA
C
C       INITIALIZE STARTING POINTS
        WRITE(4,1240)
        READ(4,1220) IFC
        WRITE(4,1250) IFC
        READ(19,1300) K,IVSA,ICSA,IDTK,IDSEC
        DO 50 I=1,2
        DLY(I)=IFC
        READ(19,1300) (KTSIG(I,J),J=2,11)
        READ(19,1300) (KTSIG(I,J),J=12,21)
        KTSIG(I,1)=KTSIG(I,2)-IFC
50      TSIG(I)=KTSIG(I,2)-IFC
        GO TO 250
C
C       ENTER DATA
1       READ(19,1300) KK,IVSA,ICSA,IDTK,IDSEC
        IF(KK.GT.15) GO TO 600
        IF(KK.EQ.K) GO TO 100
        KTSIG(1,1)=KTSIG(1,1)-ISPAN
        KTSIG(2,1)=KTSIG(2,1)-ISPAN
        TSIG(1)=TSIG(1)-ISPAN
        TSIG(2)=TSIG(2)-ISPAN
100     K=KK
        DO 200 I=1,2
        READ(19,1300) (KTSIG(I,J),J=2,11)
200     READ(19,1300) (KTSIG(I,J),J=12,21)
250     DO 300 I=1,2
        DO 300 J=1,20
        DY(I)=KTSIG(I,J+1)-KTSIG(I,J)
        DFY(I)=ALPHA*DLY(I)+(1.-ALPHA)*DY(I)
        TSIG(I)=TSIG(I)+DFY(I)
        JTSIG(I,J+1)=TSIG(I)
300     DLY(I)=DFY(I)
C
C       INCREMENT T/C TABLES BY ONE POSITION FOR COMPATABILITY
        DO 400 I=1,2
        DO 400 J=1,20
400     JTSIG(I,J)=JTSIG(I,J+1)
```

```
C
C       SET UP LEAST SQUARE SUMS
        DO 500 I=1,2
        MSUM(I)=0
        NSUM(I)=0
        JSUM(I)=0
        DO 500 J=1,20
        MSUM(I)=MSUM(I)+JTSIG(I,J)
        NSUM(I)=NSUM(I)+J*JTSIG(I,J)
        IF(IABS(NSUM(I)).GE.10000) GO TO 450
        GO TO 500
450     IC=NSUM(I)/10000
        NSUM(I)=NSUM(I)-IC*10000
        JSUM(I)=JSUM(I)+IC
500     CONTINUE
C
C       DEVELOP CONTINUITY WITH CAA08,CAL06,CAL09 AND CAA09
C       THEN WAIT FOR THEIR EXECUTION
        CALL REQST(5HCAA08,0,IEV)
        CALL REQST(5HCAL06,0,IEV)
        CALL RST(200)
        CALL REQST(5HCAL09,0,IEV)
        CALL REQST(5HCAA09,0,IEV)
        CALL RST(4000)
C
C       SET UP INITIAL POINT FOR NEXT PASS THROUGH
        KTSIG(1,1)=KTSIG(1,21)
        KTSIG(2,1)=KTSIG(2,21)
        GO TO 1
C
C       CLOSE DATA FILE
600     CALL CLOSE(19,FNAM,3HSRC,IEV)
1000    FORMAT(/1X,21H ENTER DATA FILE NAME)
1100    FORMAT(A5)
1200    FORMAT(/1X,12H FILE NAME =,A6)
1210    FORMAT(//1X,51H ENTER FILTER TIME CONSTANT (INT. # OF SAMP.
        1 TIMES))
1220    FORMAT(I4)
1230    FORMAT(/1X,8H ALPHA =,F10.5)
1240    FORMAT(/1X,55H INITIAL CONDITION FOR FILTER (DELTA
        1 VALUE--IN COUNTS):)
1250    FORMAT(/1X,8H I.  C.  =,I6)
1300    FORMAT(1016)
        CALL EXIT
        STOP
        END
```

```
C       TASK NAME_CAA08
C       TASK TO WRITE JTSIG(J,I) USED IN EMF CALCULATION.
        DIMENSION SEE(2)
        COMMON/COMA/NTICK,ITCR,
        1 JTSIG(2,21),IVS,ICS,INFO,ITK,ISEC,IVSA,ICSA,IDTK,
        1 IDSEC,JSUM(2),MSUM(2),NSUM(2),NTCCP,VO,CU,GAIN,K,
        1 SUP,SMASS,CP(2),TEMP(12),MM,ISPAN
        DO 250 J=1,2
        SOS=0.
        DO 200 I=1,20
200     SOS=(FLOAT(JTSIG(J,I))-(FLOAT(I)*.0015038-.0157895)
        1 *(FLOAT(NSUM(J))+10000.*FLOAT(JSUM(J)))
        1 -(.2157895-FLOAT(I)*.0157895)*FLOAT(MSUM(J)))**2+SOS
250     SEE(J)=SQRT(SOS/18.)
        WRITE(4,1000) (JTSIG(1,I),I=1,10)
        WRITE(4,1000) (JTSIG(2,I),I=1,10)
        WRITE(4,2000) (JTSIG(1,I),I=11,20)
        WRITE(4,1000) (JTSIG(2,I),I=11,20)
        WRITE(4,3000) (SEE(J),J=1,2)
1000    FORMAT(1X,10(I6))
2000    FORMAT(/1X,10(I6))
3000    FORMAT(/1X,10H SIGMA 1 =,F10.2,3X,10H SIGMA 2 =,F10.2)
        CALL EXIT
        STOP
        END
```

```
C       TASK NAME - CAA09
C       THIS TASK CORRECTS SPECIFIC HEAT DATA FOR HEATER
C       MASS AND DRIFT
C       NOTE:  EQUATIONS 10, 20, AND 50 MUST BE CHANGED
C       FOR EACH RUN
        DIMENSION TA(2)
        COMMON/COMA/NTICK,ITCR,
        1 JTSIG(2,21),IVS,ICS,INFO,ITK,ISEC,IVSA,ICSA,IDTK,
        1 IDSEC,JSUM(2),MSUM(2),NSUM(2),NTCCP,VO,CU,GAIN,K,
        1 SUP,SMASS,CP(2),TEMP(12),MM,ISPAN
        TA(1)=(TEMP(1)+TEMP(3))/2.+273.16
        TA(2)=(TEMP(2)+TEMP(4))/2.+273.16
        G=VO*CU/SMASS
10      IF (TA(1).LT.700) GO TO 40
        DO 30 I=1,2
20      R=0.
30      CP(I)=CP(I)/(1-R*CP(I)/(G*3600))
40      DO 50 I=1,2
50      CP(I)=CP(I)+0.*(CP(I)-0.4729)
        WRITE(4,1000) (CP(I),I=1,2)
        WRITE(6,1100) (TA(I),CP(I),I=1,2)
1000    FORMAT(//10X,24H CORRECTED SPECIFIC HEAT//1X,11HSP. HT.
        1 1 =,F8.4,5X,11HSP. HT. 2 =,F8.4///)
1100    FORMAT(1X,2(F10.2,F10.4))
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM--CAAJ0
C       USED TO TAKE THERMOCOUPLE DATA EVERY SPECIFIED INTERVAL
C       USING RUN COMMAND (MAINLY FOR DRIFT)
        DIMENSION JT(121),EXVAL(8)
        COMMON/COMA/NTICK,ITCR,
        1 JTSIG(2,21),IVS,IOS,INEC,ITK,ISEC,IVSA,ICSA,IDTK,
        1 IDSEC,JSUM(2),MSUM(2),NSUM(2),NTCCP,VO,CU,GAIN,K,
        1 SUP,SMASS,CP(2),TEMP(12),MM,ISPAN
        COMMON/COMB/L(23),ISPEC
        DATA JT(1),JT(2),JT(3),JT(4),JT(5),JT(6),JT(7)/
        1 0,1796,3466,5039,6536,7972,9357/
        DATA JT(8),JT(9),JT(10),JT(11),JT(12),JT(13),JT(14)/
        1 10697,12000,13270,14510,15725,16917,18087/
        DATA JT(15),JT(16),JT(17),JT(18),JT(19),JT(20),JT(21)/
        1 19239,20373,21492,22597,23688,24768,25836/
        DATA JT(22),JT(23),JT(24),JT(25),JT(26),JT(27),JT(28),JT(29)/
        1 26894,27943,28982,30014,31038,32054,33064,34067/
        DATA JT(30),JT(31),JT(32),JT(33),JT(34),JT(35),JT(36),JT(37)/
        1 35064,36056,37042,38022,38998,39969,40935,41897/
        DATA JT(38),JT(39),JT(40),JT(41),JT(42),JT(43),JT(44),JT(45)/
        1 42854,43806,44755,45699,46639,47575,48507,49435/
        DATA JT(46),JT(47),JT(48),JT(49),JT(50),JT(51),JT(52),JT(53)/
        1 50359,51279,52195,53107,54016,54920,55821,56718/
        DATA JT(54),JT(55),JT(56),JT(57),JT(58),JT(59),JT(60),JT(61)/
        1 57612,58502,59388,60271,61150,62026,62899,63769/
        DATA JT(62),JT(63),JT(64),JT(65),JT(66),JT(67),JT(68),JT(69)/
        1 64635,65499,66360,67217,68072,68924,69774,70620/
        DATA JT(70),JT(71),JT(72),JT(73),JT(74),JT(75),JT(76),JT(77)/
        1 71464,72305,73144,73980,74813,75644,76473,77298/
        DATA JT(78),JT(79),JT(80),JT(81),JT(82),JT(83),JT(84),JT(85)/
        1 78121,78942,79760,80576,81389,82200,83008,83814/
        DATA JT(86),JT(87),JT(88),JT(89),JT(90),JT(91),JT(92),JT(93)/
        1 84617,85418,86217,87013,87807,88599,89388,90175/
        DATA JT(94),JT(95),JT(96),JT(97),JT(98),JT(99),JT(100)/
        1 90960,91743,92523,93301,94078,94852,95624/
        DATA JT(101),JT(102),JT(103),JT(104),JT(105),JT(106),JT(107)/
        1 96394,97162,97928,98692,99454,100214,100973/
        DATA JT(108),JT(109),JT(110),JT(111),JT(112),JT(113),JT(114)/
        1 101729,102484,103238,103989,104739,105487,106234/
        DATA JT(115),JT(116),JT(117),JT(118),JT(119),JT(120),JT(121)/
        1 106979,107723,108465,109206,109945,110683,111420/
        DO 200 J=1,8
        CALL RLYON(L(J+4),1,IST)
        DO 130 N=1,8
        IF (N.EQ.J) GO TO 130
        CALL RLYOF(L(N+4),1,IST)
130     CONTINUE
        CALL RST(50)
        JTC=0
        DO 150 N=1,3
        CALL ADCIN(L(J3),ITC,IST)
        CALL RST(1)
150     JTC=JTC+ITC
200     EXVAL(J)=JTC/3
        CALL RLYON(L(6),1,IST)
        CALL RLYOF(L(12),1,IST)
        CALL RST(1)
        ISPEC=EXVAL(2)
        IF (IABS(ISPEC).GE.1850) CALL REQST(5HSUPRS,0,IEV)
        DO 500 J=1,8
        EMF=EXVAL(J)*1.0E7/(2047.0*GAIN)+FLOAT(K)*1000.0+SUP
        LA=AINT(EMF/100.0)+1.0
500     TEMP(J+4)=(FLOAT(JT(LA))+((EMF-FLOAT((LA-1)*100))/100.0)
        1 *FLOAT(JT(LA+1)-JT(LA)))/100.0
        CALL REQST(5HWRA10,0,IEV)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM--CAB10
C       USED TO TAKE THERMOCOUPLE VALUES GENERATED BY CONTROLLER
C       PROGRAM AND PRINT THE CURRENT TEMPERATURE
        DIMENSION JT(121)
        COMMON/COMA/NTICK,ITCR,
        1 JTSIG(2,21),IVS,ICS,INEC,ITK,ISEC,IVSA,ICSA,IDTK,
        1 IDSEC,JSUM(2),MSUM(2),NSUM(2),NTCCP,VO,CU,GAIN,K,
        1 SUP,SMASS,CP(2),TEMP(12),MM,ISPAN
        COMMON/COMA/L(23),ISPEC,IPCR,ITSIG(8)
        DATA JT(1),JT(2),JT(3),JT(4),JT(5),JT(6),JT(7)/
        1 0,1796,3466,5039,6536,7972,9357/
        DATA JT(8),JT(9),JT(10),JT(11),JT(12),JT(13),JT(14)/
        1 10697,12000,13270,14510,15725,16917,18087/
        DATA JT(15),JT(16),JT(17),JT(18),JT(19),JT(20),JT(21)/
        1 19239,20373,21492,22597,23688,24768,25836/
        DATA JT(22),JT(23),JT(24),JT(25),JT(26),JT(27),JT(28),JT(29)/
        1 26894,27943,28982,30014,31038,32054,33064,34067/
        DATA JT(30),JT(31),JT(32),JT(33),JT(34),JT(35),JT(36),JT(37)/
        1 35064,36056,37042,38022,38998,39969,40935,41897/
        DATA JT(38),JT(39),JT(40),JT(41),JT(42),JT(43),JT(44),JT(45)/
        1 42854,43806,44755,45699,46639,47575,48507,49435/
        DATA JT(46),JT(47),JT(48),JT(49),JT(50),JT(51),JT(52),JT(53)/
        1 50359,51279,52195,53107,54016,54920,55821,56718/
        DATA JT(54),JT(55),JT(56),JT(57),JT(58),JT(59),JT(60),JT(61)/
        1 57612,58502,59388,60271,61150,62026,62899,63769/
        DATA JT(62),JT(63),JT(64),JT(65),JT(66),JT(67),JT(68),JT(69)/
        1 64635,65499,66360,67217,68072,68924,69774,70620/
        DATA JT(70),JT(71),JT(72),JT(73),JT(74),JT(75),JT(76),JT(77)/
        1 71464,72305,73144,73980,74813,75644,76473,77298/
        DATA JT(78),JT(79),JT(80),JT(81),JT(82),JT(83),JT(84),JT(85)/
        1 78121,78942,79760,80576,81389,82200,83008,83814/
        DATA JT(86),JT(87),JT(88),JT(89),JT(90),JT(91),JT(92),JT(93)/
        1 84617,85418,86217,87013,87807,88599,89388,90175/
        DATA JT(94),JT(95),JT(96),JT(97),JT(98),JT(99),JT(100)/
        1 90960,91743,92523,93301,94078,94852,95624/
        DATA JT(101),JT(102),JT(103),JT(104),JT(105),JT(106),JT(107)/
        1 96394,97162,97928,98692,99454,100214,100973/
        DATA JT(108),JT(109),JT(110),JT(111),JT(112),JT(113),JT(114)/
        1 101729,102484,103238,103989,104739,105487,106234/
        DATA JT(115),JT(116),JT(117),JT(118),JT(119),JT(120),JT(121)/
        1 106979,107723,108465,109206,109945,110683,111420/
        DO 500 J=1,8
        EMF=FLOAT(ITSIG(J))*1.0E7/(2047.0*GAIN)+FLOAT(K)*1000.0+SUP
        LA=AINT(EMF/100.0)+1.0
500     TEMP(J+4)=(FLOAT(JT(LA))+((EMF-FLOAT((LA-1)*100))/100.0)
        1 *FLOAT(JT(LA+1)-JT(LA)))/100.0
        CALL REQST(5HWRA10,0,IEV)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM CALLED WRA10 TO BE USED IN JUNCTION WITH
C       PROGRAM CAR10 AND CAB10  TO OUTPUT RESULTS.
        COMMON/COMA/IBL(75),TEMP(12)
        WRITE(4,1000) (TEMP(I),I=5,12)
1000    FORMAT(1X,8F9.2)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM--CAL10
C       USED TO MEASURE THE TIME REQUIRED FOR THE ANALOG
C       CIRCUIT TO CHARGE--FUNCTION OF CAPACITOR SETTING AND NTICK
        DIMENSION ITC(8),EMF(8)
        COMMON/COMA/NTICK,IBL(63),GAIN,K,SUP
        COMMON/COMB/L(23),ISPEC
C
C       READ THERMOCOUPLES
C
        DO 200 J=5,12
        CALL RLYON(L(J),1,IST)
        DO 130 N=5,12
        IF (N.EQ.J) GO TO 130
        CALL RLYOF(L(N),1,IST)
130     CONTINUE
        CALL RST(NTICK)
        CALL ADCIN(L(13),ITC(J-4),IST)
        CALL RST(50)
200     CONTINUE
        CALL RLYON(L(6),1,IST)
        CALL RLYOF(L(12),1,IST)
        CALL RST(1)
C
C       SET SUPRESSION IF NECESSARY
C
        ISPEC=ITC(2)
        IF (IABS(ISPEC).GE.1850) CALL REQST(SHSUPRS,0,IEV)
C
C       CALCULATE EMF VALUES AND WRITE ON TELETYPE
C
        DO 500 J=1,8
500     EMF(J)=FLOAT(ITC(J))*1.0E7/(2047.0*GAIN)+FLOAT(K)*1000.0
        1 +SUP
        WRITE(4,1100) (EMF(I),I=1,8)
1100    FORMAT(1X,8F9.2)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME--SUPRS
C       USED TO SET SUPRESSION ON THERMOCOUPLES
        COMMON/COMA/NTICK,ITCR,JTSIG(2,21),IBL(22),
       1 K,JBL(33),ISPAN
        COMMON/COMB/L(23),ISPEC
170     IF(ISPEC .GE. 1850) GO TO 300
        IF(ISPEC .LE. -1850) GO TO 370
        GO TO 900
300     K=K+1
        DO 330 JJ=1,2
        DO 330 II=1,21
330     JTSIG(JJ,II)=JTSIG(JJ,II)-ISPAN
        GO TO 380
370     K=K-1
        DO 375 JJ=1,2
        DO 375 II=1,21
375     JTSIG(JJ,II)=JTSIG(JJ,II)+ISPAN
380     DO 400 N = 1,4
        CALL RLYOF(L(N),1,IST)
400     CONTINUE
        MK=K
        DO 500 N=1,4
        IF(2*(MK/2) .NE. MK) CALL RLYON(L(N),1,IST)
500     MK=MK/2
        CALL RST(30)
        CALL RLYON(L(6),IST,IEV)
        DO 600 N=1,8
        IF(N.EQ.2) GOTO 600
        CALL RLYOF(L(N+4),IST,IEV)
600     CONTINUE
        CALL RST(25)
        JTO=0
        DO 165 N=1,3
        CALL ADCIN(L(13),ITC,IST)
        CALL RST(1)
165     JTO=JTO+ITC
        ISPEC=JTO/3
        GO TO 170
900     CALL REQST(5HAPUFO,0,IEV)
        CALL REQST(5HAPUSO,0,IEV)
        CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME-CALIB: USED TO CALIBRATE SYSTEM GAIN
C       AND SUPPRESSION (FIXED AND VARIABLE)
        DIMENSION ICOUNT(20),F(20)
        COMMON/COMR/L(23)
        DATA YES/1HY/
10      WRITE(4,1000)
        READ(4,1100) ICHAN
        WRITE(4,1200) ICHAN
        READ(4,1300) ANS
        IF (ANS.NE.YES) GO TO 10
        DO 20 I=1,8
20      CALL RLYOF(L(I+4),1,IEV)
        CALL RLYON(L(ICHAN+4),1,IEV)
        CALL RST(10)
        N=1.
        SX=0.
        SXX=0.
        SY=0.
        SXY=0.
        WRITE(4,1400)
30      READ(4,1500) E(N)
        WRITE(4,1600) F(N)
        READ(4,1300) ANS
        IF (ANS.NE.YES) GO TO 60
        IF (N.EQ.1) GO TO 120
        KN=F(N)/1000.
        IF (KN.EQ.KT) GO TO 40
        WRITE(4,1650)
        GO TO 60
40      JTC=0
        DO 45 I=1,5
        CALL ADCIN(L(13),ITC,IST)
        CALL RST(1)
45      JTC=JTC+ITC
        ICOUNT(N)=JTC/5
        SX=SX+FLOAT(ICOUNT(N))
        SXX=SXX+FLOAT(ICOUNT(N))**2
        SY=SY+F(N)
        SXY=SXY+FLOAT(ICOUNT(N))*F(N)
        IF (N.EQ.20) GO TO 80
        WRITE(4,1700)
        READ(4,1300) ANS
        IF (ANS.NE.YES) GO TO 70
50      N=N+1
60      WRITE(4,1800)
        GO TO 30
70      IF (N.GT.2) GO TO 80
        WRITE(4,1900)
        GO TO 50
80      DEN=N*SXX-SX**2
        A0=(SXX*SY-SX*SXY)/DEN
        A1=(N*SXY-SX*SY)/DEN
        WRITE(4,2100)
        SOS=0.
        DO 90 I=1,N
        EE=A0+A1*FLOAT(ICOUNT(I))
        SOS=SOS+(EE-E(I))**2
90      WRITE(4,2200) I,ICOUNT(I),E(I),EE
        SEE=SQRT(SOS/FLOAT(N-2))
        G=1./(2.047E-4*A1)
        IF (KT.GT.0) GO TO 100
        SF=A0
        SV=0.
        GO TO 110
```

```
100     WRITE(4,2300)
        READ(4,1500) SF
        SV=A0-SF
110     WRITE(4,2400) A0,A1,SFE,G,SF,SV
        WRITE(4,2500)
        READ(4,1300) ANS
        IF (ANS.EQ.YES) GO TO 10
        GO TO 5000
120     KT=F(1)/1000.
        DO 130 I=1,4
130     CALL RLYOF(L(I),1,IEV)
        MK=KT
        DO 140 I=1,4
        IF ((2*(MK/2)).NE.MK) CALL RLYON(L(I),1,IEV)
140     MK=MK/2
        CALL RST(200)
        GO TO 40
1000    FORMAT(/1X,45H CONNECT DAUPHINEE TO ANY MULTIPLEX
        1ER CHANNEL//1X,41H CONNECT + EMF TO BLACK THERMOCOUPL
        1E LEAD/1X,41H CONNECT - EMF TO WHITE THERMOCOUPLE LEAD
        1//1X,55H ENTER CHANNEL NUMBER AND HI
        1T <CR> WHEN READY TO BEGIN.)
1100    FORMAT(I5)
1200    FORMAT(1X,17H CHANNEL NUMBER =,I5,13H CHECK Y OR N)
1300    FORMAT(A5)
1400    FORMAT(/1X,33H SET MICROVOLT VALUE ON DAUPHINEE//1X,
        147H CAUTION: ONCE THIS TEST BEGINS, DO NOT ENTER A/
        11X,50H VOLTAGE OUTSIDE THIS INTEGER 1000'S OF MICROVOLTS/
        111X,41H I.E. THIS TEST IS FOR A CONSTANT K VALUE//
        11X,37H TYPE IN MICROVOLT VALUE AND HIT <CR>)
1500    FORMAT(F10.0)
1600    FORMAT(1X,6H EMF =,F10.2,13H CHECK Y OR N)
1650    FORMAT(1X,45H EMF OUT OF RANGE OF THIS UNIT OF SUPPRESSION)
1700    FORMAT(1X,40H DO YOU WISH TO ENTER ANOTHER EMF VALUE?/
        11X,13H CHECK Y OR N)
1800    FORMAT(1X,45H SET NEW VOLTAGE ON DAUPHINEE AND EN
        1TER VALUE)
1900    FORMAT(1X,39H USE AT LEAST THREE POINTS TO PRODUC
        1E A/1X,29H MEANINGFUL LEAST SQUARES FIT)
2100    FORMAT(//6X,1H#,6X,6HCOUNTS,5X,3HEMF,5X,9HCALC. EMF/)
2200    FORMAT(1X,I6,I11,2F11.2)
2300    FORMAT(///1X,41H ENTER FITTED VALUE FOR FIXED SUPPRESSION)
2400    FORMAT(/1X,12H INTERCEPT =,F10.2//1X,8H SLOPE =,
        1F10.4//1X,8H SIGMA =,F10.4/
        1/1X,14H FITTED GAIN =,F10.1//1X,27H FITTED FIXE
        1D SUPPRESSION =,F10.2//1X,30H FITTED VARIABLE SUP
        1PRESSION =,F10.2//)
2500    FORMAT(//1X,25H TO RUN AGAIN TYPE Y OR N)
5000    CALL EXIT
        STOP
        END
```

```
C       PROGRAM NAME--CPFIT
C       USED TO FIT A SET OF HEAT CAPACITY DATA TO THE EQUATION
C       CP=A+B*T+C*T**-2
        DIMENSION XTX(3,3),XTY(3),A(3),T(600),C(600)
C
C       M=NUMBER OF TERMS IN MODEL
C       N=NUMBER OF DATA POINTS
C
C       READ IN N AND M
C
        READ(1,1000) N,M
        M=3
C
C       READ IN DATA POINTS INTO MATRIX X (M ROWS,N COLUMNS)
C       ALSO READ IN DATA FOR DEPENDENT VECTOR Y (M ROWS)
        DO 10 I=1,M
        XTY(I)=0.
        DO 10 J=1,M
10      XTX(I,J)=0.
        DO 20 J=1,N
        READ(1,1100) D1,D2,T(J),C(J)
        XTX(1,2)=XTX(1,2)+T(J)
        XTX(1,3)=XTX(1,3)+1./T(J)**2
        XTX(2,2)=XTX(2,2)+T(J)**2
        XTX(2,3)=XTX(2,3)+1./T(J)
        XTX(3,3)=XTX(3,3)+1./T(J)**4
        XTY(1)=XTY(1)+C(J)
        XTY(2)=XTY(2)+T(J)*C(J)
20      XTY(3)=XTY(3)+C(J)/T(J)**2
        XTX(1,1)=N
        XTX(2,1)=XTX(1,2)
        XTX(3,1)=XTX(1,3)
        XTX(3,2)=XTX(2,3)
C
C       INVERT MATRIX XTX
C
        CALL INVERT(XTX,M,M)
C
C       CALCULATE VECTOR OF CONSTANTS (A) AND WRITE RESULT
C
        DO 90 I=1,M
        A(I)=0.
        DO 90 J=1,M
90      A(I)=A(I)+XTX(I,J)*XTY(J)
        WRITE(4,1200) (A(I),I=1,M)
C
C       USE CONSTANTS AND INDEPENDENT VARIABLES TO PREDICT VALUE
C       OF DEPENDENT VARIABLE.  WRITE ACTUAL DEPENDENT VARIABLE
C       AND PREDICTED DEP. VAR.  CALCULATE SUM OF SQUARES OF
C       DEVIATIONS AND WRITE RESULT.
C
        COST=0.
        DO 110 I=1,N
100     YY=A(1)+A(2)*T(I)+A(3)/T(I)**2
        WRITE(4,1300) T(I),C(I),YY
110     COST=COST+(C(I)-YY)**2
        ERR=SQRT(COST/(FLOAT(N-2)))
900     WRITE(4,1400) ERR
1000    FORMAT(2I5)
1100    FORMAT(4F10.0)
1200    FORMAT(1X,4HA = ,E15.6/1X,4HB = ,E15.6/1X,4HC = ,E15.6)
1300    FORMAT(F12.2,2F12.4)
1400    FORMAT(//1X,8HERROR = ,F15.5)
        STOP
        END
```

```
C       PROGRAM NAME--INVERT
C       USED TO INVERT AN N BY N MATRIX
C       OBTAINED FROM UTCC BY VERBAL PERMISSION
C       PART OF CANNED PROGRAM SOLVE
        SUBROUTINE INVERT(A,ORDER,NA)
        INTEGER ORDER
        REAL A(3,3),B(20),C(20)
        INTEGER IP(20),IQ(20)
        N=ORDER
        DO 1 K=1,N
        PIVOT=0.
        DO 100 I=K,N
        DO 100 J=K,N
        IF(ABS(A(I,J))-ABS(PIVOT))100,100,101
101     PIVOT=A(I,J)
        IP(K)=I
        IQ(K)=J
100     CONTINUE
        IF(PIVOT)102,900,102
102     IPK=IP(K)
        IQK=IQ(K)
        IF(IPK-K)200,299,200
200     DO 201 J=1,N
        Z=A(IPK,J)
        A(IPK,J)=A(K,J)
201     A(K,J)=Z
299     CONTINUE
        IF(IQK-K)300,399,300
300     DO 301 I=1,N
        Z=A(I,IQK)
        A(I,IQK)=A(I,K)
301     A(I,K)=Z
399     CONTINUE
        DO 400 J=1,N
        IF(J-K)403,402,403
402     B(J)=1./PIVOT
        C(J)=1.
        GOTO 404
403     B(J)=-A(K,J)/PIVOT
        C(J)=A(J,K)
404     A(K,J)=0.
400     A(J,K)=0.
        DO 405 I=1,N
        DO 405 J=1,N
405     A(I,J)=A(I,J)+C(I)*B(J)
1       CONTINUE
C
C       REORDERING THE MATRIX
        K=N
        DO 500 KDUM=1,N
        IPK=IP(K)
        IQK=IQ(K)
        IF(IPK-K)501,502,501
501     DO 503 I=1,N
        Z=A(I,IPK)
        A(I,IPK)=A(I,K)
503     A(I,K)=Z
502     IF(IQK-K)504,500,504
504     DO 506 J=1,N
        Z=A(IQK,J)
        A(IQK,J)=A(K,J)
506     A(K,J)=Z
500     K=K-1
        RETURN
900     RETURN
        END
```

APPENDIX H

MAJOR VARIABLES USED IN THE PROGRAMS LISTED IN APPENDIX G

Common A

NTICK - Number of ticks delay between closure of a multiplexer channel relay and reading the T/C value.

ITCR - Rate (in ticks) at which heat capacity T/C values are taken.

JTSIG - Integer sample T/C value (in ADC counts) used in heat capacity calculation.

IVS - Used to sum sample voltage (in ADC counts) from one heat capacity calculation to another.

ICS - Used to sum sample current (in ADC counts) from one heat capacity calculation to another.

INEC - Used to count number of executions of heat capacity program (CALØ4 or CALØ5) since last heat capacity calculation.

ITK - Number of ticks used to complete total time since starting clock.

ISEC - Integer number of seconds since starting clock.

IVSA - Used to store sample voltage (in ADC counts) after NTCCP additions to IVS.

ICSA - Used to store sample current (in ADC counts) after NTCCP additions to ICS.

IDTK - Integer number of ticks used to complete total time since last heat capacity calculation.

IDSEC - Integer number of seconds since last heat capacity calculation.

JSUM - Used in least squares sums for linear fit of sample T/C values for heat capacity calculation.

MSUM - Also used in least squares fit for heat capacity calculation.

NSUM - Also used in least squares fit for heat capacity calculation.

NTCCP - Number of sample T/C values collected between heat capacity calculations

VO - Sample voltage which is an average of NTCCP readings; calculated from IVSA

CU - Sample current which is an average of NTCCP readings; calculated from ICSA.

GAIN - Calibrated data acquisition system gain.

K - Integer number of 1000 μvs suppression being used at any given time.

SUP - Calibrated data acquisition system fixed suppression.

SMASS - Sample mass.

CP - Heat capacity.

TEMP - Temperature in °C. First four are used in heat capacity calculation; last eight in printing all eight T/C temperatures.

MM - Used to close data file which stores modeling data collected by DDCAL.

ISPAN - Number of ADC counts in 1000 μvs of suppression.

Common B

L - Used to assign lines connecting the calorimetery lab to the computer lab to the DR-90 relays, A/D converter channels, and D/A converter channels.

ISPEC - Sample T/C value (in ADC counts) used to key suppression from any data acquisition program.

IPCR - Rate (in ticks) at which digital controller executes.

ITSIG - All eight T/C values (in ADC counts) collected by digital controller.

FE - Filtered error (difference between shield and sample temperature - in ADC counts); not used.

FC - Filter constant used to generate filtered error; set to zero.

E - Error (in ADC counts) between shield and sample temperature. The first subscript denotes the errors at i-1 and i-2 sample intervals. The second denotes the controller. G3 is controller #1; G1 is #2; Cyl is #3; and G2 is #4.

U - Controller output (floating point, but rounded off to DAC counts). Subscript one denotes output at i-1 and i-2 sample intervals. The second denotes the controllers as listed above.

JOF - Offset (in ADC counts) of shield temperature from sample temperature.

IRNG - Integer μv range for error meters.

MI - Minimum controller output (in DAC counts) to activate power amplifiers.

MA - Maximum controller output (in DAC counts) to reach full output of power amplifiers.

M - Sample voltage and current in DAC counts collected by digital controller.

C - Digital controller algorithm constants. Subscript one denotes constants C1, C2, C3, and C4. Subscript two denotes the controller number as described in the variable E.

GC - Digital controller gain (K_c).

TI - Digital controller integral constant (T_i).

TD - Digital controller derivative constant (T_d).

GL - Second order digital controller gain in lead compensator (K_L).

GP - Second order process gain used in digital controller (K_p).

TP - Second order time constant used in lead compensator used in digital controller (τ_p).

APPENDIX I

CALORIMETER OPERATING MANUAL

This appendix serves to present a step-by-step procedure for start-up and shutdown of the calorimeter. The complete procedure for collecting and analyzing data for heat capacity and modeling is presented. Also the procedure to use the model parameters to obtain controller tuning parameters is described.

Calorimeter Operation

1. Walk into Room 317. Load two RSX Warmstart tapes into PDP-15 using standard procedures.

2. At maxipanel, connect TTY port 00 to any available ASR33 if heat capacity data are to be collected. Connect TTY port 01 to Lab 00, Room 323. Run trunk cable from Lab 00 multiple plug to ADC panel 00 (plug with fewest pins) and to DR-90 relay panel 00 (plug with most pins). Disconnect all patch cords from ADC channels 0, 14, and 15 and DR-90 relays 1 to 13. Place yellow caps over these patch cord connections.

3. Connect patch cords for Lines 16 to 23 from Lab 00 to DAC Channels 01 to 08, respectively.

4. Return to TTØ and reassign LUN's 2, 3, and 4 to TT2 from TTØ using reassign (REA) command.

5. Place a sign on TTØ that it is in use in Room 323. If heat capacity data are to be collected, set up ARS33 to store raw heat capacity data by placing it in local and feeding out some paper tape.

This is done by turning the paper punch on and hitting the "HERE IS" key several times. Then type out the sample being used, type control (analog or digital), and date. End this with a <CR> and <LF>. Then feed out some more paper tape as described above. Turn paper punch off, place teletype in on-line position, and turn paper punch back on.

6. Go to calorimetry lab, Room 323. Put ice in the thermocouple cold junction and liquid nitrogen in the cold trap for the vacuum system. Plug the dummy load (10 ohm resistor) into the specimen power supply, turn it on, and adjust voltage to approximately 10 volts. This warms up the power supply.

7. Check the thermocouple connections in the calorimeter by reading each leg of thermocouple to ground using the standard procedure devised by Dr. C. R. Brooks. Also read the thermocouple emfs using the Dauphinee potentiometer. Return switches II-A and II-B used in reading the thermocouples to 0 or noise will be injected into the digital sampling system.

8. Check the thermocouple connections to the multiplexer. Channels 1 to 6 should be connected to thermocouples Spec. Ref., Spec. Control, G3 control, G1 control, Cyl control, and G2 control, respectively. Channels 7 and 8 can be used to read any thermocouples desired. Currently they are connected to Lower G1 and II-B-10.

9. Turn all switches on the mini-box to the on position.

10. Request program STUPP using the request (REQ) command. Terminate with a <CR>.

11. Request RSTUP and terminate with a <CR>.

12. Turn on the data acquisition system located in the cabinet on the west side of calorimeter. This is done by placing the 30 volt and 6 volt batteries in the "operate" position, setting the Astrodata® 120 amplifier to a gain of 5000, and turning it on. Read the output from the Astrodata® 885 amplifier using the DVM at the terminal strip (just below 885 amplifier in front of cabinet). It should be about -6 volts at room temperature.

13. Disconnect the DVM from the 885 output and unplug cabinet. (This is extremely important; otherwise, the DVM and AC power line will inject noise into the thermocouple signal.)

14. Execute the program CAL1Ø to check quality of digital signal using RUN command as follows: MCR> RUN CAL1Ø ØS 2ØS (<ALT MODE>). This command executes CAL1Ø every 20 seconds. It will display the microvolt values for all eight thermocouples. The first six (used in control) should be high quality and within 2 or 3 μvs of the values obtained in Step 7. They should read within ± 0.28 microvolts for successive readings. If deviations of more than ± 1.00 μvs are obtained, something is wrong and noise problems should be found. Let this program run while completing Step 15.

15. Turn on power supplies.

G3 - In cabinet left of console, set G3 selector switch to position 2, turn G3 power switch to on, and turn LABAC on (back of cabinet). Set dial on top of LABAC to 0.6.

G1 - In the cabinet left of console, set G1 selector switch to CAL II. On east side of calorimeter, turn on fan in power supply, set voltage to 73 (and not above), and turn power supply on.

Cyl - On west wall of lab, set selector switch to position 2 and set voltage selector to 250 volts. If it buzzes when this is done, turn voltage to off and then back to 250 volts. If buzz doesn't stop, this is usually symptomatic of the Cyl heater being grounded.

G2 - On panel behind calorimeter, set G2 selector switch to CAL II and set to analog or digital control.

Bottom Heater - Turn on bottom heater power supply located below G1 power supply. This heater is driven by a percentage of the power to the Cyl.

Set switches on panel in cabinet left of console to analog or digital control for the shields G3, G1, and Cyl.

16. Cancel CAL1Ø using CAN command after it has run 15 to 20 minutes.

17. Attach the "Specimen Power/Digital" lead to the standard 1 ohm resistor.

18. If making a modeling run, proceed to Step 32.

19. If making a heat capacity run under digital control, proceed to Step 32.

20. At this point, a heat capacity run under analog control should be under way.

21. In the cabinet to the left of the console, turn the speedomax switch to on. This activates the recorders and controllers. Set all controllers to manual and the current adjust to zero (full counterclockwise). Set the controller values and pot (variable resistor located to the left of each controller) setting to the values listed in Table XIV.

TABLE XIV

ANALOG CONTROLLER VALUES

Shield	Proportional Band	Rate Time	Reset	Pot Setting
G3	80	0.07	3.0	2.55
G1	2	0.05	0.1	7.70
Cyl	25	1.00	0.2	9.00
G2*	60	1.00	0.2	--

*Impulse frequency = 4.

22. Turn on microvolt amplifiers to the right of the console. Set all scale multiplers to 4 (except G2, set it to 200). Turn on all choppers located just below amplifiers. Turn switches on the amplifiers to the "Recorder" position.

23. Standardize all four controllers using standard technique. (If red dot shows in hole beside standardization knob, replace 6 volt battery which drives recorder.)

24. Put all controllers in automatic.

25. Request the program HTCAP <ALT MODE>. It will ask several questions about the heat capacity run to be made. Answer these and then type N when it asks if you want digital control. This program will then calculate the rate at which to take heat capacity data (ITCR) and print its value in ticks. It then fixes CALØ4 in core and starts its execution.

26. Let the heat capacity program run until it has printed 4 or 5 sets of heat capacity data with no power to the specimen. The raw voltage, current, time, and temperature data will be printed on TT1 and, therefore, stored on paper tape. The uncorrected heat capacity will be printed on TTØ along with other pertinent data. The value of sigma printed on TTØ should be less than 1.0 (usually between 0.2 and 0.7). If it is greater than 1.0, the data are too noisy and problem areas should be checked.

27. At this point, heating can be started by adjusting the voltage on the power supply for the specimen to zero, disconnecting the dummy load, connecting the "Specimen Power/Digital" lead to the power supply, and turning up the voltage again. Attach the DVM to the

"Specimen Voltage/Digital" lead at the calorimeter and turn up the power supply voltage until it reaches the value printed by the program HTCAP in step 25. Then disconnect the DVM from the specimen voltage lead or noise will be injected into the digital signal.

28. Monitor control action and heat capacity data throughout the entire run. If any upsets occur, take corrective action or begin shutdown procedure which follows.

29. After completing the heat capacity run, cancel CALØ4 and turn off power to specimen. Take drift data if necessary using the Dauphinee potentiometer. Then shut calorimeter down by putting all controllers in manual. Set the amplifiers to "non-linear" and turn them off. Turn off choppers and turn speedomax switch to off.

30. Turn off power supplies by doing the reverse of step 15. Switch all selectors to Cal III from Cal II. This protects current from accidently being pulsed to Cal II. Turn off Astrodata ®120 amplifier, flip the 30 and 6 volt battery chargers to "charge," plug in data acquisition cabinet, and flip all switches in the mini-box to the off position.

31. Collect heat capacity data from TTØ and TT1. Save all this along with the paper tape.

32. This is the digital control section. Check the connection of power supplies to the computer by pulsing power to the heaters as follows. Put the analog/digital selector switches in the cabinet to the left of the console in the analog position. Set DAC #1 (G3 controller) to 4 volts using the STD commands (MCR>STD 1:1:400). Flip G3 selector switch to "Digital" and the voltmeter attached to G3

heater should read ~ 6 volts. Flip the switch back to the analog position and use the STD command to reset DAC #1 output to Ø. Repeat this procedure for G1 (using DAC #2) and the Cyl (using DAC #3) except set their DAC values to 3 volts. G1 should put out ~ 12 amps and the Cyl should put out ~ 40 volts (on the 150 volt scale) at this DAC value. If these values are not obtained, check to make sure that the pot settings on the variable resistors located below the analog/digital switches are set to 7.0 for G3, 9.6 for G1, and 8.9 for the Cyl. In the case of the shield G2, set its DAC output to 3 volts (using DAC #4), check to make sure the heater voltage is ~ 70 volts on the 300 volt scale on the voltmeter attached to G2 heater, and immediately reset its DAC output to 0.

33. Request RSTUP (<CR>) and flip the selector switches for G3, G1, and Cyl to the digital position that were used in step 32 to check the computer link.

34. If a modeling run is to be made, proceed to Step 40.

35. Request HTCAP (<ALT MODE>). It will ask several questions about the heat capacity run to be made. Answer these and type Y when it asks if you want digital control. This program will then calculate the rate at which to take heat capacity data (ITCR) and print its value in ticks. It then fixes CALØ5 and DDCAL. Then it starts the execution of DDCAL, waits 1 second, and starts the execution of CALØ5. This synchronizes these two programs.

36. Follow Steps 26, 27, and 28; but return to this step after their completion and not to step 29.

37. After completing the heat capacity run, turn off the power to the sample. Allow DDCAL and CALØ5 to continue to run. The drift rate can be calculated from the values DT1 and DT2 by

$$D_R = \frac{DT\ (3600)}{(.19)\ ITCR} \qquad I\text{-}1$$

where DT = DT1 or DT2

D_R = drift rate (°K/hr).

(This equation arises from the fact that DT is the temperature change from the first point and the twentieth point in the least squares fit of the 20 T/C values taken by CALØ5. The total time from one heat capacity calculation to another is 20 * ITCR/100 seconds; therefore, the time from the first point to the last point is 19/20 times this value. This calculates to a value of 0.19 * ITCR for the total time over which DT is measured. Thus, the drift rate is equal to the temperature change (DT) divided by this time change (.19 * ITCR) times 3600 to convert from °K/sec to °K/hr).

38. Allow these programs to run until a steady drift rate is obtained. If drift at a lower temperature is desired, then cancel DDCAL and CALØ5. Request RSTUP and run CAL1Ø every 60 seconds using the command in step 14 but with 6ØS rather than 2ØS. Allow the sample to cool to the desired emf value, cancel CAL1Ø, and request HTCAP. Give the same values used when run was initially started. This will reactivate the controller and heat capacity programs again. Then collect drift as described in step 37.

39. Terminate the run by cancelling DDCAL and CALØ5. Then turn off all equipment as described in steps 30 and 31.

40. This is the modeling section. Fix DDCAL in core using the FIX command. Then request DDCAL. Start the specimen heating as described in step 27. Manually calculate the specimen voltage from the desired heating rate to be used in the modeling run (usually between 2 and 4 °C/minute). Set this voltage as also described in step 27.

41. Watch the voltmeters which display the controller errors in μvs and controller output in DAC volts. When the controllers are in good control, collection of modeling data may begin.

42. Place an empty Dectape on Channel 8 and place in "Remote" and "Write Enable."

43. Request the program STORE (terminate using <ALT MODE>). Enter the filename under which data are to be stored. Hit CTRL C and wait 60 to 90 seconds. Then request the program STEP (terminate with <ALT MODE>). Specify the controller number (G3 = 1, G1 = 2, Cyl = 3, G2 = 4) and step size in DAC counts (usually 30 for G3 and G1; 10 for Cyl and G2 - but as the temperature increases larger values can be used, especially in the case of G1). When the response in this shield reaches steady state, enter Ø and this returns the shield to controlled conditions and closes the data file.

44. Repeat step 43 until the calorimeter reaches 700 or 800°C. Terminate run by cancelling DDCAL and following steps 30 and 31.

45. Remove the Dectape used to store the modeling data. Note: During the run, periodically check the Dectape directory to make sure enough room remains on tape to make additional tests. (G3 and G1 tests usually use 20-100 base 10 blocks on the tape while the Cyl and G2 use 50 to 200 blocks.) Actually the amount of storage in base 10 blocks

is equal to six times the number of minutes during the step test. This time can be estimated from the data given in Chapter III.]

Analysis of Heat Capacity Data

Once a heat capacity run has been made, the data stored on paper tape can be used to regenerate the uncorrected heat capacity. This can then be corrected for heater mass and drift rate.

46. Load the KM15-V5A monitor into the computer. Assign LUN -10 to the paper reader. Request the editor. Open a file consisting of the data of the run (month, day, and last digit of year, e.g., a run made on Sept. 12, 1979 would be called 09129). Use the GET command to read in the paper tape generated on TT1 during the original run. Read 50 lines at the time until the entire tape is read in (this prevents parity errors). Check for any reading errors and put a one in the first space after the last line in this data file. Close this data file and dump it into a Dectape.

47. Make any corrections in bad data points either in the KM15 monitor or in RSX. These corrections are of two types:

1) Errors due to a mismatch in the suppression system, and

2) Stray bad values.

The only data sets in which the first type error occurs are those in which the suppression is set. These sets are easily recognized because the T/C emf (in ADC counts) jump from positive with a value > 1600 to negative with a value of < -1500. They can be easily corrected by plotting all 20 T/C values (in ADC counts) against the sample number I (I = 1, 20). This will yield points which lie on two straight lines with identical slopes but have a constant offset from each other.

By adding this offset to one of the segments so that all points lie on the same line, the suppression mismatch can be eliminated. This is only necessary if the sigma value on the 20 values is greater than 1.0. This correction eliminates the loss of one heat capacity value when the suppression is set.

The second type error is also found by examining the sigma value on the least squares fit of the 20 T/C values in counts. If this value is greater than 1.0 and the data set is not one in which the suppression was set, one bad value is usually the cause. It can be easily found by examining the difference between successive values of the T/C points. Since the specimen is heating at a constant rate, the difference between values will be a constant. A data set heating at a rate of 20 counts per sample interval with one bad point might appear like this:

10 30 50 65 90 110 . . .

The differences between these values are:

20 20 15 25 20 . . .

It is obvious from this that the one bad point is 65 since the total change from 50 to 90 is 40 counts or an average of 20. By adjusting 65 to a value of 70, the bad estimate of the slope of this line can be alleviated and the erroneous heat capacity value can be corrected.

Based upon claims made in the main body of this work, suspicions as to these claims might be raised. But these errors should be put in perspective. A mismatch in the suppression of 4 to 8 counts is sometimes encountered. Since the total range a 1000 μv suppression unit spans is 3600 counts, this is only an error of 0.2% in the worst case. Also in previous work (26), some 978 heat capacity points were obtained.

Since 20 T/C values were collected for each of these, a total of 19,560 values were obtained. Of these, less than 25 were of the second type error. This is less than 0.13%.

48. After these corrections have been made, dump this corrected file back onto the Dectape.

49. Load in the RSX operating system. Install the programs RADFR and CAAØ8 into RSX using the Install command. (These files are on a Dectape labeled RSX SRC and TSK files #2.) Use the programs CALØ1 and CAAØ1 to specify the variables SMASS and NTCCP to the values printed by CALØØ during the original run.

50. Edit the file CAAØ9 and change the equations for heater mass and drift rate. Rebuild this file using the Task Builder using a priority of 300 and a partition of P1. Reassign LUN 5 to RFØ from DTØ using the reassign command. Install the program CAAØ9.

51. Connect any available ASR33 to TTY port 00. Feed out some paper tape using the "HERE IS" key. Turn paper punch off, put TTY in "on line" and turn paper punch back on. Place the Dectape with the corrected heat capacity T/C values on Channel 8 and put in "Remote." Request RADFR (terminate with <ALT MODE>). It will ask for the filename and filter constant. Enter the name of the file under which the T/C data are stored and a filter constant of 0. At this point RADFR will read the data from the Dectape and start execution of CALØ6, CAAØ8, and CAAØ9. The heat capacity data along with the uncorrected and corrected heat capacity values will be printed on TTØ. The average temperature (in °K) and corrected heat capacity will be printed on TT1 and simultaneously punched on paper tape. Save the printed heat capacity values and the paper tape.

52. Load in the KM15 monitor. Load in the programs CPFIT and INVERT from the same Dectape that contained RADFR. Put these files in binary using the F4 compiler.

53. Put the paper tape containing the corrected heat capacity values in the paper tape reader. Transfer this file to DSK 1 using PIP. Edit the file and put the number of heat capacity data points and the number 3 in the first line of this file using a 2I5 format. (This tells the heat capacity fitting program the number of points in the fit and the number of terms (3) in the fit.) Close this file and dump it back onto the paper punch. It should be noted here that several heat capacity runs can be put together using the Editor and the GET command. The number of points will then be the total number for all the runs in the file.

54. Place this new paper tape in the paper tape reader. GLOAD the program CPFIT. It will read the data on the paper tape, calculate a, b, and c in the heat capacity equation, and print out all the results. This completes the heat capacity analysis.

Analysis of Modeling Data

After the collection of modeling data, it is necessary to set up the files necessary to analyze the data. These files can then be used with the appropriate programs in the DEC-10 to determine the process parameters.

55. Load the RSX operating system into the PDP-15.

56. Install the program RESP into the system from the Dectape (on LUN 8) labeled RSX SRC and TSK files #2.

57. Place the Dectape containing the modeling data on LUN 8. Put the file to be set up for modeling on the TDV disk. Edit this file and remove the first several lines up until the step began. Close the file and dump it onto a clean Dectape (different from the one the data were taken from).

58. Put this new Dectape on LUN 8. Set up an ASR33 teletype on TTY part 00. Turn it to "on line" and turn the paper punch on. Request RESP (terminate with <ALT MODE>). It will ask for the file-name to be analyzed, the sampling time when the test occurred (in ticks), and the controller that was stepped, and the controller step size. It will then write these on TTl along with the step test data for that shield. At the end, it will print the total number of points in the step test.

59. Place these data on another Dectape. Carry this tape and the original tape to Stokley Management Center. Place the tape with the step test data on Channel Ø of the DEC-10 Dectape Drive. Log onto the DEC-10. Go to the routine FILEX by typing:

.R FILEX

Then type:

*Filename. DAT = DTAØ: Filename. SRC/F

This places the step test file into the DEC-10. Then place the Dectape with the original modeling data on drive Ø and type:

Filename . TST = DTAØ: Filename. SRC/F

This places the original file with all the modeling data into the DEC-10.

60. Copy Filename. TST in a file called FOR21.DAT. Execute the program CPLST.F10 (obtained from C. F. Moore). Edit the file Filename. DAT and place the number of data points in the file in the second line of the program right after the sampling time in an I5 format. (The number of points should be the highest multiple of 12 but does not exceed the total number of points printed by RESP. For example, if RESP prints the total points to be 292, enter 288 in the DEC-10 file in the first line.) Copy this file into a file called FOR20.DAT. Then execute either MOD1.F10 (for the shields G3 and G1) or MOD2.F10 (for shields Cyl and G2).

61. Collect the outputs of CPLST.F10 and MOD1.F10 (or MOD2.F10) from the BINS. CPLST gives a complete response for all shields and the modeling programs give the parameters and plots for a first or second order lag plus dead time model depending on the program used.

Controller Tuning

Once modeling data have been obtained, they can be used to tune the appropriate controllers.

62. Place the modeling parameters in a file called FOR20.DAT. The first line contains the sample time in seconds, shield name, minimum controller output and maximum controller output (in DAC counts), and the number of parameters to be optimized (NP). The second line contains the process gain, dead time, time constant (for first order shields) or damping ratio and reciprocal process frequency (for second order shields), and the average controller output (in DAC counts). Several of these can be repeated. The end is signified by entering a process gain of 0.0.

63. Execute the program TUN.F10 to tune a PI controller (NP=2) or a PID controller (NP=3) to a first order lag plus dead time process.

64. Execute the program TUN2.F10 to tune a PI controller (NP=2) or a PID controller (NP=3) to a second order lag plus dead time process.

65. Execute the program LDLG.F10 to tune a PI plus lead compensator to a second order lag plus dead time process (NP=3).

66. Pick up the results at the BINS.

VITA

William Merle Cash was born in Milan, Tennessee, on February 3, 1950. He attended elementary schools in that city and was graduated from Milan High in 1968. The following September he entered The University of Tennessee, Knoxville, where he graduated in June of 1973 with a Bachelor of Science in Chemical Engineering. The following September he entered The Graduate School at The University of Tennessee, Knoxville, where he graduated in December of 1976 with a Master of Science in Chemical Engineering.

He is married to the former Miriam Elizabeth Toomey.