Development and evaluation of a dynamic phantom using four independently perfused in vitro kidneys as a tool for investigating hyperthermia systems

Order Number 1337996

Development and evaluation of a dynamic phantom using four independently perfused *in vitro* kidneys as a tool for investigating hyperthermia systems

Zaerr, Jon Benjamin, M.S.

The University of Arizona, 1989

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

DEVELOPMENT AND EVALUATION OF A DYNAMIC PHANTOM USING FOUR INDEPENDENTLY PERFUSED IN VITRO KIDNEYS AS A TOOL FOR INVESTIGATING HYPERTHERMIA SYSTEMS

by

Jon Benjamin Zaerr

A Thesis Submitted to the Faculty of the

DEPARTMENT OF AEROSPACE AND MECHANICAL ENGINEERING

In Partial Fulfillment of the Requirement
For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN MECHANICAL ENGINEERING

In the Graduate College

UNIVERSITY OF ARIZONA

1 9 8 9

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Jon B. Zaer

APPROVAL BY REPORT COMMITTEE

This report has been approved on the date shown below:

Robert B Roemer — 7/28/89

Dr. R. Roemer
Professor of Aerospace
and Mechanical Engineering
Report Advisor

Date

7/28/89

Dr. K. Hynynen
Assistant Professor
Department of Radiation Oncology
Second Reader

Date

ACKNOWLEDGMENTS

I stepped from Plank to Plank
A slow and cautious way
The Stars about my Head I felt
About my Feet the Sea.

I knew not but the next
Would be my final inch-
This gave me that precarious Gait
Some call Experience.

Emily Dickinson

This poem expresses to me the difficulties sometimes encountered in the pursuit of knowledge. I would like to thank the following people for helping me from plank to plank: Kullervo Hynynen, Robert Roemer, Chris Diedrich, Dennis Anhalt, Bonnie Billard, Win-Li Lin, and Edwardo Moros. I would also like to thank my parents Joe and Lois, my sisters Linda and Laura, Margaret Hanson and all of my other friends for their moral support and helpful ideas.

TABLE OF CONTENTS

TABLE OF CONTENTS--Continued

TABLE OF CONTENTS--*Continued*

LIST OF FIGURES

LIST OF FIGURES--Continued

Figure Page

LIST OF TABLES

ABSTRACT

A dynamic phantom for use in investigating hyperthermia heating systems has been designed, constructed, and tested. A computer controlled the flow rate of 80% Ethanol to each of 4 preserved _in vitro_ canine kidneys which acted as the phantom material. The flow rates were regulated with stepper motor controlled valves and measured with flow meters by the computer. This provided a flexible system for adjusting the perfusion as desired. The system was tested with step and ramp changes in perfusion under constant power ultrasound and with a temperature controlled perfusion algorithm, all of which yielded repeatable results.

The dynamic phantom developed in this work shows potential for expediting investigations of hyperthermia controllers, temporal blood flow patterns, and inverse problems. It's computer based nature gives it great flexibility which would lend itself well to automated testing procedures.

CHAPTER ONE

INTRODUCTION

1.1 Background

A necessary criterion for any hyperthermia heating system is the ability to quickly reach and maintain a therapeutic target temperature. Well designed controllers are needed to accomplish this feat for several different heating systems. One of the problems in designing such a controller is that it must behave similarly under a wide variety of loading conditions. In the interest of testing such controllers prior to patient use, a dynamic phantom has been designed and constructed. Dynamic phantoms can be perfused at various rates which enable them to simulate certain treatment conditions such as changing blood flow. Such changes can drastically alter the power required to maintain therapeutic temperature.

The system developed in this work is based on a process developed by Holmes et al. (1984). Johnson (1987) and Kress (1988) also used preserved in vitro canine kidneys. In their adaptive hyperthermia controller studies, they used water as a perfusate, which required rehydrating the organ prior to use. The roller pump they used required calibration at every flow rate.

Benkeser et al. (1989) developed a perfused porcine kidney and showed its potential for studying hyperthermia, but presented no treatment simulations.

Although dynamic phantoms using in vitro kidneys have been used before, none have been developed under computer control, which gives great flexibility in designing experiments.

The remainder of this chapter deals with the criteria used for the design of the phantom. Chapter Two details the design, construction and calibration of the phantom and associated paraphernalia, while Chapter Three demonstrates the utility of the phantom and presents some results of an investigation of a hyperthermia controller.

1.2 Criteria

In order to produce a reliable system, a set of design criteria was developed. They are listed below with brief explanations.

Power absorption: The acoustic power absorption in the phantom tissue must be sufficient to provide at least a 6 °C temperature elevation at 'normal' power levels.

Temperature measurement: A high density of thermocouples was desired to provide good knowledge of the temperature distribution.

Longevity: The phantom should last one year without major changes in perfusion patterns.

Flow rate: An accurate knowledge and control of flow rate into each perfusion region was essential. Also, an approximation for the spatial variations in perfusion was desired.

Ease of Use: The procedure to bring the phantom from storage to usage must be brief.

CHAPTER TWO

CONSTRUCTION AND CALIBRATION

This chapter deals with the construction and operation of the dynamic phantom. First, a general description of the system is given. Then, a description of transducers concerning flow, pressure and temperature measurement is presented. Calibration of pertinent transducers is also given here. The operation of the computer program which controls the flow rates follows, and last, an estimate of the perfusion in an in vitro canine kidney is developed using effective thermal conductivity measurements.

2.1 Overview

An Apple II Plus computer controlled the flow rates to, and measured the temperatures in the phantom, as shown in Figure 2-1. An external interface was constructed which provided multiplexing and conditioning for the thermocouple temperature sensors, denoted 'flow control thermocouples, and amplification to adjust and measure the flow rates. The flow rates were adjusted with stepper motor controlled valves. Flow rate feedback was provided to the computer from flow meters in line with each valve.

The dynamic phantom is shown schematically in Figure 2-2. Four kidneys were sealed in an air tight clear PVC box with a PVC sonication membrane on the bottom. The air tight seal was necessary to maintain the degassed state of the Ethanol (the

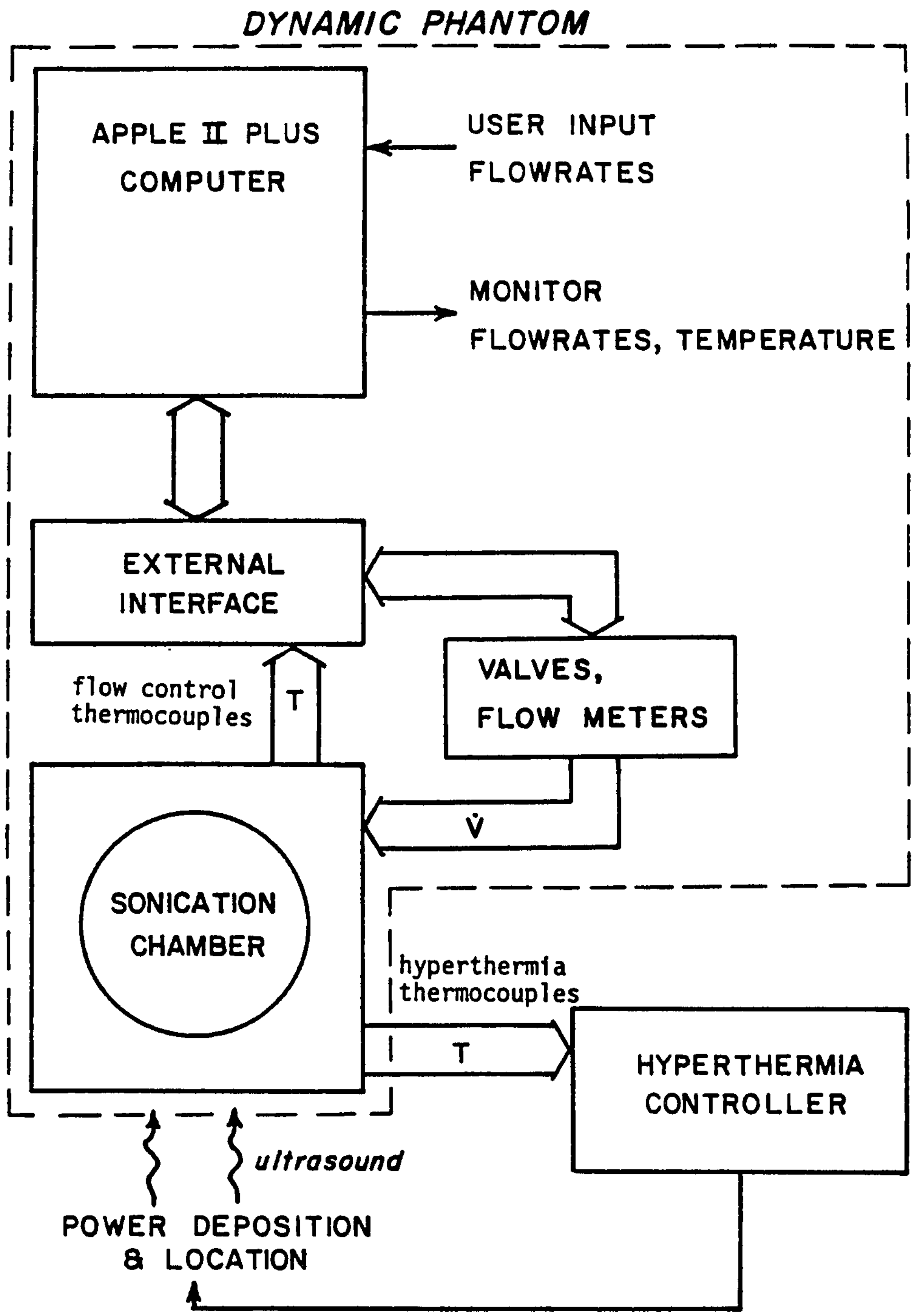


Figure 2-1 Block Diagram of the Dynamic Phantom. The Apple computer controlled flow rates to the phantom and measured the temperature and flow rate through the external interface. The hyperthermia controller measured the temperature and controlled the heating power.

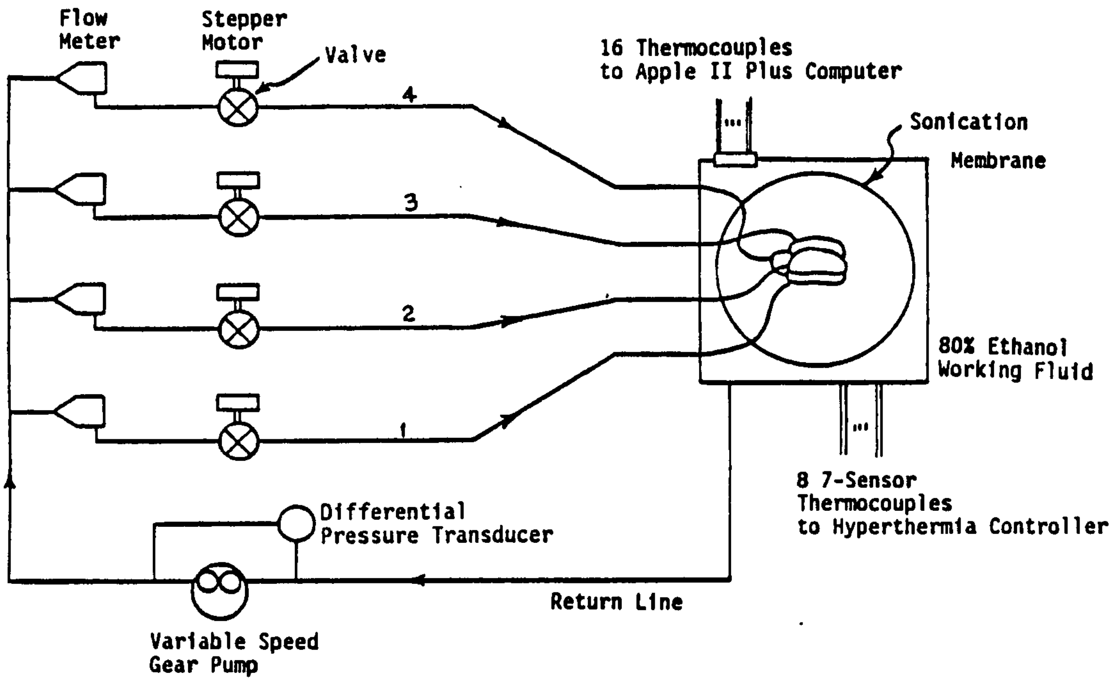


Figure 2-2 Schematic Diagram of the Dynamic Phantom. 80% Ethanol was pumped through 4 flow meters and valves into the kidneys. One set of thermocouples in the kidneys interfaced with the Apple computer, and a separate set was connected to the hyperthermia controller.

fluid must be degassed for efficient ultrasound coupling). As noted earlier, the kidneys were preserved using an alcohol fixation process developed by Holmes. The last stage of this process involves perfusing the organ with 80% alcohol, which was the working fluid of the dynamic phantom presented here. This minimized biological degradation of the phantom while maintaining ease of operation, since the rehydration process is eliminated. Although the effects of long term storage (over one year) on alcohol fixated kidneys is not well known, the organs can easily be replaced when degradation interferes with operation.

The phantom temerature was measured by 16 'flow control' thermocouples. Each kidney contained 4 thermocouples. A gear pump forced the fluid through the system, and the flow rates to each region were set by stepper motor controlled valves and measured with rotating disk flow meters. Physiological characteristics of various tissue types could be approximated by setting the flow rates to those found in the corresponding tissue types. These flow rates could either be programmed to change at pre-set time intervals, programmed to change as the phantom temperature changes, or could be changed by the operator during a simulation. Changes in blood flow due to tissue heating is a normal response of many kinds of tissue (Song et al. 1984), so the ability to change perfusion levels during a treatment enabled the phantom to simulate a wide range of situations which the hyperthermia controller may encounter, thereby providing a realistic test for the hyperthermia system.

An Applesoft Basic program was written to control the flow rates in the aforementioned manner. The main control modes of this program were as follows:

Passive: The flow rates and temperatures were simply displayed on the screen with no attempt to regulate flow.

Interactive: Target flow rates could be specified at any time. The flow meters were continuously monitored and the valves manipulated to achieve the specified flow pattern.

Programmed Flow: A list of flow rates and time durations was programmed, and the computer simply stepped through this list.

Temperature Controlled Flow: The flow to each kidney increases as a function of time at an elevated temperature. This was meant to model the physiological response to heating of various tissue types.

The program is listed in Appendix A, and the supporting machine code subroutines are shown in Appendix B.

Communication between the computer and the external interface was provided through two interface cards in the Apple computer. One was a 16 bit California Computer Systems parallel interface, and the other was an Applied Engineering 16 channel, 12 bit ADC (analog to digital converter). These two interface

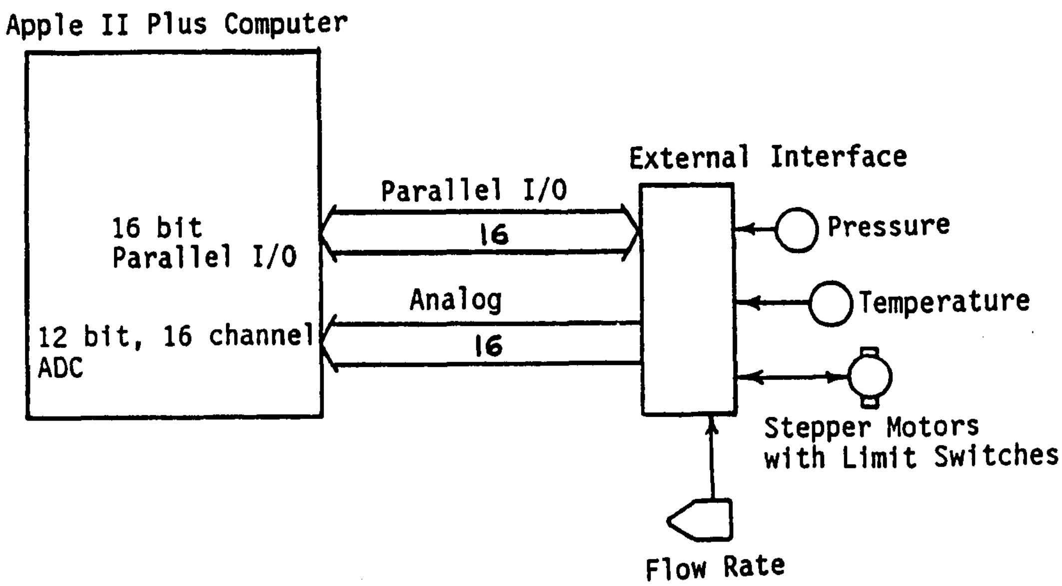


Figure 2-3 Block Diagram of the Instrumentation. Two interface cards in the Apple computer communicated with the external interface. Pressure, temperature and flow rate were measured with the ADC. The parallel I/O positioned the stepper motors, read the associated limit switches and multiplexed the thermocouples.

alcohol were brass. The exact alloy type was unknown, but some types of brass used in the development were found to be unacceptable. This necessitated compatibility testing of materials prior to use. Prospective materials were simply soaked in 80% Ethanol for up to two months. They were periodically checked, and if no changes were apparent to the material or the Ethanol, the material was judged to be compatible. One type of brass tubing, for example, dissolved slowly in the ethanol leaving a blue tint (perhaps due to a high copper content). Other parts which did not contact the Ethanol were machined from aluminum or plexiglass.

2.3 Pump and Valves

The Ethanol was forced through the system with a gear pump equipped with an overpressure by-pass set at 13 psi. The by-pass prevented the gear head from stalling during no-flow conditions. The valves were driven by stepper motors (7.5° per step) with notched belts. The motor pulleys had 12 teeth and the valve pulleys had 48 teeth for a gear ratio of 1:4. This provided greater valve torque and better resolution of valve position (192 motor steps per 1 valve rotation). The motor phase, generated by the computer, was amplified in the external interface as shown in Figure 2-4. The 2N2222 transistors (Q1, NPN array) acted as switches for the D41K (Q2) and D75FY4D (IC4) PNP power transistors. This signal (14V) was applied to only one motor at a time through 8 four pole relays. These relays

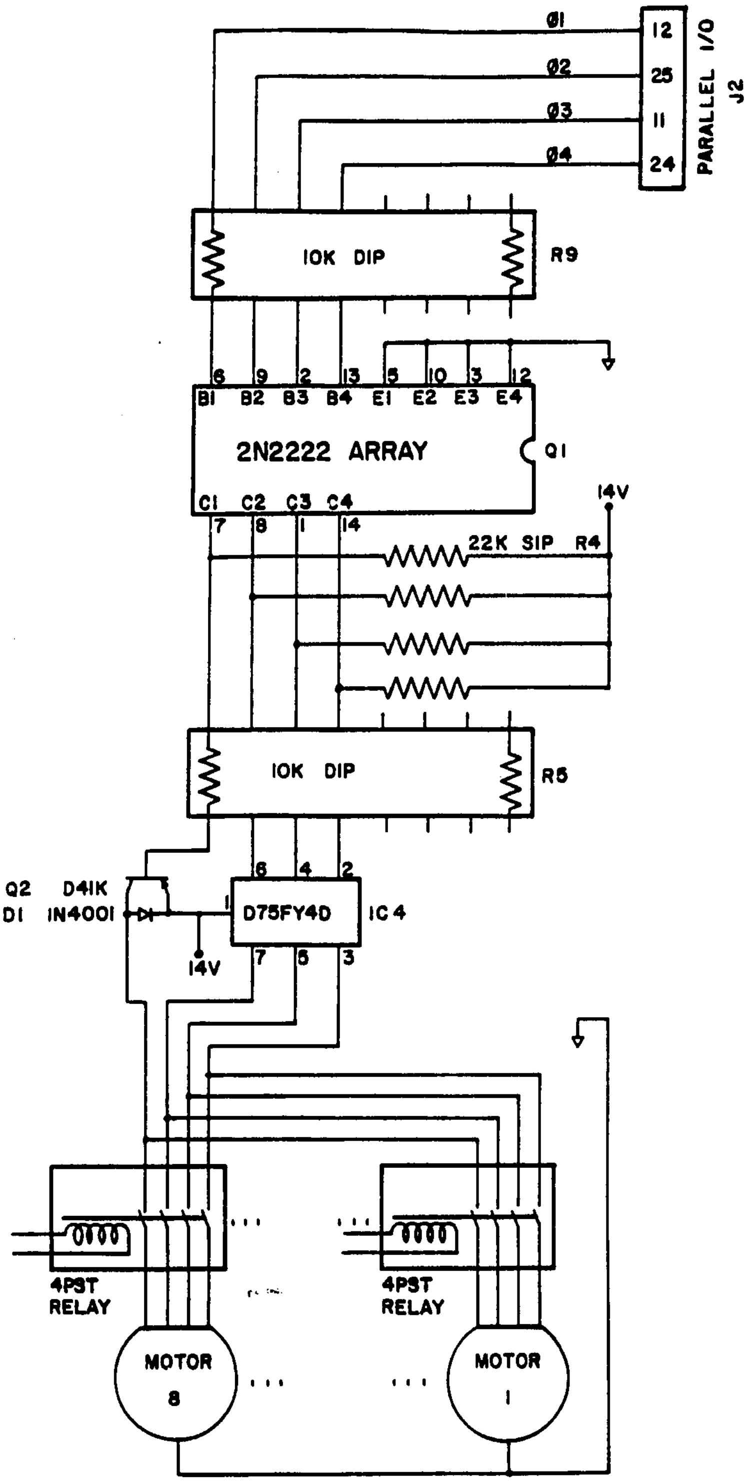


Figure 2-4 Phase Amplification for Stepper Motors. The motor phase, generated by the Apple computer was amplified to 14V in the external interface.

were activated by the expander circuit in Figure 2-5. The LS138 (IC7) expanded three input lines from the computer into 1 of 8 outputs. These outputs were inverted by the F540 (IC8) and amplified by 8 2N2222 transistors (Q3). This signal activated the coil of one of the 4 pole relays which enabled the corresponding motor. Limit switches, shown in Figure 2-6, were used to initialize the motor position upon start up. A 4.7k resistor array (R3) acted as a pull-up to 5V, so each switch line was high unless it's limit switch was closed. The state of each switch line was determined by the 74150 data selector (IC2). One of 8 inputs (E0-E7) was selected with control lines A, B and C. The output was present on line W (pin 10 of the data selector). A machine code subroutine was used to provide sufficiently fast stepping speed and to decrease the Basic program software requirements. The program compensated for mechanical hysteresis in the motor-valve linkage by moving one step farther than requested when traveling opposite to the previous movement. In this manner, the slack in the belts was compensated for by the computer.

The valves were brass 1/4" compression x 1/8" MIP (male iron pipe) angle valves with a modified stem taper. The original taper of 90° was turned down to 10°. This modification decreased the sensitivity of flow rate to valve position, enabling a more accurate control of flow rate.

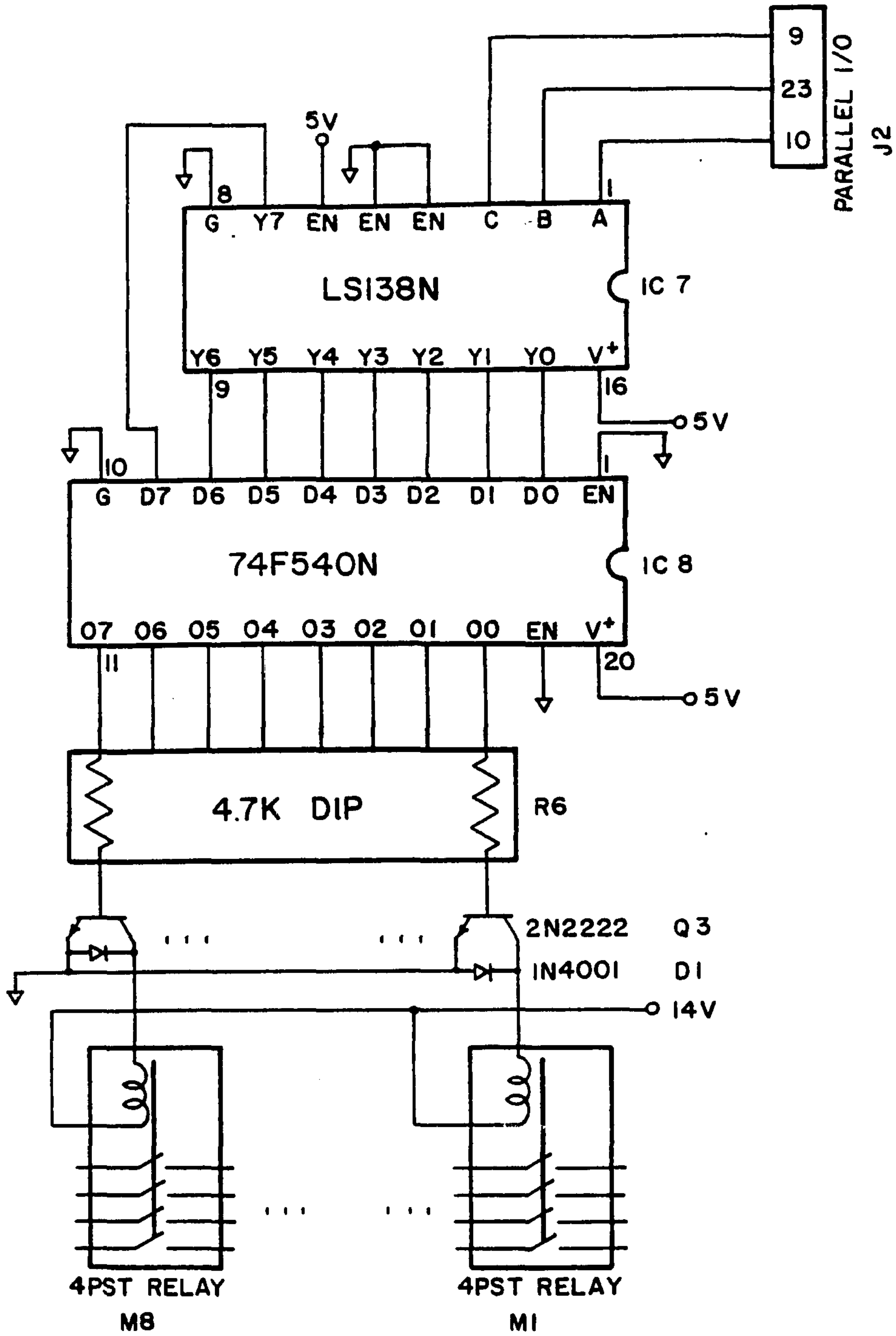


Figure 2-5 Stepper Motor Selection. Three data lines from the Apple computer selected 1 of 8 motors to be active.

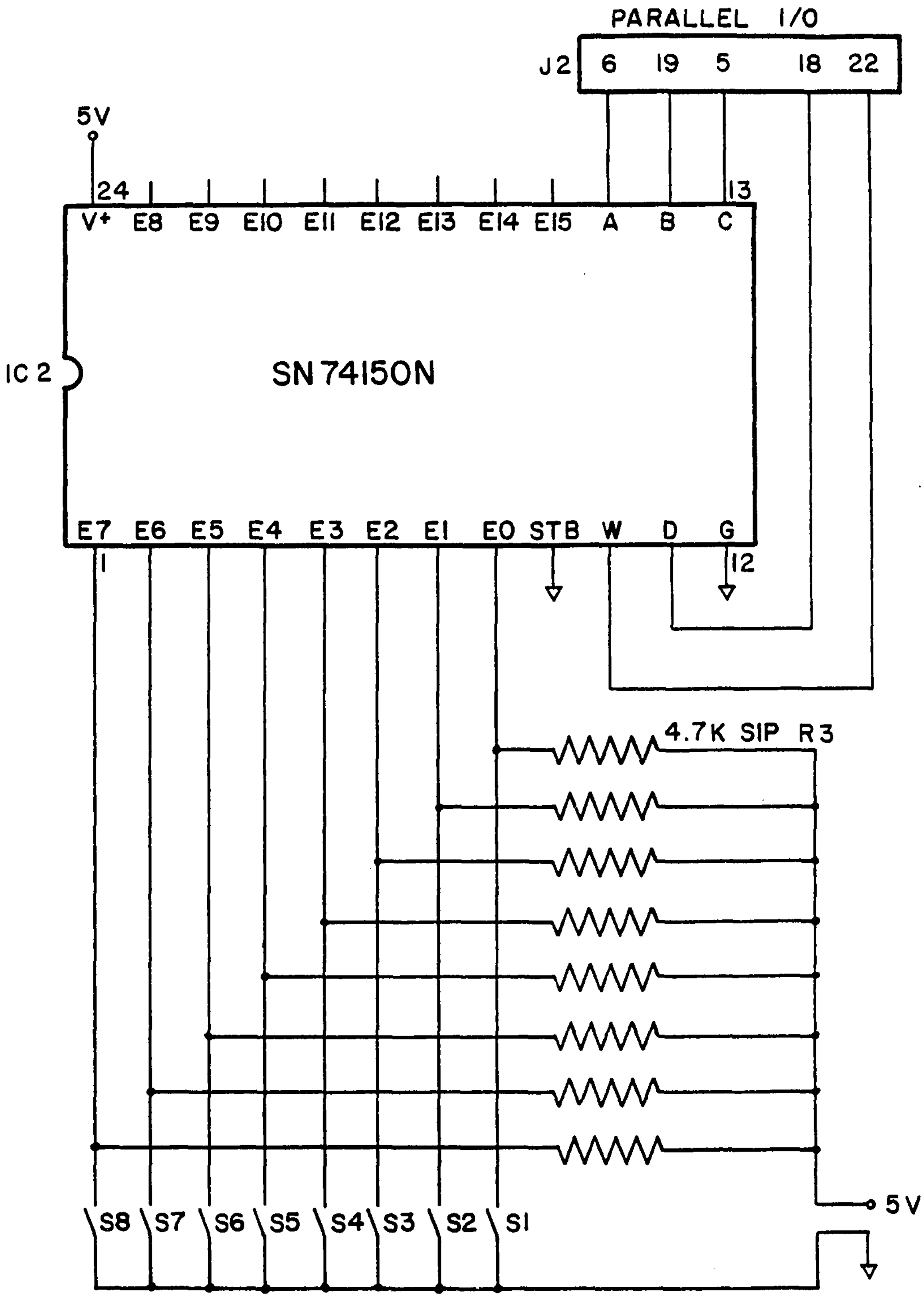


Figure 2-6 Limit Switch Selection. A data selector (controlled by the Apple computer) in the external interface selected 1 of 8 limit switches to be read by the Apple computer.

2.4 Pressure Measurement

The nature of the flow meters (described in the next section) necessitated use of a pressure measurement to obtain flow estimates at low flow rates. This differential pressure was measured across the gear pump with a SenSym SCX15DNC 0-15 psid transducer. The analog output (0.25-9.25V) was applied to the ADC card (channel 2, 0-10V input range) on the Apple computer, as shown in Figure 2-7. The amplifying and scaling electronics were assembled and calibrated by the manufacturer, so the external interface merely provides power to the transducer. The combined nonlinearity and hysteresis errors (provided by the manufacturer) for this transducer amounted to about 1%.

2.5 Flow Measurement

2.5.1 Meter Operation

The flow meter was a Digital Precision meter utilizing a spinning bearingless optical disk. The speed of the disk was proportional to the flow rate. The meter electronics and interface to the ADC are shown in Figure 2-7. The reflected light from the rotating optical disk was sensed by a photo transistor whose output voltage level changed as the intensity of the reflected light changed. The phototransistor was biased with a 10k resistor (R1). A machine code subroutine (see Appendix B) was written to monitor the voltage level at the collector of the phototransistor via the ADC (channels 8-16, 0-

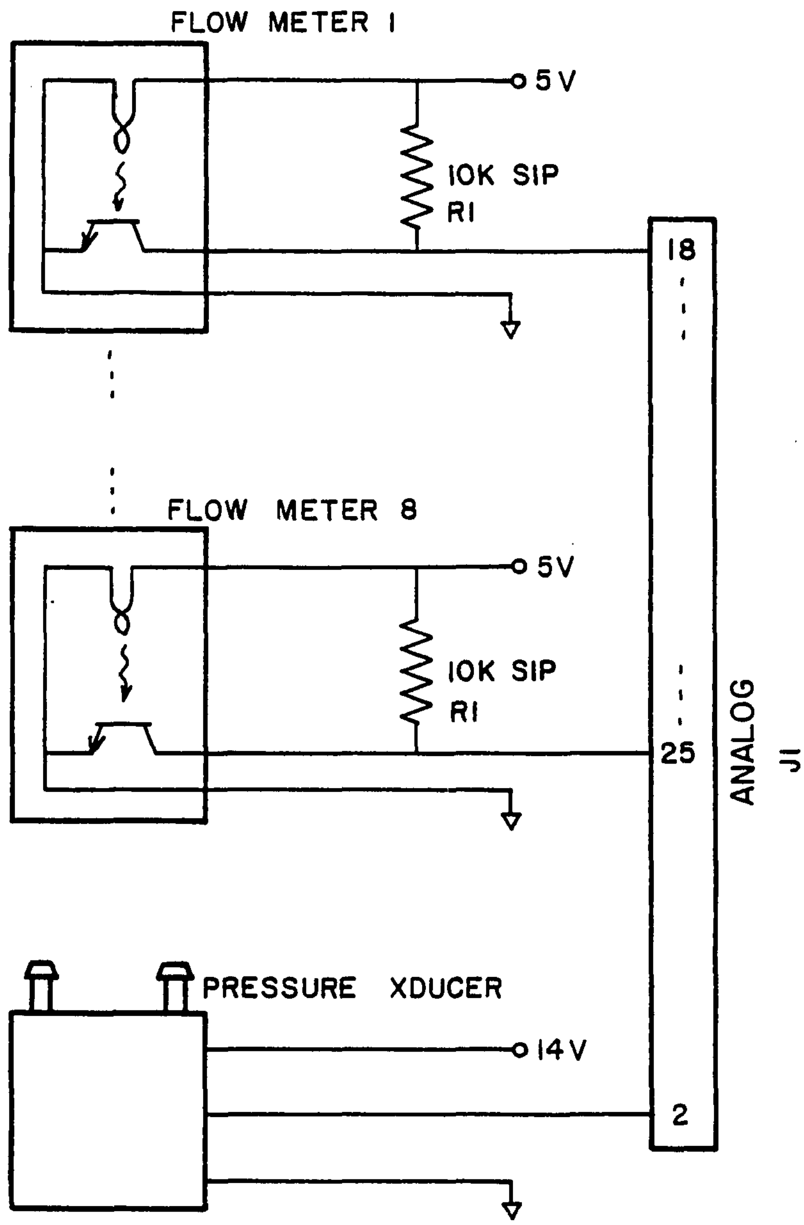


Figure 2-7 Flow Meter and Pressure Transducer Interface. The flow meters output a frequency and the pressure transducer output a voltage, both of which were measured by the ADC in the Apple computer.

10V input range). It measured the frequency output within 1.0 Hz of each meter, simultaneously, averaged over 1 second. Due to slight imperfections in the power supply and the electrical components associated with the flow meters, there was a low level of noise superimposed on the output of the phototransistor. This resulted in a variation of about 10 mV (4 bits out of 12) measured by the ADC, which could cause the frequency counting subroutine to erroneously record a higher-than-actual frequency. As a means of preventing such errors, the frequency counting subroutine implemented a 13% dead band (with respect to the voltage swing of the flow meter output) as shown in Figure 2-8. The phototransistor output must rise above V_H (4.712V) to count as high, but must drop below V_L (4.673) to count as low. Thus, the random noise could not contribute to frequency measurements since it was smaller in amplitude than the dead band.

Note that the use of this flow meter prohibits the use of a roller pump as a prime mover. The temporal variations in flow produced by such pumps yield nonrepeatable flow measurements when the flow rate is averaged over such a short time.

2.5.2 Meter Calibration

The meters were calibrated using a 200ml graduated cylinder and a digital stopwatch. The output from the meter was directed into a container until there was at least 30 ml fluid. During this time, the frequency output from the flow meter was

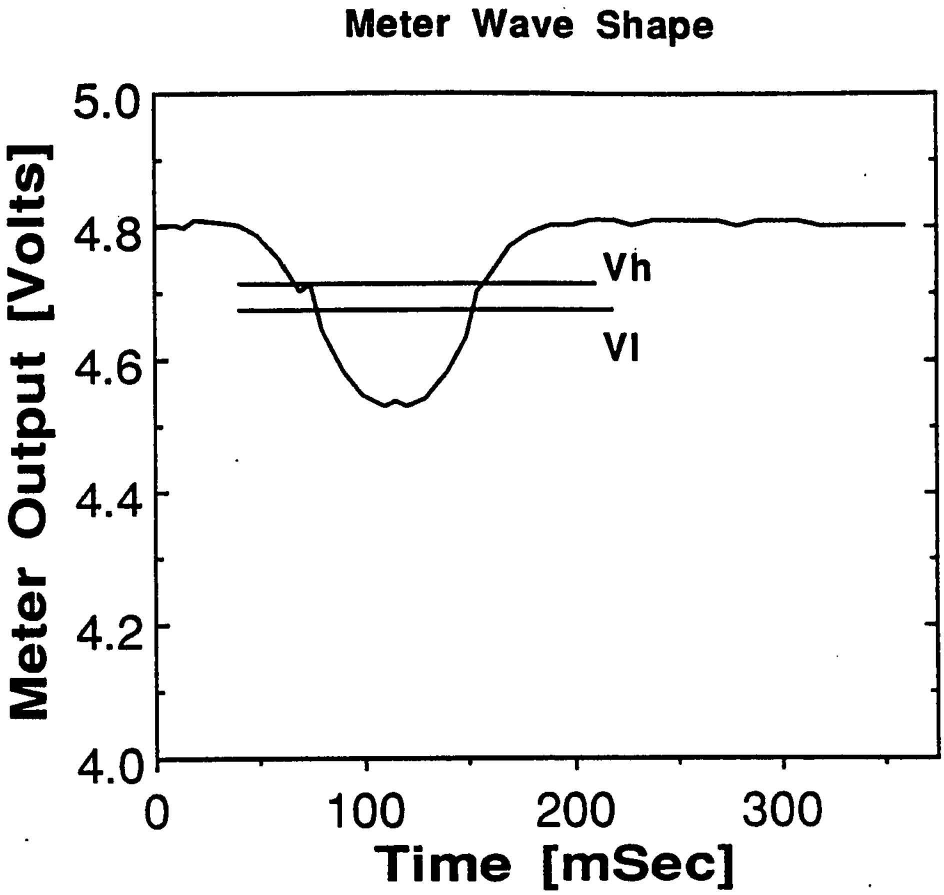


Figure 2-8 Flow Meter Wave Shape. Noise in the meter output was rejected by a 13% deadband performed software in the computer. The limits V_l and V_h correspond to the upper and lower bounds on the deadband. One full period is shown here.

repeatedly measured and averaged (30-60 measurements). The standard deviation of the frequency measurements ranged from 0.13 to 0.54 Hz, while the error in flow rate measurement was 0.5 ml/min. Each meter was calibrated in water and 80% Ethanol from 30 to 200 ml/min. The four meter calibrations in Ethanol are shown in Figure 2-9. The meter output was very linear in the range from 30-200 ml/min, with a maximum nonlinearity of 2 ml/min. The regressions for the four meters in Ethanol were very similar. An average regression was implemented in the computer since over the entire range, the maximum deviation was only 4%. The regression used was:

$$\dot{V} = 1.7f + 28 \tag{2-1}$$

where:

$\dot{V}$ = flow rate of 80% Ethanol [ml/min]

f = meter output [Hz]

2.5.3 Temperature Effects

The meters were calibrated at 21, 32 and 38 °C. Variations in these calibrations can be seen in Figure 2-10. The maximum difference between the hottest and coldest calibration was about 5 ml/min. Since the expected temperature range for the phantom was 25-35 °C, the coefficients obtained at 32 °C were used in the computer program.

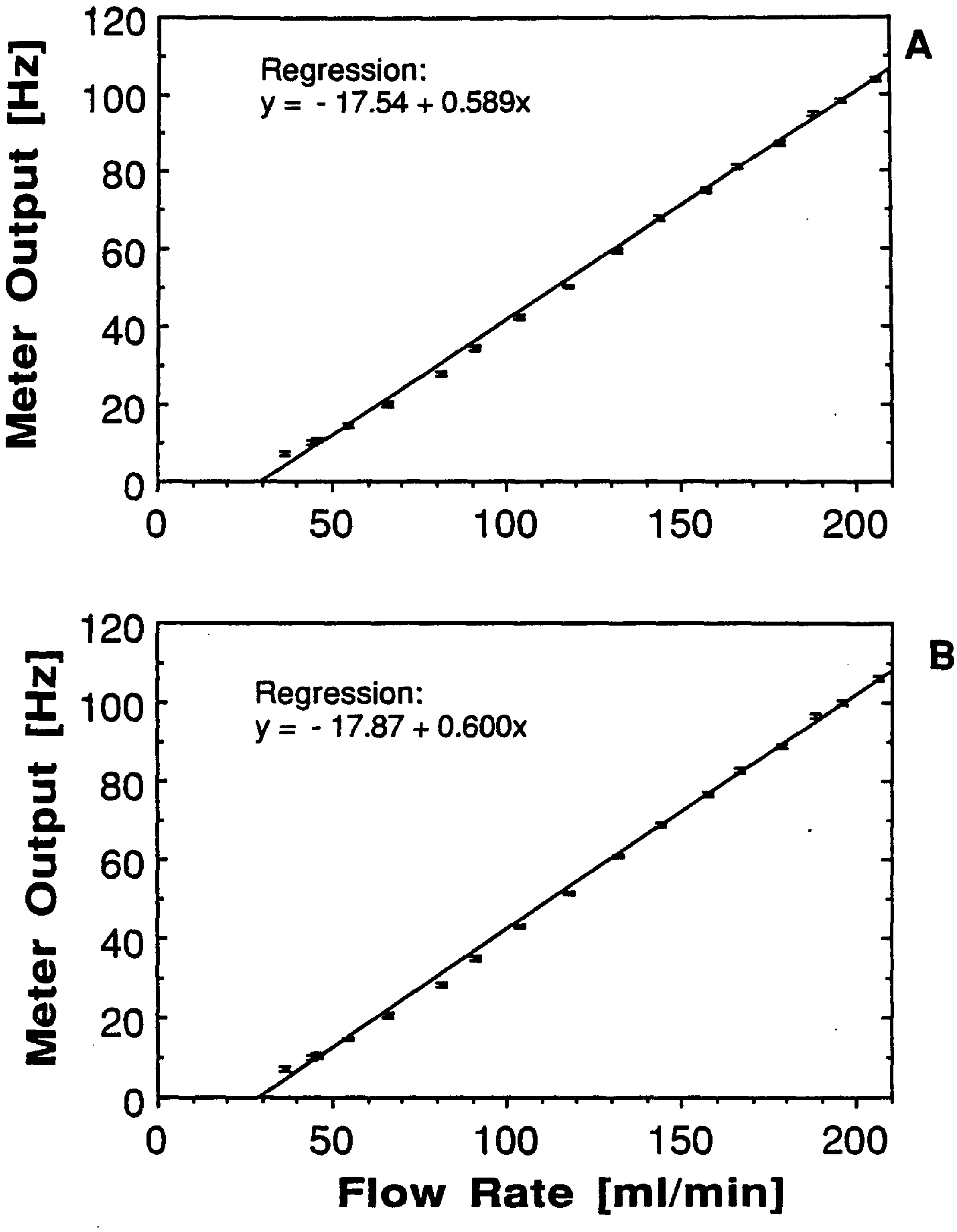


Figure 2-9 Flow Meter Calibration. The frequency output of the flow meter was proportional to the flow rate. A) Meter 1. B) Meter 2.

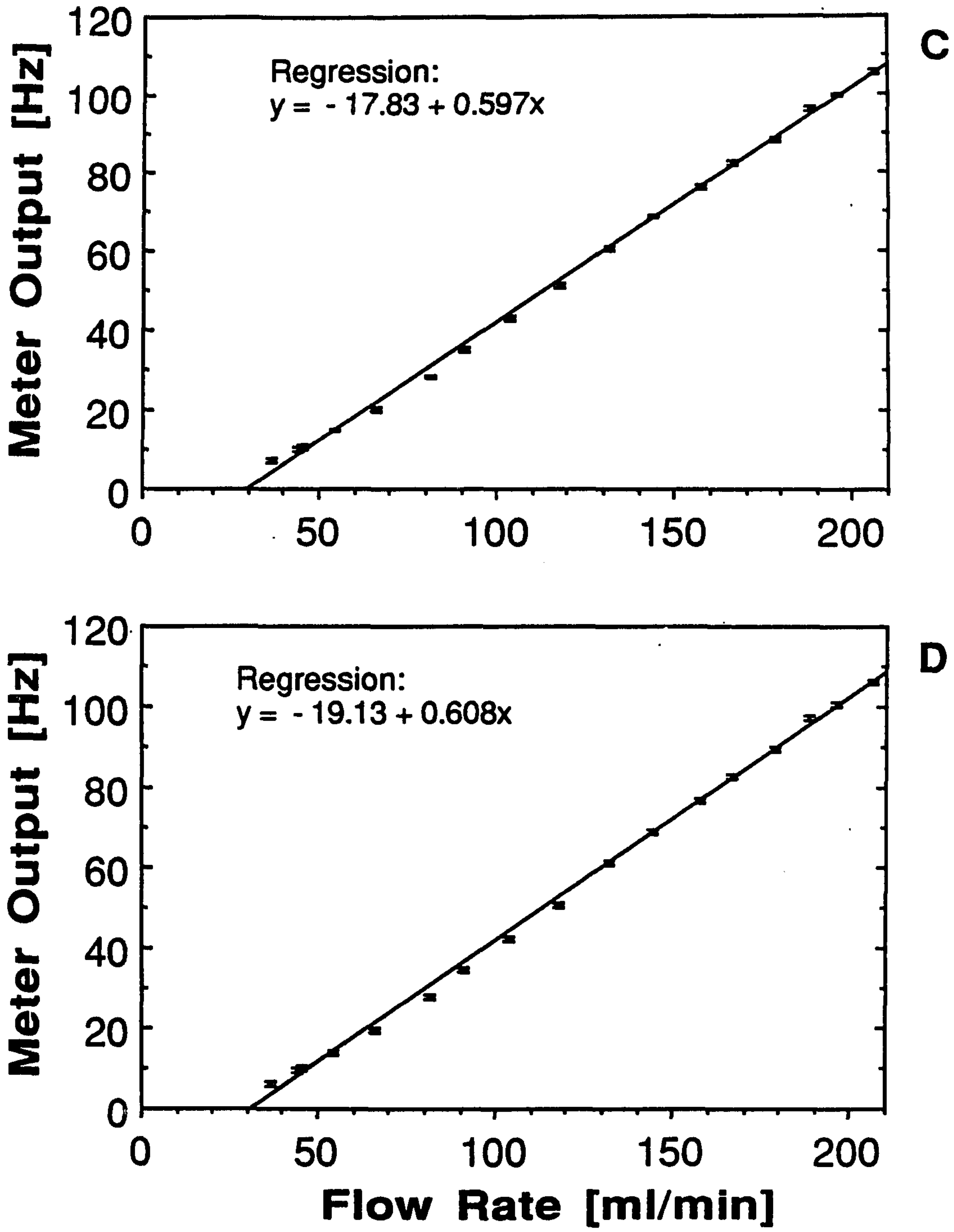


Figure 2-9 Flow Meter Calibration--Continued.
C) Meter 3, D) Meter 4.

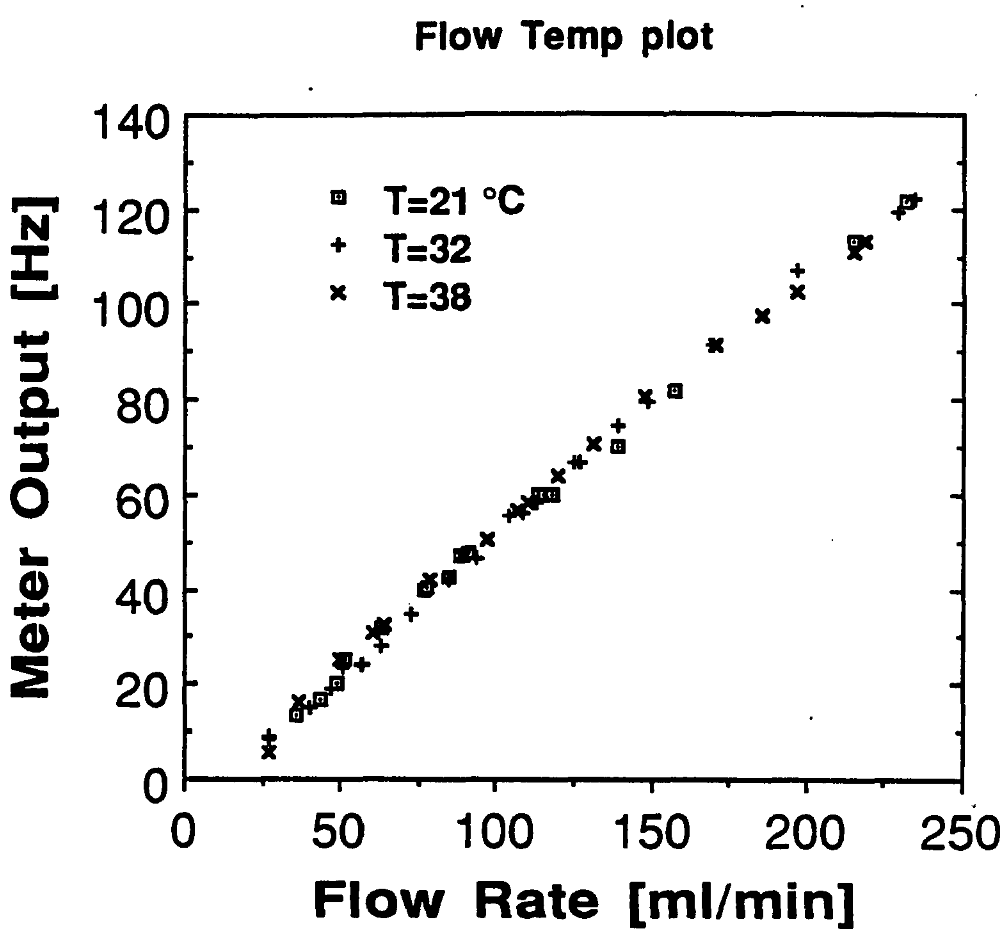


Figure 2-10 Temperature Calibration of Flow Meters. The flow meters were calibrated at 3 temperatures.

2.6 Flow Adjustment

The flow rates to each region were set with one of two algorithms, depending on the rate. At flow rates below 35 ml/min, the optical disk in the meter did not spin evenly and thus yielded erroneous measurements. Therefore, the valves were set based on the pressure rise and valve position in this regime. At flow rates above 35 ml/min, feedback from the flow meters was used.

As the flow rate through any one valve changed, the pressure drop across all the valves also changed. This pressure change naturally caused the flow rate in those valves to change. However, at medium pump speeds (6 to 10 psi pressure rise), the change in flow rate was small, and could be corrected in 1 or 2 iterations of valve position. Table 2-1 summarizes the results of an experiment in which the total number of valve position corrections (for all four valves) was recorded for various changes in target flow rates. Even in the most extreme case tested, where 2 flow rates decreased and 2 flow rates increased, the total number of position corrections was only 6.

2.6.1 Low Flow Rates

The low flow estimate was based on the approximately linear dependence of flow rate with respect to pressure rise and valve position. Figure 2-11A shows flow rate vs. valve position at various pressures, and Figure 2-11B shows flow rate vs. pressure at various valve positions. The linearity of these two

Valve #	Initial Flow Rate [ml/min]	Final Flow Rate [ml/min]	Total Number of Valve Position Corrections (Average of 5 trials)
1	20	60	
2	20	60	4.0
3	20	20	
4	20	20	
1	37	60	
2	37	60	4.8
3	37	37	
4	37	37	
1	20	15	
2	20	15	6.0
3	20	60	
4	20	60	

Table 2-1 Valve Position Correction. The total number of valve position corrections initiated by the computer for various changes in target flow rate.

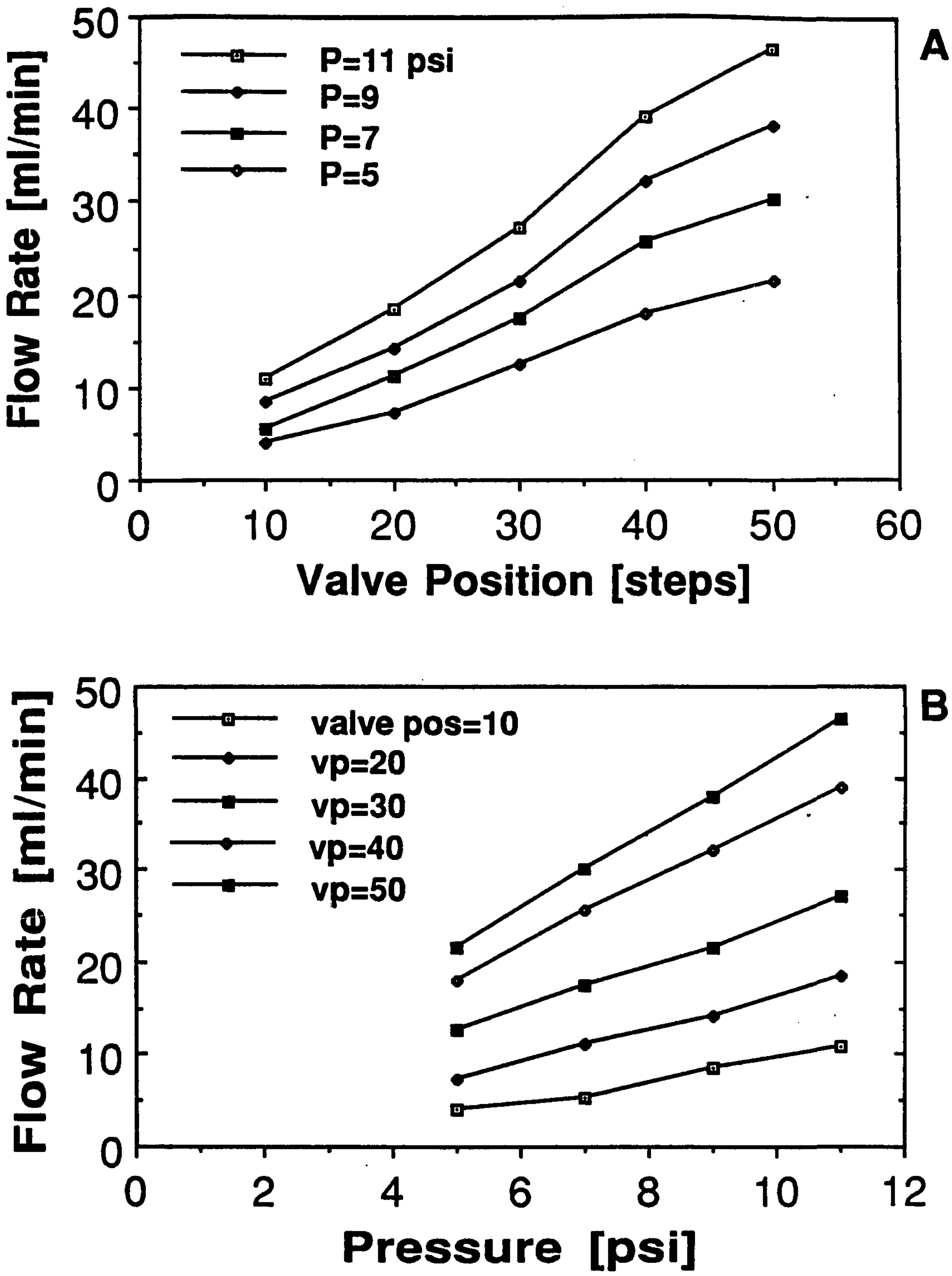


Figure 2-11 Flow Rate at Various Pressures and Valve Positions. A) Flow rate was proportional to valve position at several pressures. B) Flow rate was proportional to pressure at low valve positions.

relations was exploited in the following way to obtain a low flow estimate:

$$\dot{V} = \dot{V}_c\left\{\frac{\theta}{\theta_c}\right\}\left\{\frac{P}{P_c}\right\} \qquad (2\text{-}2)$$

where:

$\dot{V}$ = flow rate [ml/min]
θ = valve position [steps]
P = Pressure [psi]
c indicates a calibration point

Measurements from the flow meters at the lowest measurable rates were used to calculate $\dot{V}_c$, θ_c, and P_c. These parameters formed a calibration point and were updated using the flow rate from equation 2-1, the measured pressure and the current valve position whenever the flow rate was in the range from 30 to 40 ml/min. This compensated for any possible short-term changes in vascular resistance due to heating. Figure 2-12 illustrates this low flow estimation technique. The parameters in equation 2-2 were calibrated at 38 ml/min and 9 psi with a valve position of 50 steps. The flow rate estimates (from equation 2-2) match the actual flow rates to within 2 ml/min.

2.6.2 High Flow Rates

At high flow rates, the flow meters were used directly to set the valve positions. First, an estimate was used to provide an initial guess for the valve position. Then feedback from the flow meters (via equation 2-1) was used in a proportional algorithm to obtain the specified flow rate. The approximate

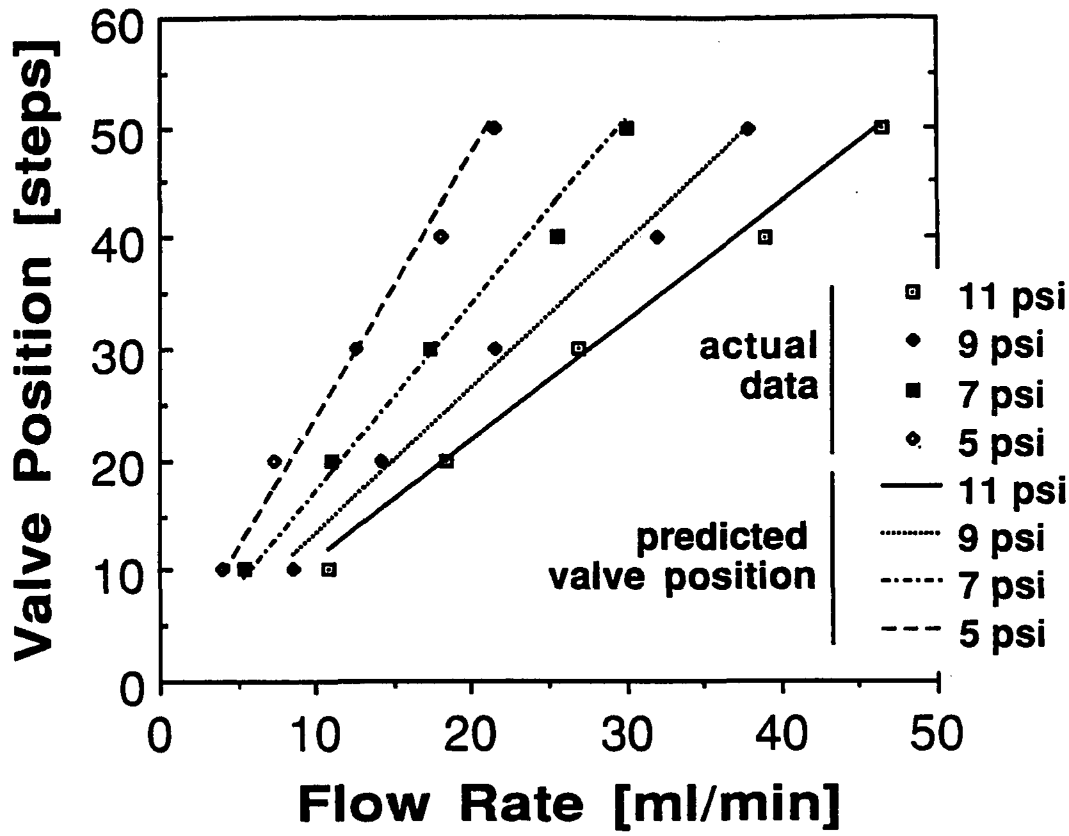


Figure 2-12 Low Flow Rate Prediction. Predicted and actual valve positions required for various target flow rates and pressures using a linear estimate based on a single calibration point at 38 ml/min and 9 psi with the valve at 50 steps.

empirical relation used for the initial estimate of valve position was of the form:

$$\theta = a + by + cy^2 + dy^3 \tag{2-3}$$

where:

$$y = \left\{\frac{\mathring{V}_t}{P} - 0.2P\right\}e$$

$\mathring{V}_t$ = target flow

e = correction factor

The four constants a, b, c, and d were coefficients which were different for each valve, and empirically determined. An example of this is shown in Figure 2-13 for flow meter number two. The cubic approximation is close to the actual data, but not exact. The correction factor, e, was used as an iterative, adaptative mechanism to improve the estimate of valve position since the cubic estimate was not accurate at all pressures, and since the preserved kidneys degrade with time. Thus, even if the vascular resistance changed, the valve position estimate would remain accurate. The valve adjusting process ran as follows. Initially, the factor e was assumed to be unity. Equation 2-3 was used to calculate a valve position estimate for the specified target flow rate. After the valve was moved to the calculated position, the flow rate was measured and used to correct e such that equation 2-3 would be exact at that combination of 4 valve positions and 4 measured flow rates. This was done by finding the root of equation 2-3 which has exactly one real root as follows:

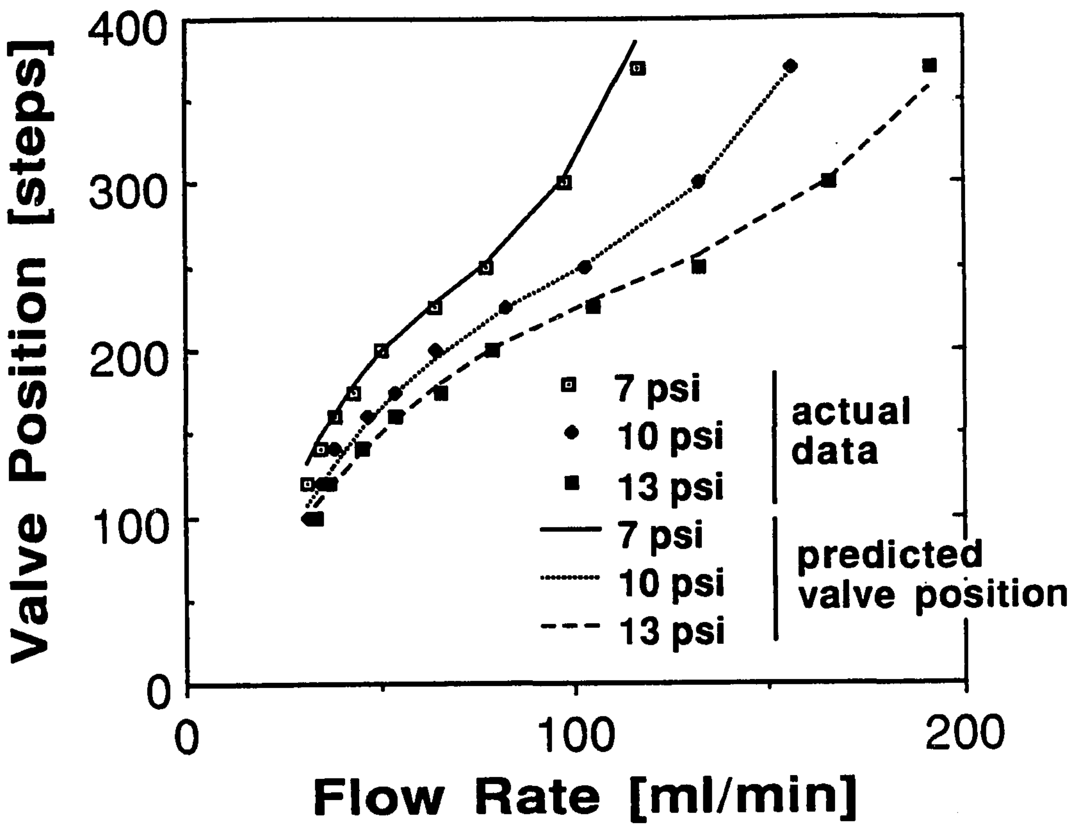


Figure 2-13 High Flow Estimate. A cubic equation involving target flow rate and pressure provided an initial estimate for the valve position prior to use of feedback when attaining target flow rates. This estimate and the actual valve positions are compared at 7, 10 and 13 psi.

$$e = \frac{\left\{-2K_3\sinh\left[\frac{\sinh^{-1}(x)}{3}\right] - c\right\}}{3d\left\{\frac{\dot{V}}{P} - .2P\right\}}$$

where:

$$x = \left\{\frac{K_1+K_2(a-\theta)}{2K_3^3}\right\}$$

$$K_1 = 2c^3 - 9cdb$$

$$K_2 = 27d^2$$

$$K_3 = \sqrt{3db - c^2}$$

This new value of e was used together with the target flow rate to find a new estimate for valve position using Equation 2-3. The flow rate was again measured and the process iterated until the actual flow rate was within 10 ml/min of the target (usually 1 or 2 iterations). At this point, a simple proportional correction was used to adjust the valve position until the flow rate was within 2 ml/min of the target flow rate. This iterative correction of the factor e yielded very accurate estimates for valve position at the next change in flow rate.

2.6.3 Error In Flow Rate Measurement

The theoretical error in measuring the flow rate was divided into two regions corresponding to the two measurement techniques previously described. At flow rates above 35 ml/min, the error, derived from equation 2-1 (Dally 1984), was simply:

$$\delta\dot{V} = 1.7f\sqrt{\left(\frac{\delta 1.7}{1.7}\right)^2 + \left(\frac{\delta f}{f}\right)^2} + 28\left(\frac{\delta 28}{28}\right)$$

where:

$\delta 1.7 = 0.03$ error in multiplier from equation 2-1 for the 4 meters used

$\delta f = 1$ error of frequency measurement as per section 2.5.1

$\delta 28 = 1.2$ error in constant term from Equation 2-1 for the 4 meters used

This yielded $\delta\mathring{V}$ = 4.9 ml/min for the worst case. For flow rates below 35 ml/min, the various errors in equation 2-2 combined to yield:

$$\frac{\delta\mathring{V}}{\mathring{V}} = \sqrt{\left[\frac{\delta\mathring{V}_c}{\mathring{V}_c}\right]^2 + \left[\frac{\delta\theta}{\theta}\right]^2 + \left[\frac{\delta\theta_c}{\theta_c}\right]^2 + \left[\frac{\delta P}{P}\right]^2 + \left[\frac{\delta P_c}{P_c}\right]^2}$$

where:

$\delta\mathring{V}_c$ = 2.9 = error in measured flow at calibration pt.

$\delta\theta = \delta\theta_c$ = 2 = error in valve position

$\delta P/P = \delta P_c/P_c$ = 0.01 = relative error in pressure measurement

Using these numbers, the largest error (in the low flow range) was 3.5 ml/min at $\mathring{V}$ = 35 ml/min, decreasing to 1.2 ml/min at $\mathring{V}$ = 5 ml/min.

The worst case errors over the entire flow range are shown in Figure 2-14. The discontinuity at $\mathring{V}$ = 35 ml/min is due to the switch in measurement techniques from equation 2-1 to equation 2-2. The highest error is 4.1 ml/min at $\mathring{V}$ = 160 ml/min (the expected upper limit of measurement).

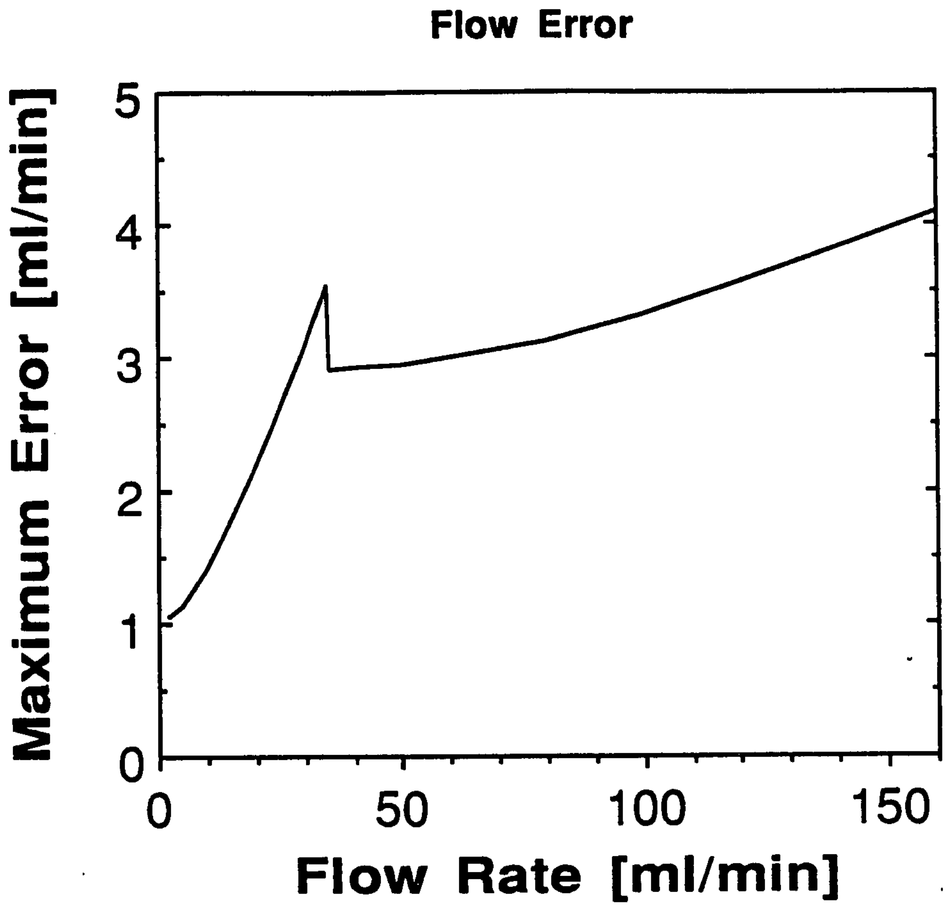


Figure 2-14 Maximum Flow Rate Error. The error in flow rate measurement over the range of expected flow rates. Due to different measurement techniques, the error was discontinuous at 35 ml/min.

2.7 Temperature Measurement

The temperature in the phantom was measured by the Apple computer with 16 bare-wire manganin-constantan welded junction thermocouples, denoted 'flow control thermocouples'. The following equation describes the conversion from a voltage and reference temperature to the actual junction temperature. The constants used are those for copper-constantan thermocouples, which are essentially identical to manganin-constantan thermocouples (Hynynen, et al. 1987).

$$A = V_{tc} + a_1 + T_r(a_2 + T_r(a_3 + a_4 T_r)) - b_4$$
$$T_{tc} = b_5 + A(b_1 + A(b_2 + b_3 A)) \qquad (2\text{-}4)$$

where:

V_{tc} = thermocouple voltage
T_r = reference temperature
T_{tc} = thermocouple junction temperature

$$\left.\begin{array}{l} a_1 = 1.782E-7 \\ a_2 = 3.866E-5 \\ a_3 = 3.990E-8 \\ a_4 = 1.950E-11 \\ b_1 = 2.587E+4 \\ b_2 = -6.649E+5 \\ b_3 = 2.229E+7 \\ b_4 = -5.069E-5 \\ b_5 = -1.2942 \end{array}\right\} \text{Empirical constants for Copper-Constantan thermocouples}$$

These equations are valid for a range larger than typical use:

$0 < T_{tc} < 80\ ^\circ C, \quad 0 < T_r < 40\ ^\circ C$

The 16 thermocouples (4 in each Kidney) were referenced to an AD590JH integrated circuit temperature sensor. The thermocouples were connected to the ADC in the Apple computer through an analog multiplexer and an instrumentation amplifier as shown in Figure 2-15. The Manganin side of each thermocouple was connected to an input of the MUX-16 multiplexer (IC3) where one of 16 channels was selected by 4 logic lines from the Apple computer. The output from the multiplexer was connected to the positive input of the AMP-01 instrumentation amplifier (IC5). The negative input of the amplifier was connected to the common constantan side of all the thermocouples. Two resistors associated with the amplifier, R_g and R_s, formed a gain of 1000. This amplified the thermocouple signal to a range which could be accurately measured by the ADC (channel 0, 0-1V input range). The AD590JH (IC1) was a "constant current" temperature sensor, i.e. the current through the device was proportional to it's temperature in Kelvin. This current was directed through a 1k, 1% resistor (R2) so that the ADC (channel 1, 0-0.5V input range) could simply measure the voltage drop due to the current.

The linearity of the A/D electronics and instrumentation amplifier were investigated to insure proper voltage measurement of the thermocouples. A voltage source, monitored with a volt meter was connected to thermocouple input #1 on the external interface. The applied voltage was compared with the output of the A/D converter over a range from 0 to 1 Volt DC. The resulting relation was very linear (R^2=.9991, using a least squares fit) of the form:

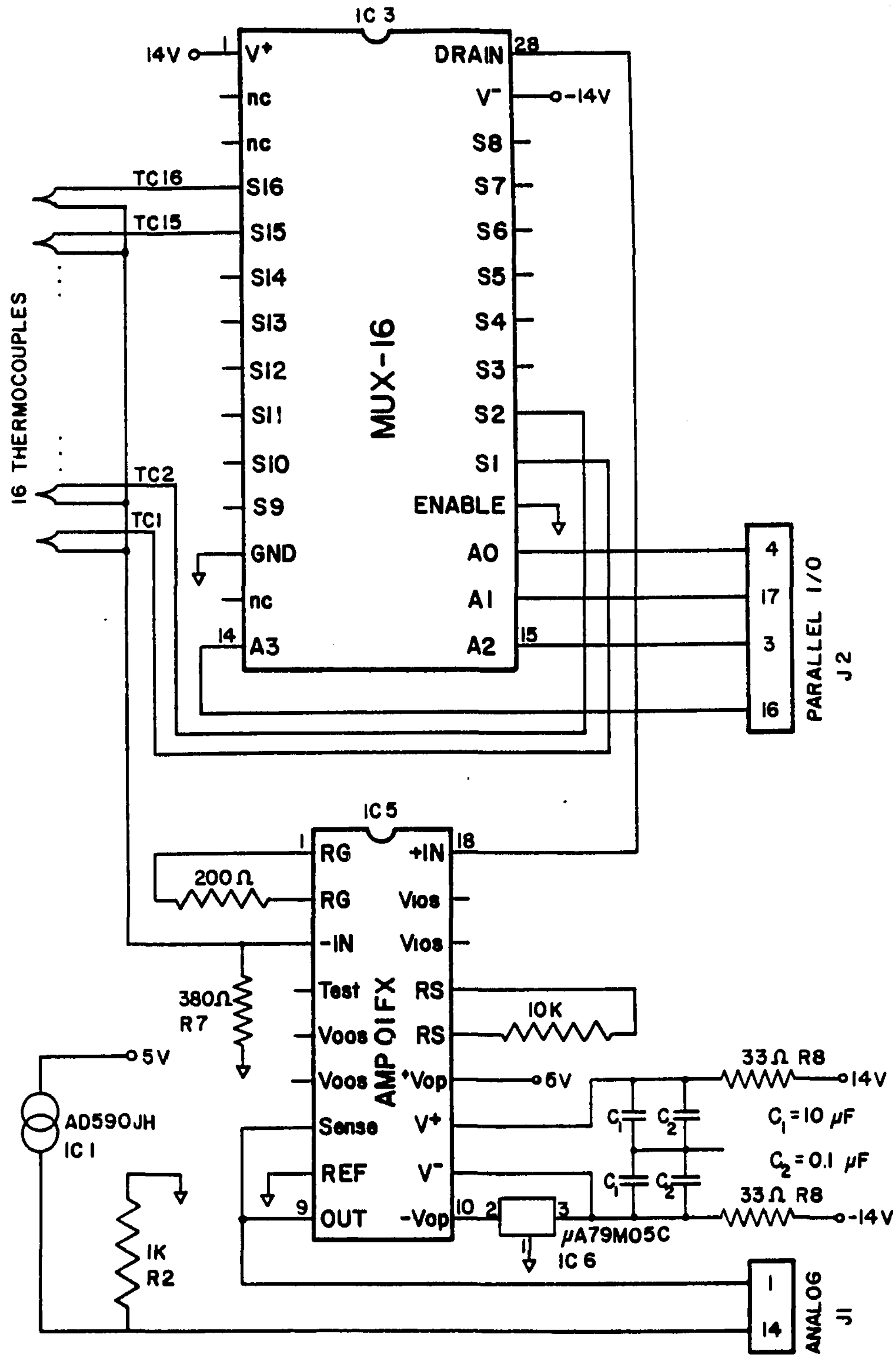


Figure 2-15 Temperature Measurement System. The Apple computer measured 16 thermocouples using an analog multiplexer and an instrumentation amplifier. An integrated circuit temperature sensor was used as a reference temperature.

$$V_{actual} = mV_{ADC} + b$$
$$m = 0.8583$$
$$b = -.0062$$

This voltage correction was used with equation 2-4 to calculate the actual thermocouple temperature. The flow control thermocouples were calibrated with a Fisher Scientific mercury thermometer (±0.5°C) at 8 temperatures. After offset corrections, the thermocouples were accurate to within 0.25°C.

The hyperthermia thermocouples consisted of 2 each 7-junction silica enclosed probes inserted in every kidney. These were terminated in 15 pin sub-D connectors which interfaced with the hyperthermia controller (HP 9836). These thermocouples were positioned in the phantom as shown in Figure 2-16. The perfusion regions are numbered 1 through 4 and the tick marks indicate thermocouple locations. The thermocouple location method is discussed in section 3.1.

2.8 Computer Code

A Basic program was written for the Apple Computer to accept user input and to communicate with the various sensors. This program listing can be found in Appendix A. Machine code subroutines were written, as mentioned in previous sections, to drive the stepper motors, measure the flow rate, multiplex the thermocouples, and perform timing operations. These subroutines are given in Appendix B.

When the Basic program, called 'FOUR VALVE', is run, the

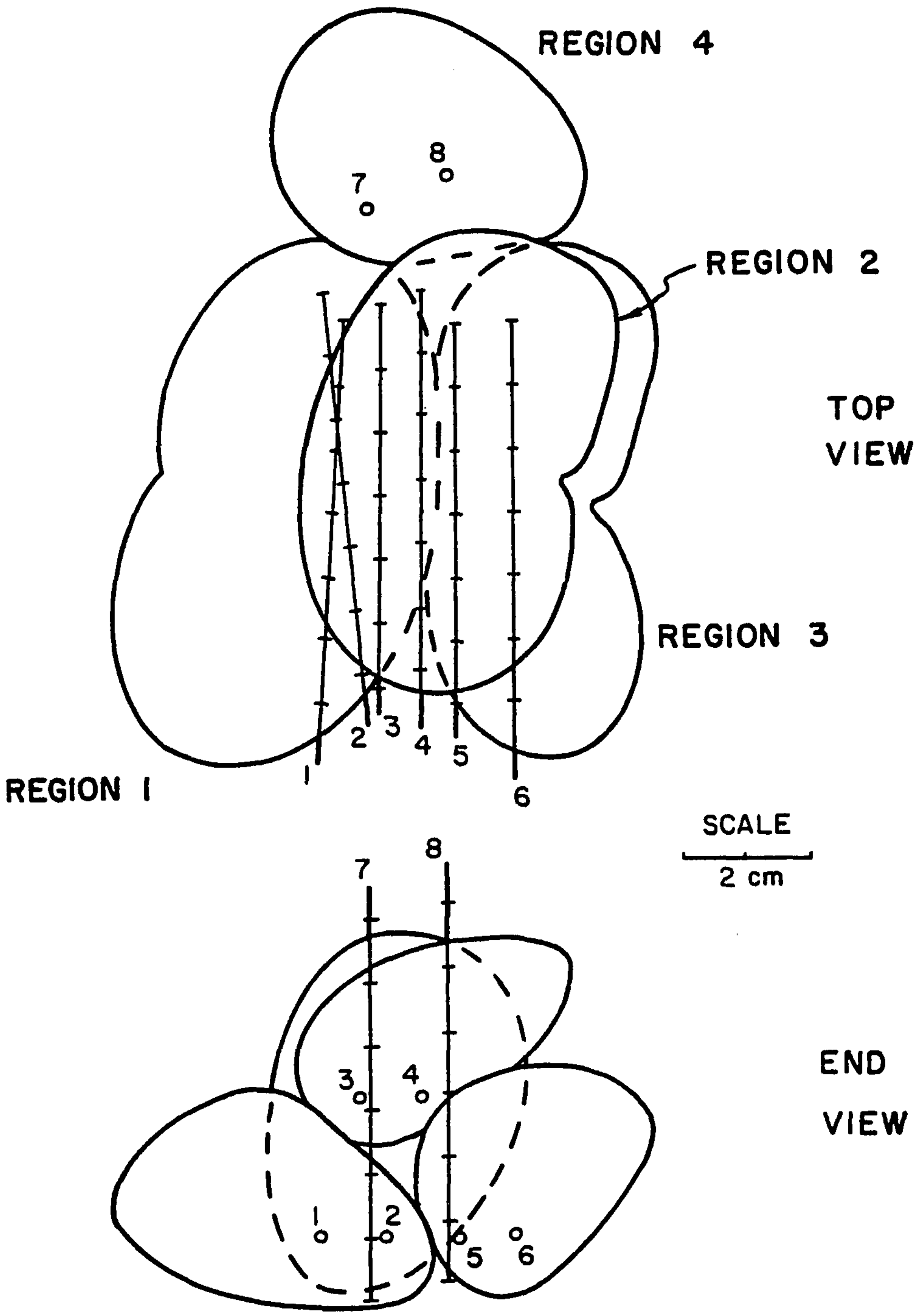


Figure 2-16 Kidney and Thermocouple Placement. The 4 kidneys (regions 1-4) were positioned as shown. The thermocouple probes numbered 1-8 interfaced with the hyperthermia controller.

machine code subroutines are automatically loaded into the appropriate locations, and the user is prompted for the number of valves to control. After this entry, the main program menu is displayed. The program has three menus, as shown in Figure 2-17. These menus, described in the following sections, are the main menu, the mode menu, and the interactive menu. In order to clarify the use of the program, the upper line of the monitor continually displays the menu name or task being performed.

2.8.1 Main Menu

The main menu of the program presents the following choices:

1) ENTER NEW DATA
2) LOAD DATA FROM DISK
3) SAVE DATA TO DISK
4) LIST DATA
5) SELECT CONTROL MODE
6) END

ENTER NEW DATA: Flow rates for each region are sequentially entered with a time duration at that combination of flow rates. This process is continued for additional flow rates and durations until a 'Q' is entered in response to the time duration input. Then the data entry section is terminated and the main menu is again displayed. If there is any data already in memory when option 1 is selected, the new data is simply appended to the old.

LOAD DATA FROM DISK: The user is prompted for a name. That

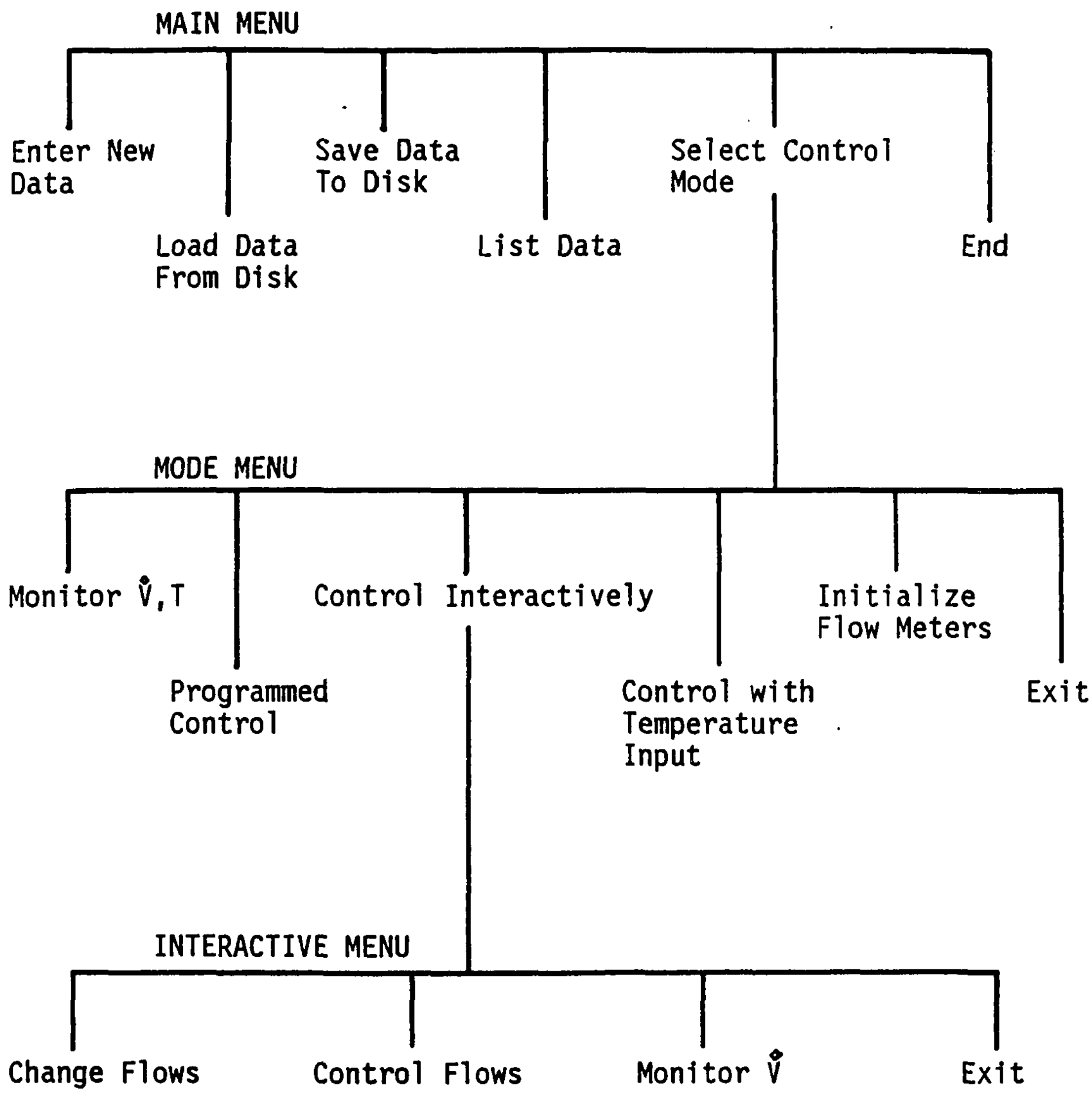


Figure 2-17 Program Options. The various options on the 3 menus of the phantom control program.

file is loaded from the default drive, and must be a file written by the 'SAVE DATA' option. The number of valves to control is set at the number used when the file was written.

SAVE DATA TO DISK: The entire sequence of flow rates and durations in memory is saved on the disk in the default drive under a name specified by the user.

LIST DATA: The sequential list of flow rates and durations is output to the screen. Upon completion, the user may choose to make a hard copy of this information, or to return to the main menu.

SELECT CONTROL MODE: Initializes the stepper motors controlling the valves and presents the mode menu (below).

END: Terminates the FOUR VALVE program, resetting the text window to it's default configuration.

2.8.2 Mode Menu

Under this menu, the mode of flow rate control may be specified. The choices here are:

1) MONITOR V,T
2) PROGRAMMED CONTROL
3) CONTROL INTERACTIVELY
4) CONTROL WITH TEMPERATURE INPUT
5) INITIALIZE FLOW METERS
6) EXIT

MONITOR V,T: The flow rate and temperature of each region are displayed and continually updated at a rate of about 1.5

seconds. The 'ESC' key terminates this mode and returns to the mode menu.

PROGRAMMED CONTROL: The sequence of flow rates and time durations in memory is executed. For each step, the flow rates are set, then the elapsed time is displayed until the next step. The timer may be terminated by pressing 'CTRL-A', in which case the next step is immediately begun.

INTERACTIVE CONTROL: This initiates the Interactive Menu, described below.

TEMPERATURE CONTROL: This mode controls the flow rates based on the temperature of the corresponding region. For each region, three items must be entered: The basal flow rate (flow rate at zero temperature elevation), the flow gain (increase in flow per °C), and the time delay. The time delay is the elapsed time between a temperature rise and the flow increase. These three factors form an algorithm for setting the flow rates as follows:

$$\dot{V}(t) = \dot{V}_b + G\Delta T(t-t_d) \qquad (2\text{-}5)$$

where:

$\dot{V}$ = flow rate [ml/min]
$\dot{V}_b$ = basal flow rate [ml/min]
G = flow gain [ml/min per °C]
ΔT = Temperature elevation [°C]
t = current time [seconds]
t_d = time delay [seconds]

When all the parameters have been entered, the computer sets the basal flow rates, measures the initial temperature and waits for the user to select for a hard copy of the results or none. If hard copy mode is selected, the flow rate and highest temperature of each perfusion region are printed at every measurement time. As soon as this selection is made, the control algorithm is activated, the temperature is measured and flow rates updated at about 20 second intervals. The 'ESC' key terminates the temperature control mode and returns to the mode menu.

INITIALIZE FLOW METERS: This option must be performed prior to any attempt to control flow rates, and after any change in pump speed. For each valve, the flow rate is measured at two valve positions. These values are used to estimate the flow at rates below those measurable by the flow meter, and to predict the required valve position at high flow rates. During this process, the flow rates, valve positions and pressure drop are displayed on the screen. After long periods of disuse, the flow meters may stick a little, causing zero flow readings. This can be remedied by tapping on the meter housing with a finger or eraser, while the flow rate is greater than 35 ml/min.

EXIT: Returns to the main menu.

2.8.3 Interactive Menu

This section allows the user to continually control or monitor the flow rates to each perfusion region. The following menu is provided:

1) CHANGE FLOWS
2) CONTROL FLOWS
3) MONITOR V
4) EXIT

CHANGE FLOWS: Target flow rates are specified for each perfusion region in this section. The entry is made by typing the valve number and the target flow rate, separated by a comma.

CONTROL FLOWS: The flow rates are continually monitored and adjusted by the computer to achieve the target flow rates. The 'ESC' key terminates computer control of the flow rates.

MONITOR V: The computer continually displays flow rates and pressure drop. Temperatures displayed here are are not updated, and simply set to zero. The 'ESC' key exits this subroutine.

EXIT: Returns to the Mode Menu.

2.9 Perfusion Pattern For An In Vitro Kidney

In order to form an estimate of the spatial variations in perfusion in the phantom, a series of perfusion measurements were made using a Thermal Technologies Thermal Diffusion Probe. This instrument measures k_{eff} (effective thermal conductivity) in a localized region (Valvano and Patel 1986). Knowledge of these variations is important since there is at least a 10:1 difference in perfusion between the cortex and the medulla of an alcohol fixed canine kidney.

2.9.1 Procedure For Flow Pattern Measurement

A gear pump was used to circulate 80% ethanol through an in vitro kidney, as shown in Figure 2-18. A regulated water bath was used to stabilize the kidney temperature. The ethanol temperature was regulated (before entering the kidney) with a coiled copper heat exchanger submerged in the water bath. To make a 'pullback measurement', the Thermal Diffusion Probe was inserted into the kidney using 16 Ga catheter. Next, the catheter was pulled back, leaving the exposed probe fully inserted in the kidney. The probe base was attached to a stage with micrometer movement, and k_{eff} measurements were performed as the probe was pulled back in 1 mm steps. Three pullback studies were performed in one kidney, with 2 to 5 measurements taken at each position and flow rate, depending on the variance of the first two measurements. Measurements of k_{eff} at zero flow were recorded to estimate the intrinsic conductivity of the medium.

2.9.2 Results For Flow Pattern Measurement

The average intrinsic thermal conductivity (k_{eff} at zero flow) of all locations was about 2.48 mW/cm °K, which was very close to the thermal conductivity of 80% Ethanol (2.54 mW/cm °K, CRC Handbook of Chemistry and Physics 1985). This small

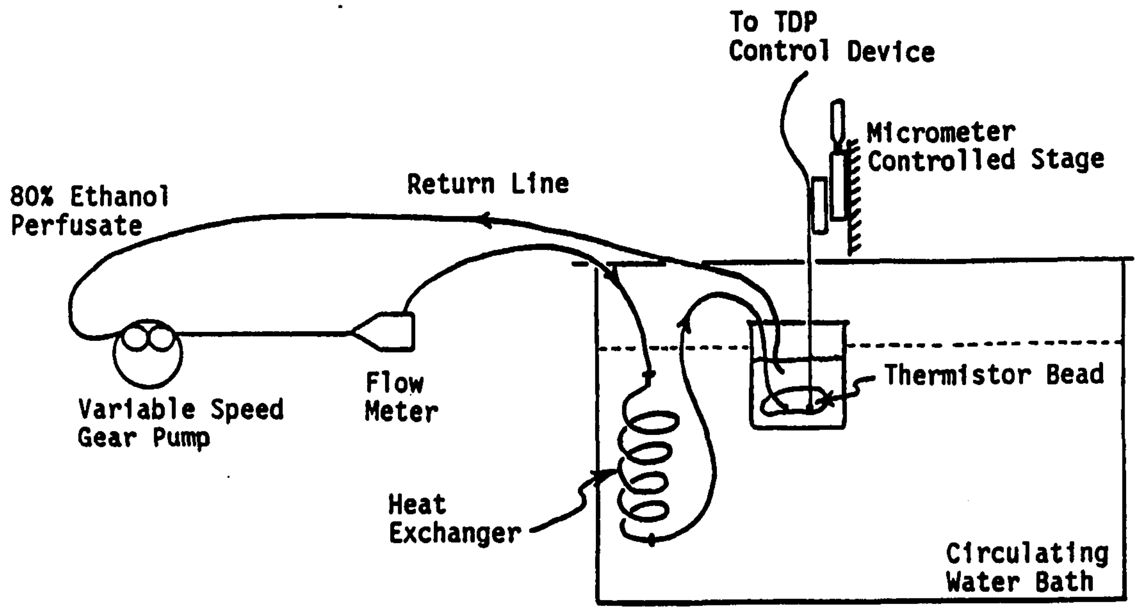


Figure 2-18 Experimental Apparatus for Thermal Conductivity Measurements. 80% Ethanol was pumped through a heat exchanger into the kidney. The thermistor bead, used to measure the thermal conductivity, was pulled back with a micrometer.

difference could easily have been due to the conductivity of the phantom material itself. A plot of K_{eff} versus distance along the line of measurement is shown in Figure 2-19A with the flow rate at zero, 18, 54 and 78 ml/min. Distance equal to zero corresponds to the first measurement taken when the probe was fully inserted. At 78 ml/min, the first peak is K_{eff}=14 mW/cm °K located 4 mm from the edge. The last peak is 1 mm from the far edge with a magnitude K_{eff}=28.5 mW/cm °K. This large discrepancy in magnitudes indicates that the leakage of fluid past the probe through the entrance hole may have had a significant effect near that edge (this effect was also notice by Kress, pg145 1988). For this reason, only the data for the first half of the pullback was used.

The change in k_{eff} due to perfusion, denoted Δk_{eff}, was calculated by subtracting the intrinsic value, k from the measured effective value, k_{eff}. Figure 2-19B shows Δk_{eff} vs. distance for flow rates of 18, 54 and 78 ml/min. Figure 2-20 shows k_{eff} vs $\dot{V}$ for 6 locations at various flow rates. The linearity of this plot indicates that if Δk_{eff} and $\dot{V}$ are known at a given position, then Δk_{eff} can be estimated at other flow rates at that point using a linear approximation.

2.9.3 Perfusion Estimate

The effective thermal conductivity data obtained in the previous section was used to estimate the actual perfusion at each measurement location. This was done by assuming a linear

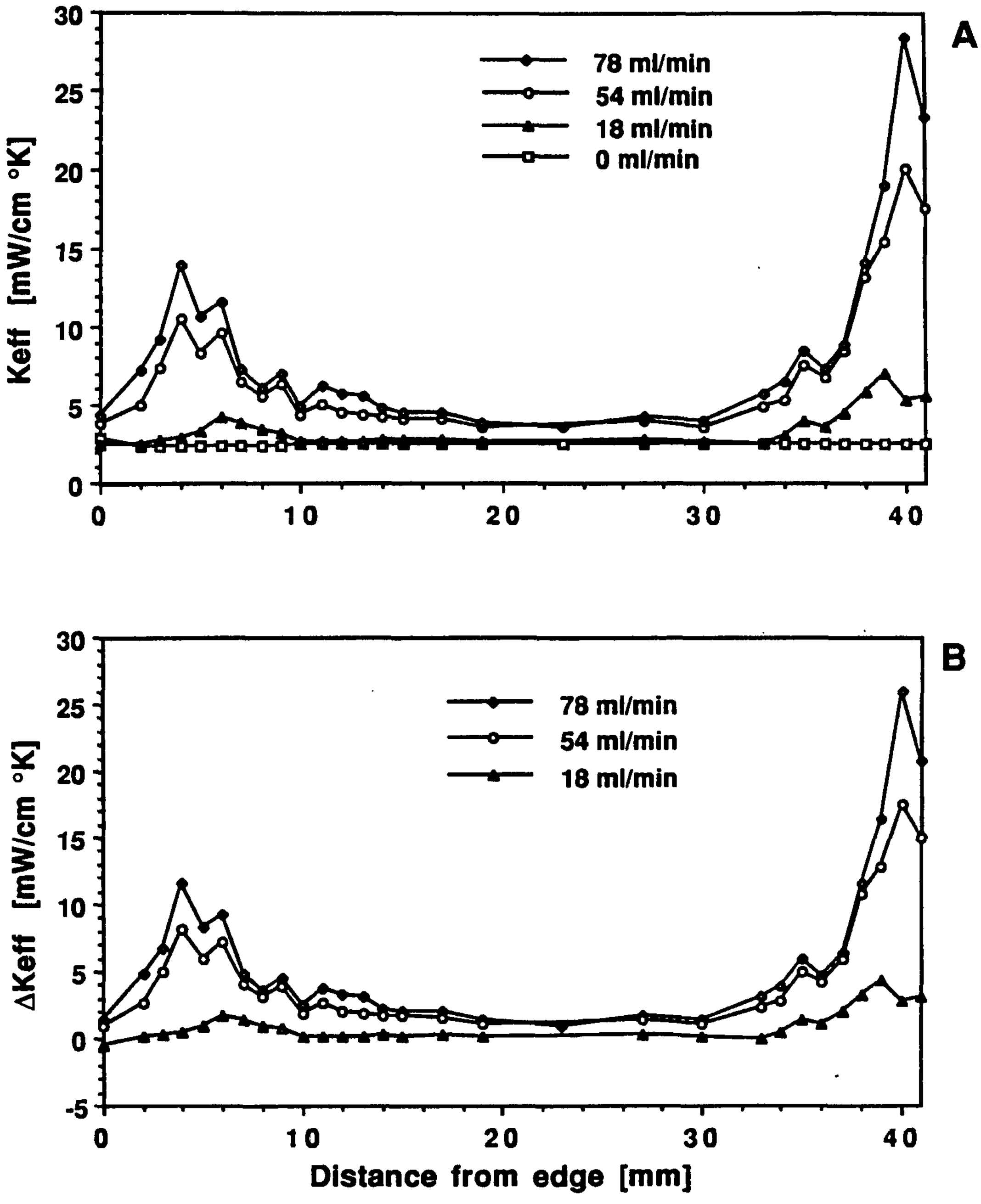


Figure 2-19 Thermal Conductivity Mapping. A) K_{eff} at 4 flow rates. B) ΔK_{eff} at 3 flow rates. The pullback distance corresponds to two opposite edges of an <u>in vitro</u> kidney.

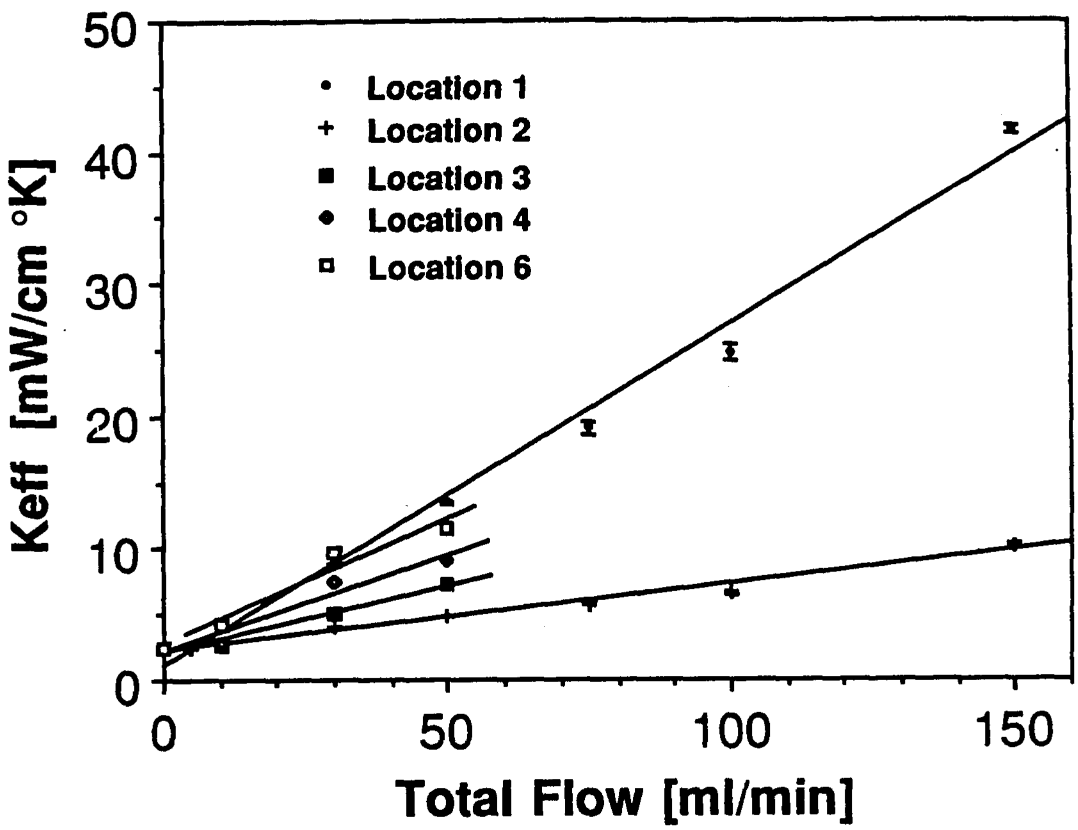


Figure 2-20 Effect of Flow Rate on Effective Thermal Conductivity. k_{eff} increased linearly with total organ flow rate at all positions measured.

dependence of perfusion with changes in thermal conductivity (Valvano and Patel 1986) as follows:

$$W = K_w \Delta k_{eff}$$

This meant that since the Δk_{eff} was not constant throughout the organ, neither was the perfusion. The constant K_w was obtained by an integration technique, in which the perfusion was assumed to be dependent only on the radius and total organ flow. The perfusion was integrated over the entire kidney volume and compared with the total flow into the kidney. This process was carried out on the aforementioned pullback data as follows. The pullback distance was divided by 2 to form an approximate radius at that position (the pullback distance was from one edge to the opposite side, so half that distance was the radius at that location). The Δk_{eff} profile along this radius was normalized to the radius of a sphere with the same volume as the kidney. Then the perfusion (as a function of radius only) $W(r) = K_w \Delta k_{eff}(r)$ was integrated over the sphere volume to find the factor K_w as follows:

$$K_w = \mathring{V}/I \tag{2-6}$$

or,

$$\mathring{V} = \int_V W(r)dV = 8\int_0^R \int_0^{\pi/2} \int_0^{\pi/2} K_w \Delta k_{eff}(r) r^2 \sin\phi \; d\phi \; d\theta \; dr$$

$$= K_w 4\pi \int_0^R \Delta k_{eff}(r) r^2 \; dr \equiv K_w I$$

where:

$$\Delta k_{eff}(r) = k_{eff}(r) - k(r)$$

$k_{eff}(r)$ = effective conductivity at radius r
$k(r)$ = intrinsic conductivity at radius r
V = Kidney volume
$\dot{V}$ = flow rate into kidney
R = radius of sphere with same volume as kidney
r = distance from sphere center
ϕ,θ = dummy angles for integration of sphere

A smoothing function (in the Apple-MacIntosh computer software package 'Cricket') was used on Δk_{eff} before integrating as shown in Figure 2-21. The result using trapezoidal integration yielded K_w = 20.1 and 20.7 ml/100gm min per mW/cm °K for $\dot{V}$ = 78 and 54 ml/min respectively. This result, combined with the linear dependence of k_{eff} on total flow rate lead to an estimate of the perfusion as follows.

$$W(r) = (\dot{V}/\dot{V}_c)K_w\Delta k_{eff}(r) \qquad (2\text{-}7)$$

where:

$\dot{V}_c$ = flow rate into kidney when pullback measurement was performed
$\Delta k_{eff}(r) = \Delta k_{eff}$ at radius r and $\dot{V}_c$

Figure 2-22 shows perfusion vs. distance mappings for the in vitro kidney using equation 2-7 to convert thermal conductivity to perfusion. The maximum perfusion (excluding the second half of the pullback, as discussed earlier) was 225 ml/110gm min. Diedrich (1986) used three methods of obtaining perfusion, and found the maximum perfusion in the range 520 to 2,600 ml/100gm min at the same total organ flow rate: over twice as large as the value calculated by the integration method. One

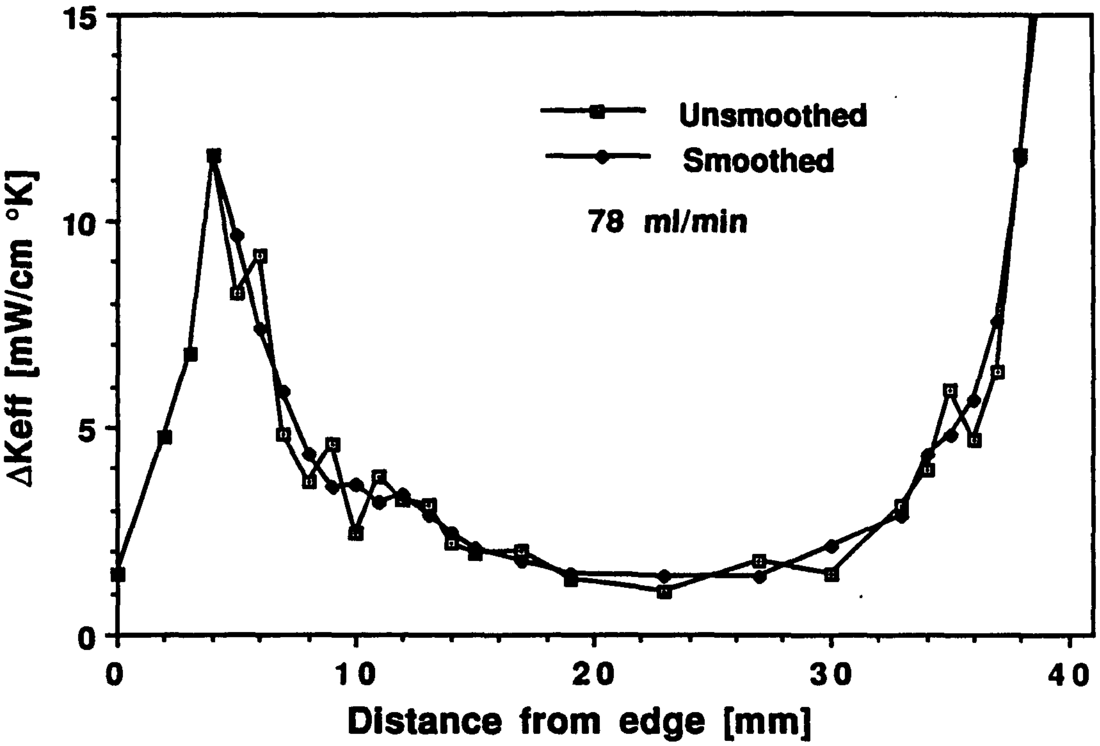


Figure 2-21 Effect of Smoothing. The ΔK_{eff} data was smoothed prior to integration.

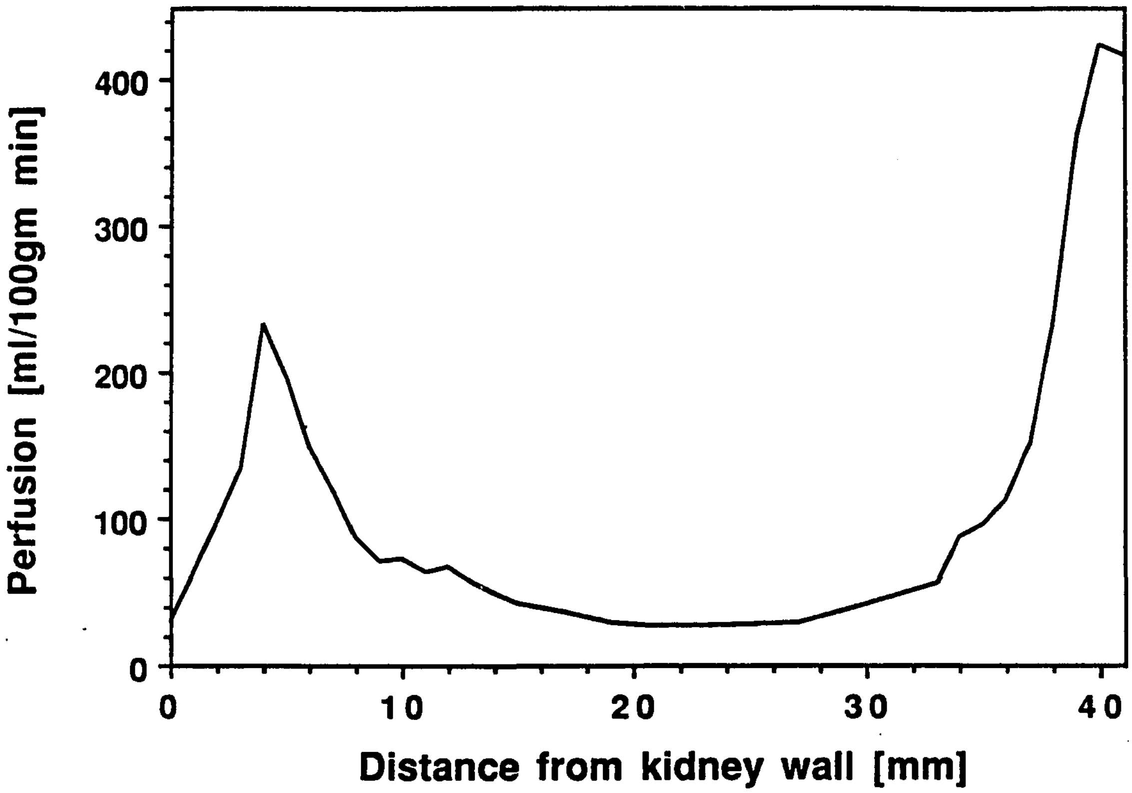


Figure 2-22 Predicted Perfusion Mapping. The Δk_{eff} data was multiplied by a factor obtained with the integration technique to yield an estimated perfusion mapping.

possible reason for this large difference is that the pullback measurement may not have been performed at a location where the perfusion was characteristic of the entire kidney. Since the perfusion is not simply radially dependent, pullback measurements made at non-characteristic locations would result in errors. Another reason is simply the size of the kidney. This enters into the factor K_w through the equivalent radius, R for integration in equation 2-6.

Although the difference between these two sets of data is large, the integration method presented here is still valid as a first order approximation for variations in perfusion with respect to position and total organ flow. The error could be due to a difference in kidney size or to a noncharacteristic pullback location.

CHAPTER THREE

EVALUATION AND UTILIZATION OF PHANTOM

The operation of the dynamic phantom was investigated using three types of tests. These were: 1) programmed flow rates with constant power input, 2) temperature controlled flow rates with constant power input, and 3) programmed flow rates with PID (proportional, integral, differential) power input. These tests were denoted (using initials) PF/CP, TF/CP and PF/PID respectively. The PID controller was a nine-region temperature controller program used for scanned, focused ultrasound hyperthermia, developed by Lin, Roemer and Hynynen (1989). It used temperature feedback from thermocouples to set the power deposition in 9 regions of the scan path. An additional study was performed in which the phantom was used to investigate the optimal controller gain and sampling interval of this PID controller. These experiments were designed to demonstrate the capability of the phantom as a research tool and to verify it's operation.

3.1 Experimental Apparatus

The hyperthermia treatment system described by Hynynen et al. (1987) was used for the experiments. Thermocouples were distributed in the phantom as shown in Figure 3-1. The kidney edges were located with B-scans on the hyperthermia system, and the thermocouples were located with the treatment transducers as

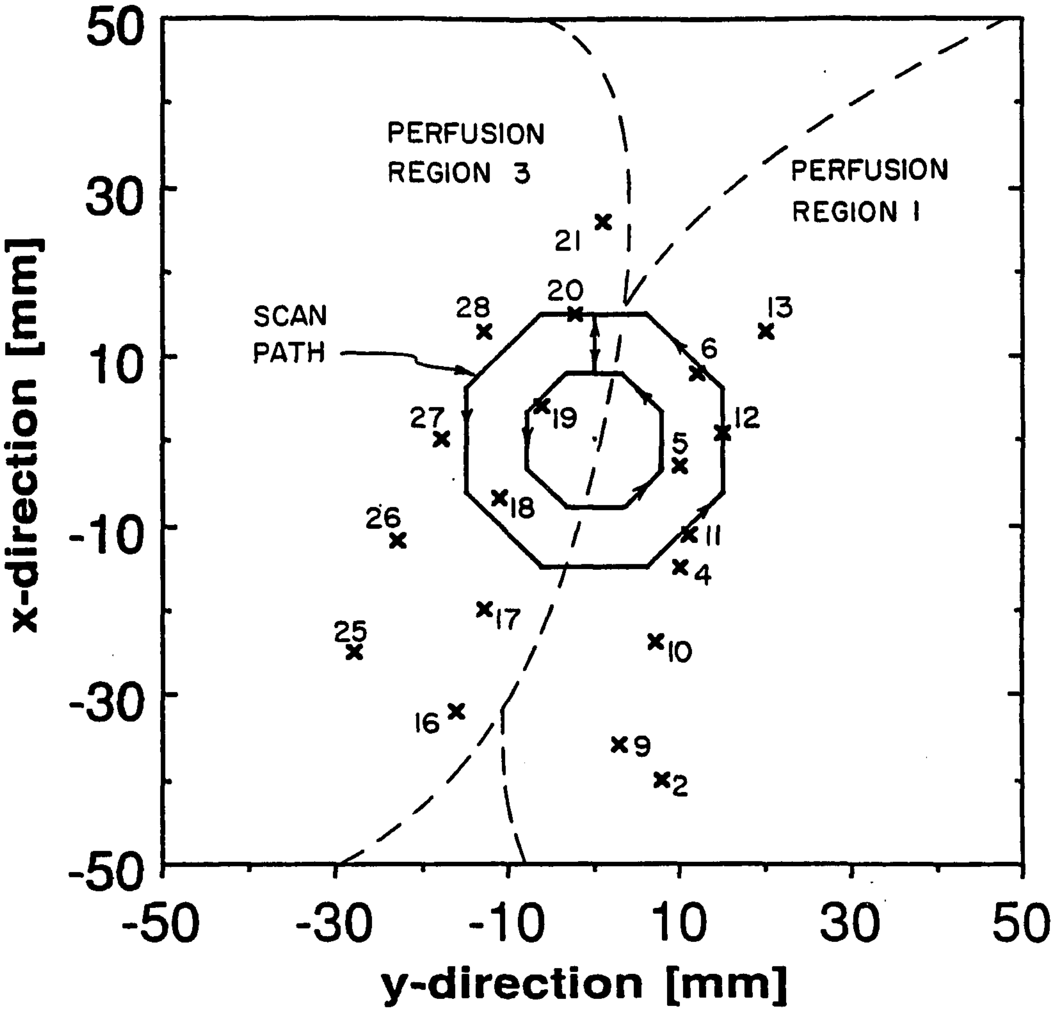


Figure 3-1 Thermocouple and Scan Location. Perfusion regions 1 and 3 contained thermocouples as shown which interfaced with the hyperthermia controller. The arrows indicate scan direction.

described by Hynynen et al. (1987). The X's represent thermocouple locations to within 2 mm, and the associated numbers uniquely identify them. These numbers will be used in the remainder of this chapter to refer to specific thermocouple locations. For example, the thermocouple with the number 5 next to it is referred to as TC5. Regions 1 and 3 correspond to separate kidneys with separate flow inputs (see Figure 2-16). A focussed ultrasound transducer (frequency= 1MHz, diameter= 13cm, radius of curvature= 25 cm, acoustic efficiency= 70%) was scanned in a translational manner through the phantom as shown in Figure 3-1. The scan path location was referenced to the thermocouple locations, and the total scan time was 9 seconds to traverse both paths shown. The heated region was definged as the area within which the temperature was 75% of the peak temperature. Hynynen, DeYoung, Kundrat and Moros (1989) reported results with a similar tranducer at various perfusion levels, showing that the heated region extends about 5 mm beyond the perimeter of the scan path. Thus, the heated region was assumed to include a 5 mm margin around the outer octagonal scan path.

3.2 Programmed Flow, Constant Power (PF/CP)

3.2.1 Procedure for PF/CP Study

A constant input of 15 Watts acoustic power was applied to the phantom by the transducer as it scanned. The Apple Computer controlled the flow rates from a series of pre-programmed steps.

Both step and ramp changes in flow rate were used.

3.2.2 Results of PF/CP Study

First, the temperature response of the phantom to step changes in perfusion was investigated. One such response is shown in Figure 3-2 A and B for regions one and three respectively. The thermocouples shown represent the upper and lower bounds on the temperature measured inside the heated region. In region one, as the flow was increased from 15 to 25 ml/min, the temperature dropped about 2.5 °C. When the flow rate dropped back to its original rate, the temperature increased to its previous level. Region three had 3 step changes in flow rate. The first increase in flow caused temperature drops of 2.1 °C for TC17, and 4.8 °C for TC28. Then the flow rate was increased to 50 ml/min, which caused a drop in temperature of 3.4 °C for TC17, and 1.3 °C for TC28. When the flow rate decreased to 15 ml/min, the thermocouple temperatures began increasing to their original levels until the heating power was terminated at 40 minutes. These changes in temperature demonstrate the sensitivity of the phantom temperature to changes in perfusion initiated by the computer.

This flow pattern was performed three times with very similar results. A comparison of the temperature changes associated with each change in flow rate for these three repetitions is shown in Table 3-1. The difference in temperature change between runs was less than 22% for all cases

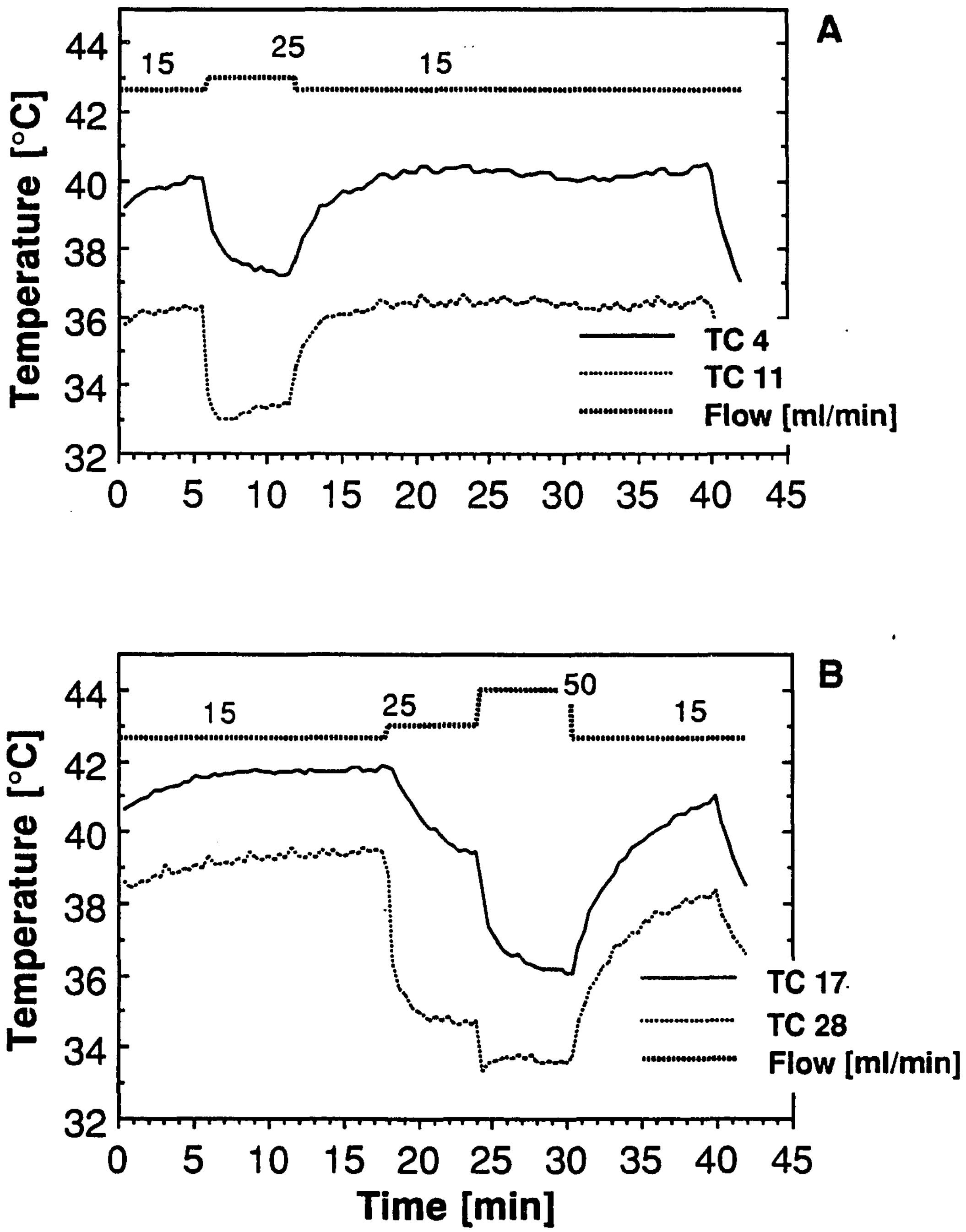


Figure 3-2 Temperature Response to Step Changes in Perfusion Under Constant Power Heating. A) Minimum and maximum temperatures in the heated region of kidney 1, and the flow rate into kidney 1. B) Minimum and maximum temperatures in the heated region of kidney 3, and the flow rate into kidney 3.

TC#	Run #	Flow Change [ml/min] 15 to 25	25 to 15	25 to 50	50 to 15
4	1	2.58	2.86	-	-
	2	2.11	2.48	-	-
	3	2.25	2.72	-	-
11	1	2.95	2.97	-	-
	2	2.92	3.50	-	-
	3	2.32	2.62	-	-
17	1	2.31	-	3.24	4.83
	2	1.83	-	3.50	4.49
	3	2.24	-	-	-
28	1	4.69	-	1.15	4.75
	2	3.89	-	1.53	4.78

Table 3-1 Repeatability of Temperature Response. These repetitions were performed with step changes in perfusion under constant power.

(based on the largest temperature change of the 3 runs), which indicates good repeatability.

The temperature response for a ramp change in perfusion is shown in Figure 3-3. As the flow rate increased, the temperaure decreased. This shows the phantom's versatility in providing various flow patterns.

3.3 Temperature Controlled Flow, Constant Power (TF/CP)

3.3.1 Procedure for TF/CP Study

In this procedure, the phantom was heated at a constant rate of 15 Watts acoustic power, while the flow rate to each perfusion area was controlled by the Apple computer with the temperature controller algorithm (equation 2-5), repeated here for convenience.

$$\dot{V}(t) = \dot{V}_b + G\Delta T(t-t_d)$$

where:

$\dot{V}$ = flow rate [ml/min]
$\dot{V}_b$ = basal flow rate [ml/min]
G = flow gain [ml/min per °C]
ΔT = Temperature elevation [°C]
t = current time [seconds]
t_d = time delay [seconds]

Throughout the entire TF/CP study, the parameter $\dot{V}_b$ was set at 8 ml/min. Experiments were carried out with the other two

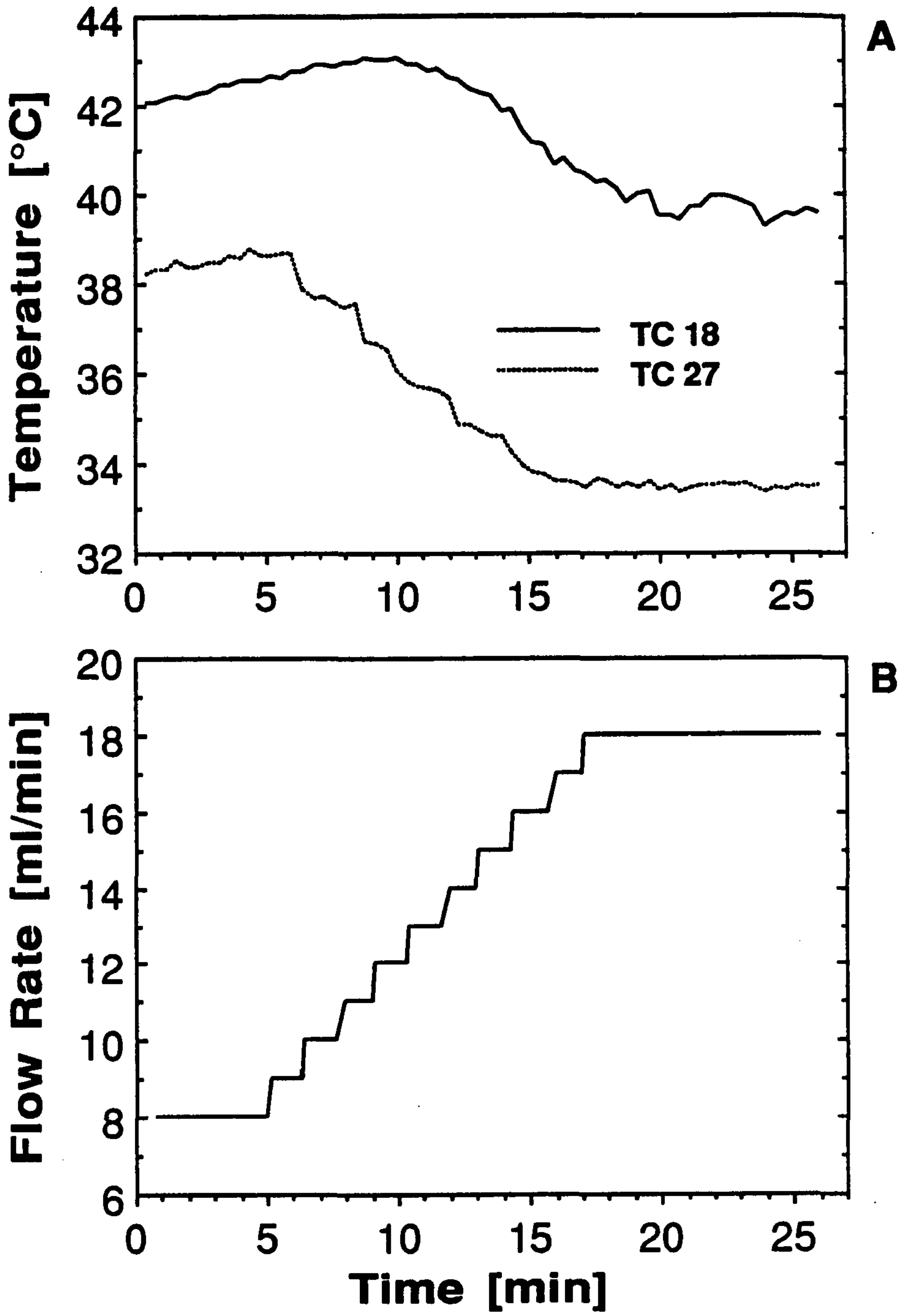


Figure 3-3 Temperature Response to a Ramp Change in Perfusion.
A) Temperature profile of region 3. B) Flow into region 3.

parameters as follows: t_d = 240 seconds with G = 4 and t_d = 240 seconds with G = 2. These experiments were performed to verify the temperature measuring ability of the Apple computer and to show the system's potential to simulate physiological responses to heating.

3.3.2 Results of TF/CP Study

Figure 3-4 shows a comparison of the temperature response measured by the Apple computer and by the hyperthermia controller temperature measuring system (HP 9836) with the temperature controller parameters set at t_d = 240 seconds and G = 4. The corresponding flow rates generated by the Apple computer are shown in Figure 3-5. The temperatures measured by the Apple flow control thermocouples represent the highest of the four measurements in that region (4 thermocouples in each region). The two temperature measurements recorded by the HP (TC10 and TC11 for region 1; TC17 and TC18 for region 3) are the highest and lowest measured temperatures in the corresponding treatment region. In both regions, the temperature measured by the Apple computer closely followed the trends of the temperatures measured by the HP. This demonstrates that the Apple Computer calculates a reasonable estimate of the average temperature in the treatment region, which validates the Apple's ability to perform functions based on the phantom temperature.

The operation of the temperature controller algorithm can be demonstrated by comparison of the flow rates and

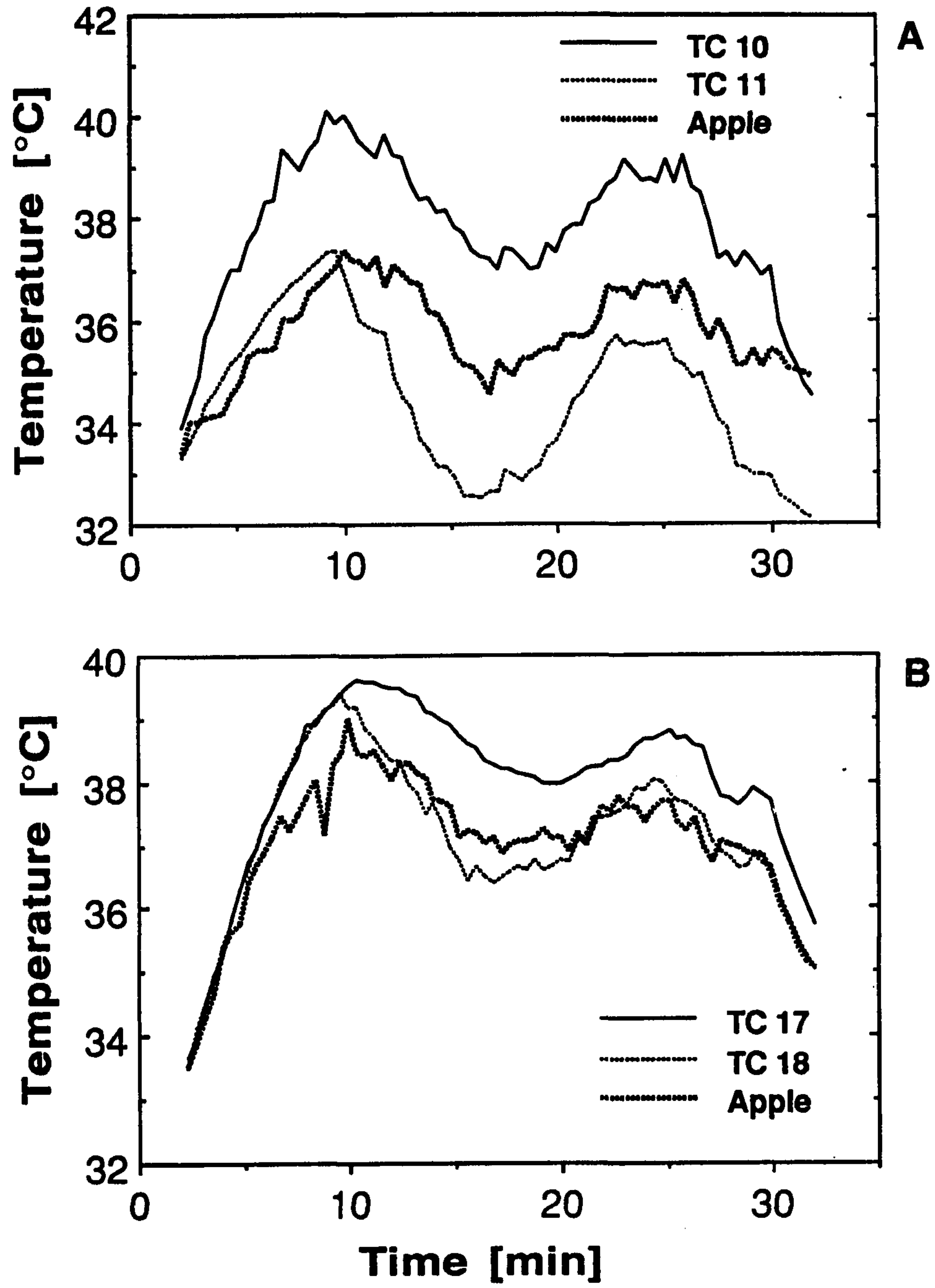


Figure 3-4 Temperature Response at High Flow Rate Gain. Two measurements from the HP computer and the highest measurement from the Apple computer are given in A) region 1, and B) region 3. The flow rates were controlled with the temperature controlled flow algorithm in the Apple with parameters t_d = 240 sec, G = 4.

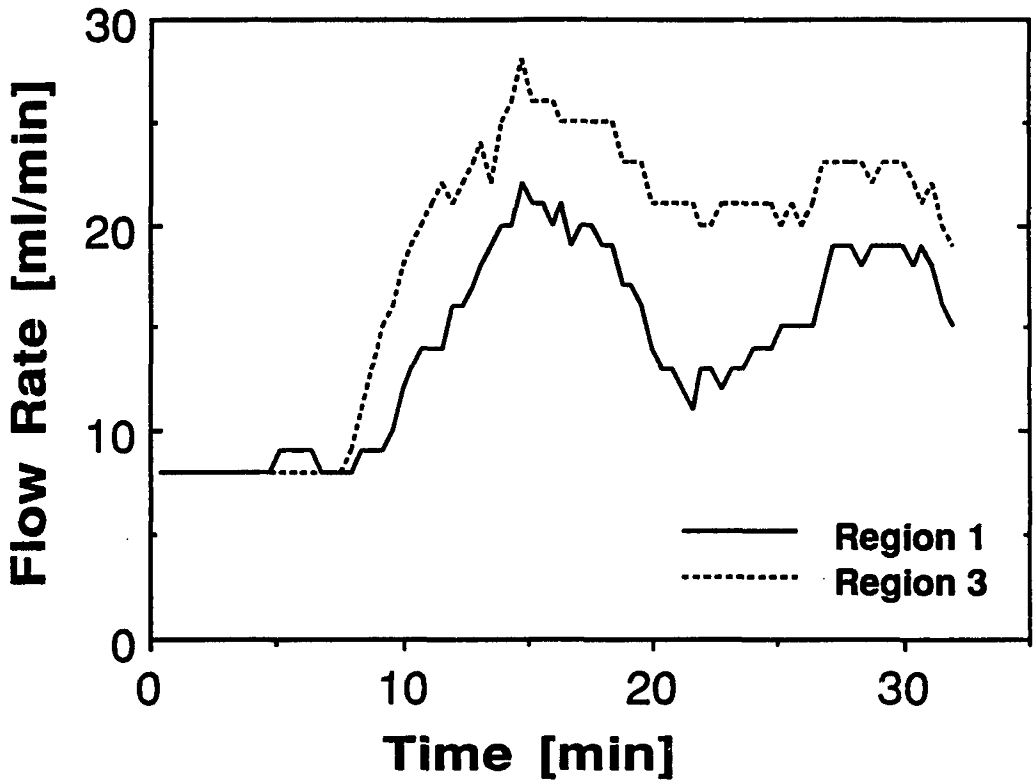


Figure 3-5 Temperature Controlled Flow Rate at High Gain. Flow rates generated by the computer corresponding to Figure 3-4.

temperatures in Figures 3-4 and 3-5. The highest flow rate in region three is 28.4 ml/min which occurred at 14.5 minutes. This is about 4 minutes after the peak temperature (38.8 •C measured by the flow control thermocouples) which corresponds to the programmed time delay of 240 seconds. Furthermore, the increase in flow rate is about 3.7 times the temperature increase (5.5 •C), which corresponds to the programmed flow gain of 4. In region one, the flow increase was about 4.5 times the temperature rise. These results are summarized in Table 3-2. The actual flow gain differs from the programmed flow gain for two reasons. The flow control thermocouple temperatures were weighted over 3 measurements to reduce the effect of measurement errors. This altered slightly the temperature percieved by the flow control algorithm from the temperuare indicated in Figure 3-4. The second reason is the flow rate setting algorithm allowed an error of 1.5 ml/min (see Figure 2-14) in the pertinent flow range. These two effects account for the difference between actual and programmed flow gains.

A set of three experiments was performed with these temperature control parameters. Temperature profiles of TC10 are shown in Figure 3-6 for each of these repetitions. These profiles exhibit little variation, with a maximum temperature rise difference of 28 %. This close grouping of the temperature profiles indicates that the phantom can yield repeatable results.

The second set of temperature controller parameters was t_d

Perfusion Region	Peak temperature rise [°C]	Peak flow rise [ml/min]	Programmed flow gain [ml/min per °C]	Actual flow gain [ml/min per °C]
1	+3.17	+14.1	4	4.5
3	+5.5	+20.4	4	3.7
1	+3.97	+7.2	2	1.81
3	+5.79	+10.3	2	1.78

Table 3-2 Summary of Flow Gain. Programmed and actual flow gains from the temperature controlled algorithm.

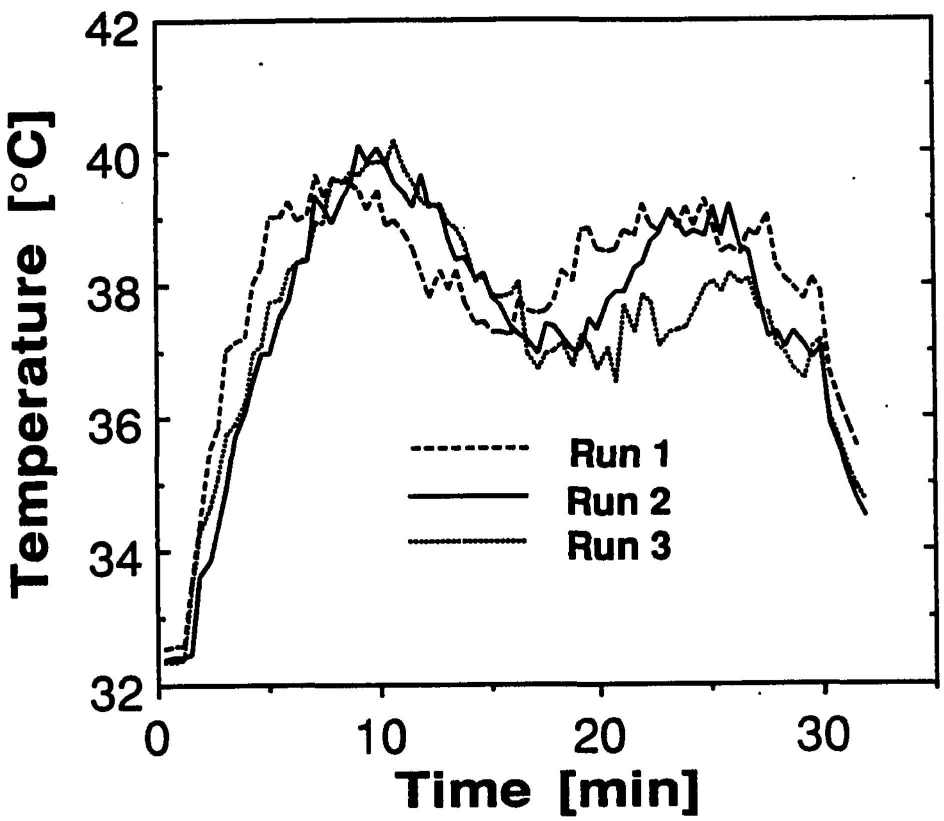


Figure 3-6 Repeatability of Temperature Controlled Algorithm. Three repetitions were performed with very similar results at t_d = 240 seconds, G = 4.

= 240 seconds and G = 2. The temperature profiles and flow rate of region one is shown in Figure 3-7. For this case, the temperature fluctuated less than the previous case, and the flow rates lagged the temperature changes by about 4 minutes, as they should. These results are summarized in Table 3-2. The correlations between the programmed parameters and the actual time delay and flow gain in these two experiments demonstrate the proper functioning of the temperature controlled flow section of the program in the Apple computer.

The fluctuating temperatures of these studies is a result not only of the temperature control parameters in the computer, but also of the physical properties of the phantom tissue. Similar temperature fluctuations have been found by Roemer, Oleson, and Cetas (1985) in dog thighs under constant power heating. This suggests that the physiological mechanism in the dog thigh may have a form similar to the temperature control algorithm (equation 2-5) used in this work.

3.4 Programmed Flow, PID Controlled Power (PF/PID)

3.4.1 Procedure for PF/PID Study

The PID controller was used to control power deposition to the phantom under step changes in flow rate. The target temperature was 36°C, and the PID controller parameters were set at gain = 0.3 and sampling interval = 12 seconds with a maximum available power of 38 Watts (acoustic). The nine regions used by the controller are shown in Figure 3-8.

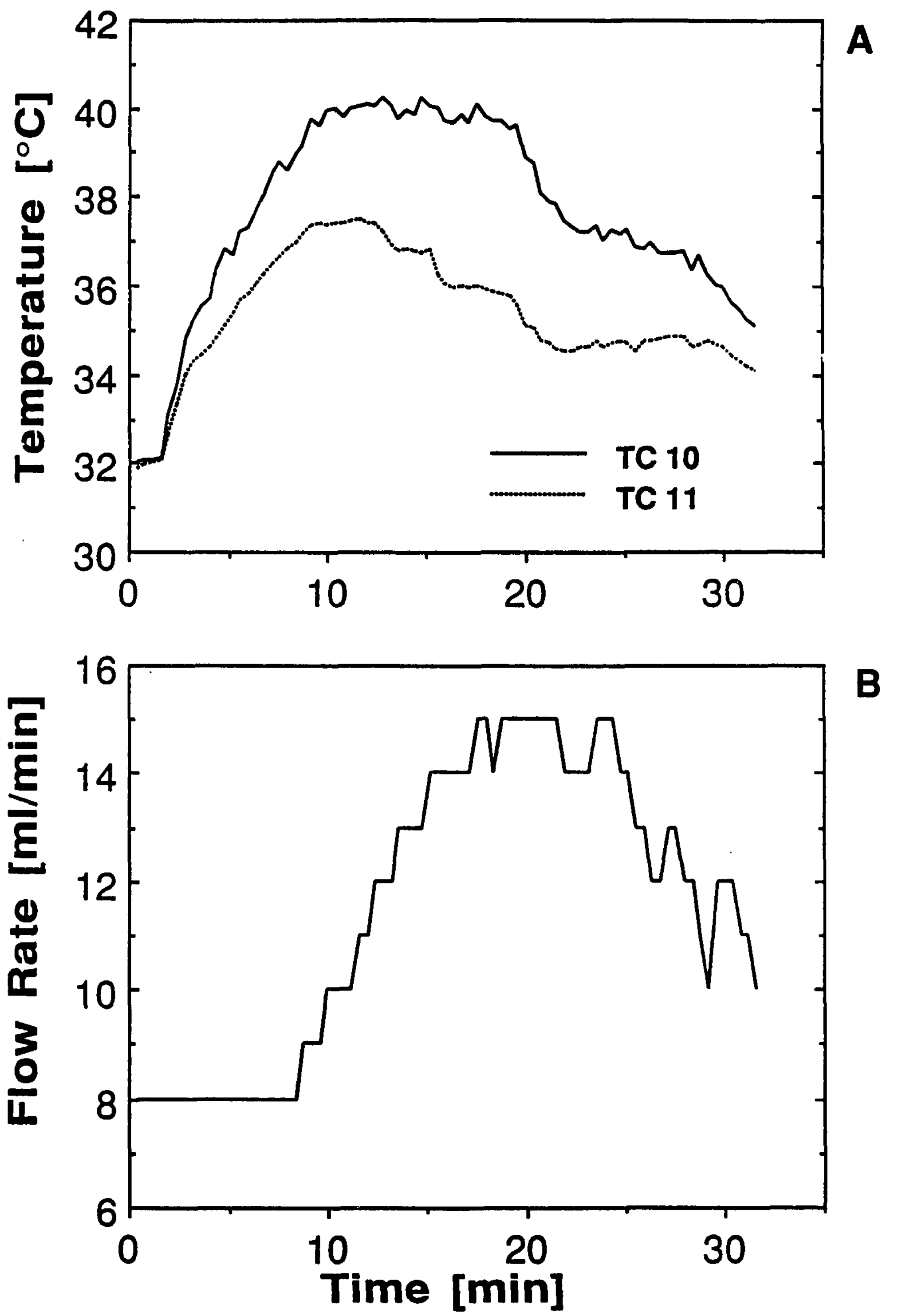


Figure 3-7 Temperature Response at Low Flow Rate Gain. A) Temperature response. B) Flow rate generated by the computer. The temperature controller parameters were t_d = 240 seconds and G = 2.

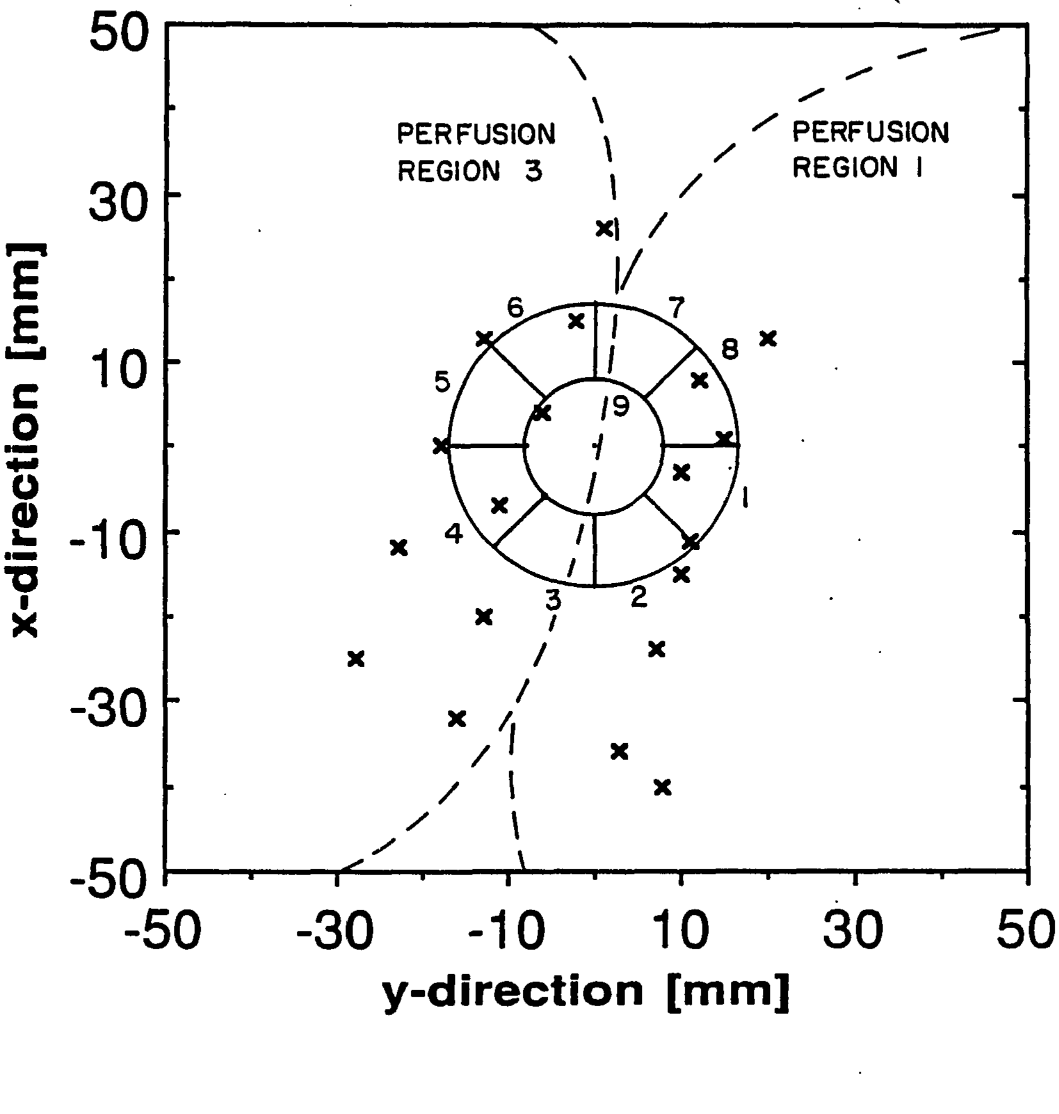


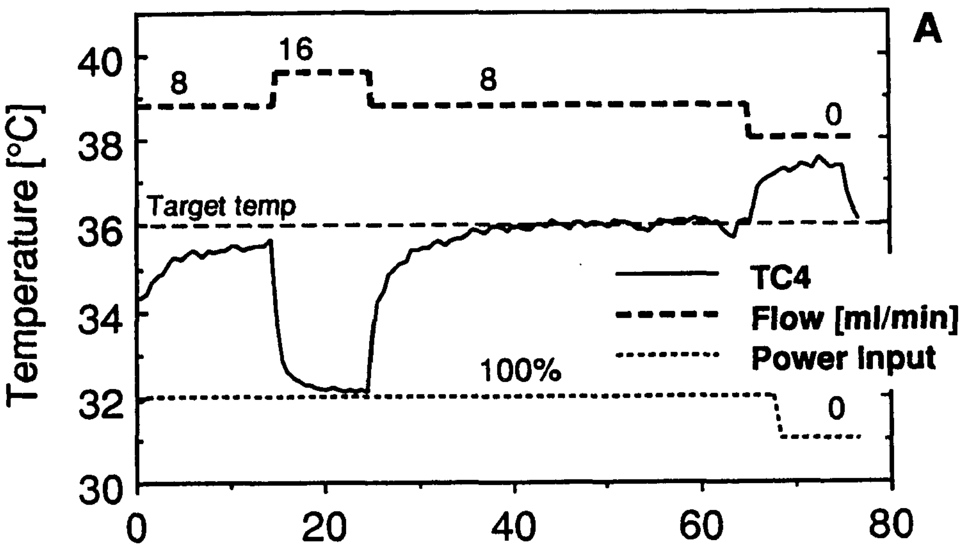
Figure 3-8 PID Controller Regions. Nine Regions used by the PID temperature controller are numbered and superimposed on perfusion regions 1 and 3 of the phantom.

3.4.2 Results of PF/PID Study and PID Optimization

The temperature response resulting from a step function in flow rate is shown in Figure 3-9. In part A, the temperature, flow rate, and power input of controller region 2 (in perfusion region 1) are shown, and in part B, the temperature, flow rate, and and power input of controller region 4 (in perfusion region 3) are shown. The target temperatures were not well regulated due to the low available power, and to non-optimal placement of the scan path. Nevertheless, the temperature and power still changed in response to perfusion changes showing that the phantom can provide a severe test of controllers.

In controller region 2, the power was set to 0 by the controller since conduction or power carry-over from the neighboring regions was sufficient to heat that section above the target temperature. When the flow rate increased to 16 ml/min, the temperature dropped below the target, and the power began to increase. Then the flow increased again, and the power was switched on fully as the temperature decreased even further. When the flow rate dropped back to zero, the temperature rose above the target and the power dropped to zero.

In perfusion region 3, the temperature never rose above the target until the flow dropped to zero at 62 minutes, at which point the temperature increased. In response to this, the power dropped to 0 which allowed the temperature to stabilize at about 1.8 °C above the target. In both cases, the temperatures and hence the heating power were clearly responding to programmed



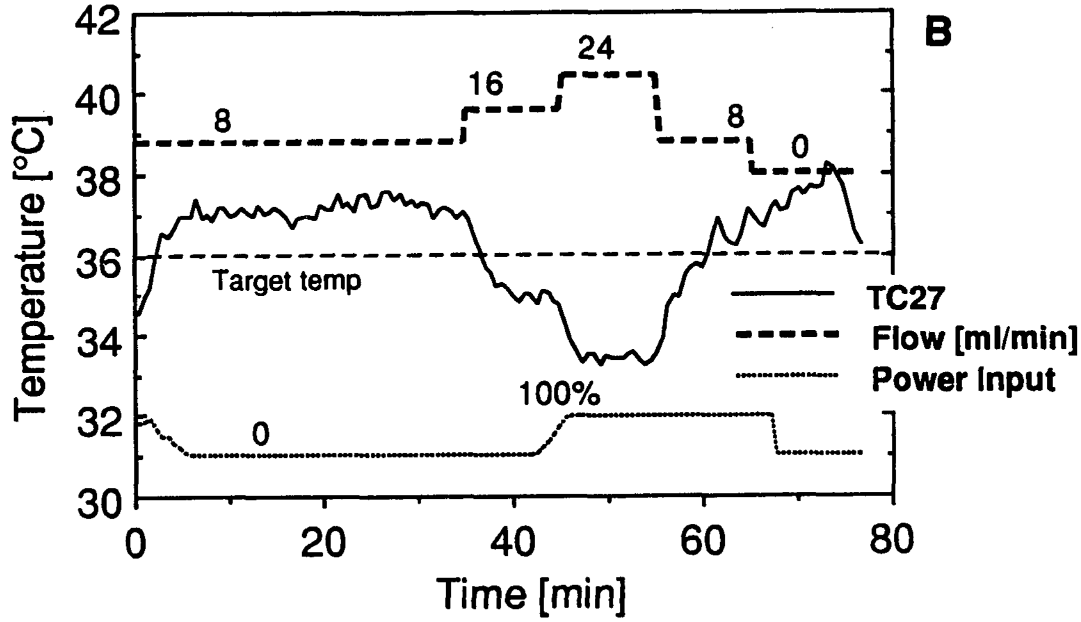


Figure 3-9 Temperature Response with PID Power. Step changes in flow rate were applied to the phantom, while a PID hyperthermia controller adjusted the power based on a 36 °C target temperature. A) Controller region 2 (in perfusion region 1). B) Controller region 4 (in perfusion region 3).

changes in perfusion. An ability to evoke such responses in power set-point is prerequisite for investigating hyperthermia controllers. In this case, various perfusions were attained which required maximum power and zero power to drive the temperature towards the target.

An additional investigation was performed which centered on the PID controller. The details of this experiment are in Appendix E, as it is not directly related to the function of the dynamic phantom.

The effect of gain and sampling interval on the stability of the PID controller were investigated at 3 perfusion levels. Although the experimental error seems to have been substantial and only single data points were taken, definite trends were nevertheless apparent. The optimal controller parameters (corresponding to small percent overshoot, fast settling time, and low steady state temperature fluctuations) were achieved with a gain of 0.3 and a sampling interval of 10 seconds. These values are close to the values obtained through computer simulations by Lin, Roemer, and Hynynen (1989), which were a gain of 0.3 and a sampling interval of 12 seconds.

CHAPTER FOUR

CONCLUSIONS AND RECOMMENDATIONS

4.1 Conclusions

The dynamic phantom in this work satisfied most of it's design criteria. It provided repeatable temperature elevations at normal power levels, and controlled the flow rate to each of four regions repeatably using either the temperature controlled mode or the pre-programmed mode. Since the phantom had only been in use for six months at the time of this writing, the question of longevity could not be definitively answered. However, there was no readily apparent degradation over that time, indicating the preservation technique works satisfactorily. The phantom was simple to prepare for use: It was easy to place on the treatment device, and to connect the various interfaces. But due to slow leaks in the joints and connections of the sonication chamber and valves, air was gradually bled into the system, necessitating periodic degassing of the Ethanol. This problem could have been avoided by using extreme care during construction, and by carefully removing all bubbles in the tubing, valves, flow meters and sonication chamber. Currently, the system requires degassing every 4-6 weeks.

All of the experiments performed thus far involved only two kidneys scanned in two dimensions with thermocouples only in the scan plane. This was due to the nonexistence of three-dimensional treatment programs for the hyperthermia system used here. Although only two regions have been used, the method is the same for three-dimensional treatments, and the phantom could expedite the development of such three-dimensional programs without any modification.

The dynamic phantom presented in this work is a flexible system, which produces repeatable results. Although there is room for improvement, it has been successfully used in the stability analysis of a PID hyperthermia controller, and to investigate certain physiological phenomena in hyperthermia treatments. The capability of this system imparted by it's computer based nature invites automated testing. Such a technique could greatly speed investigations of hyperthermia controllers.

4.2 Recommendations

The flow meters measured the flow repeatably and quickly (1 second), but the software only measured the rate to within 1.8 ml/min, which caused a noticeably quantized output. The stepper motors used to position the valves were relatively slow (15 seconds from closed to full open). In addition, the flow rate setting routine could require up to 3 iterations of valve position for any one valve to settle on the specified flow rate due to the quatization of the flow meter output, and the non-linear behavior of the valves at high flow rates. These factors combined to produce rather large flow rate settling times (up to

30 seconds), depending on the number of valves being controlled and the change in target flow rate. This delay could be reduced in four ways: 1) The stepping speed of the positioning motors could be increased, which would necessitate a ramp-up and ramp-down of stepping rate at the beginning and end of a motor movement. This could be done in the machine code which controlling the motors. 2) The flow measuring program, which measures the frequency output of the flow meters, could be converted to measure over a smaller time interval by measuring the period between pulses instead of the number of pulses in a given time. This would decrease the error at low flow rates, and decrease the measurement time by at least 1/4. 3) The flow rate setting routine could be improved by allowing it to 'learn' combinations of valve positions and flow rates and interpolating from those to improve future position estimates. 4) The valves and flow meters could be replaced by independent stepper motor-controlled pumps for each kidney. This would allow rapid, accurate changes in flow rate, but would require complex electronics and software.

APPENDIX A

BASIC PROGRAM

A supplementary description of the Basic program used to control the phantom is given here. The program required a printer interface in slot one of the Apple computer, a parallel intertface in slot two, and an ADC card in slot four. Since most of the program is self-explanatory, only certain references are made to the program listing which immediately follows.

Lines 100-560 initialize the text window, memory allocation, arrays, I/O ports and the number of valves to control. Following this, the main menu with it's subroutines is given. Line 1590 sets the text window to ignore the upper line of text, which is used to display the currents status of the program. Lines 1600-1910 initialize the stepper motor positions by closing the valves until the limit switches contact. Line 1930 calls a subroutine which initializes the fluid type to 80% Ethanol (water as a working fluid may be chosen as well), and loads the constants for the valve position estimate (Equation 2-3) and for the thermocouple voltage conversion (Equation 2-4). The flow meters are initialized in lines 2200-2450 and the factor e of Equation 2-3 is initialized in lines 2400-2430. The subroutine at lines 3550-3780 reads the pressure transducer (averaged over 5 readings) and calls a machine code subroutine (see appendix B) to measure the flow meter frequency output. The machine code loads the frequency measurements directly into the Basic integer array 'ADC%' to locations ADC%(32-39). Equation 2-1 is implemented to convert these frequency measurements to flow rates. The temperature is measured in a subroutine at lines 3790-4090. The thermocouple voltage (loaded into ADC%(0-15) by the machine code) and reference temperature are converted in lines 3960-4022 to a thermocouple temperature using Equation 2-4. The valve positions are set to provide the desired flow rates in lines 4100-4530. This subroutine calls the stepper motor set-up subroutine at lines 4660-4920, which pokes the motor number, distance and direction into memory and then calls a machine code subroutine which moves the motors.

```
100  REM  *********** FOUR VALVE ***********
110  REM  *********  BY JON B. ZAERR *******
120  REM   UPDATE JUNE , 89
130  POKE 34,0
140  HOME : PRINT ">>>>   ";
150  PRINT "FOUR VALVE CONTROL PROGRAM";
160  PRINT "   <<<<"
170  PRINT "********  FEB 89 BY";
180  PRINT " JON ZAERR";
190  PRINT "  *******"
200  PRINT : PRINT
210  HIMEM: 31200
220  REM  M=MOTOR #
230  REM  CUR(M)=CURRENT POS OF MTH MOTOR
240  REM  OK(M)=STATUS OF MTH MOTOR (0=DOES'T WORK)
250  REM  GOAL=WHERE MTH MOTOR SHOULD GO
260 D$ =  CHR$ (4)
270 G$ =  CHR$ (7)
280  IF  PEEK (32768) +  PEEK (32770) <  > 366 THEN  PRINT
 D$"BLOAD BIGN"
290  DIM ADC%(40),FL(8)
300  DIM T(50),V(4,50),TZ(3,15),TCUR(15)
310  DIM DT(7,110)
320 HYST = 2: REM  MECH HYST
330 MX = 372: REM   MAX DIST
340 M =  - 1: REM FLG TO INIT MTRS
350 MM = 32762: REM  MOTOR # FOR 'MTR'
360 MD = 32763: REM  DIRECTION FOR 'MTR'
370 MS = 32764: REM  # STPS TO MOVE FOR 'MTR'
380 MTR =  - 32352: REM  MOTOR MOVER ENTRY PT
390 D5 = 32768: REM  5 SEC TIMER
400 D05 =  - 32656: REM  .05 SEC TIMER
410 PA =  - 16224: REM  PORT A
420 CA =  - 16223: REM  COMM A
430 PB =  - 16222
440 CB =  - 16221
450  REM  INIT I/O
460  POKE CA,0
470  POKE PA,127: REM  7 OUT
480  POKE CA,36
490  POKE CB,0
500  POKE PB,255: REM  8 OUT
510  POKE CB,36
520  POKE PA,0: POKE PB,0
530  INPUT "ENTER # VALVES TO CONTROL: ";NV
540 S = 0: REM  #STEPS
550  FOR I = 1 TO 5:BK$ = BK$ +  CHR$ (8): NEXT
560  HOME
570  REM   LOOP TO HERE ************************
580  VTAB 1
590  PRINT NS" STEPS, ";
```

```
600  PRINT NV" VALVES";
610  HTAB 25: PRINT "MAIN MENU    "
620  PRINT : PRINT "1) ENTER NEW DATA"
630  PRINT "2) LOAD DATA FROM DISK"
640  PRINT "3) SAVE DATA TO DISK"
650  PRINT "4) LIST DATA"
660  PRINT "5) SELECT CONTROL MODE   "
670  PRINT "6) END"
680  POKE  - 16368,0
690  PRINT "ENTER CHOICE: ";
700  GET C$:C =  VAL (C$)
710  VTAB 1: HTAB 1: PRINT "                                   ": VTAB 10
720  VTAB 10
730  ON C GOSUB 760,1130,1310,1460,1580,1570
740 S = NS
750  GOTO 570
760  REM   DATA ENTRY SECTION---------------------
770  IF NV = 0 THEN  INPUT "ENTER # VALVES: ";NV
780  HOME
790 VT = 5:HT = 1 - 10
800 S = S + 1
810 T(S) = T(S - 1)
820  VTAB 1: HTAB 1
830  PRINT "STEP "S
840  VTAB 4: PRINT "========================================
";
850  VTAB 2: HTAB 1
860  PRINT "ENTER Q OR DURATION ("5 * T(S)" SEC):       ";
870  PRINT BK$;: INPUT T$
880  IF T$ = "Q" THEN  HOME :S = S - 1:NS = S: RETURN
890  IF T$ <  > "" THEN T(S) =  INT ( VAL (T$) / 5 + .5)
900  FOR J = 1 TO NV
910  VTAB 3: HTAB 1
920 V(J,S) = V(J,S - 1)
930  PRINT "FLOW #"J"  ("V(J,S)" ML/MIN):       ";
940  PRINT BK$;: INPUT V$
950  IF V$ <  > "" THEN V(J,S) =  VAL (V$)
960  NEXT J
970 HT = HT + 10
980  IF HT > 31 THEN HT = 1:VT = VT + 3 + NV
990  IF VT > 19 THEN  HOME :VT = 5
1000  GOSUB 1020: REM  PRINT RESLT
1010  GOTO 800
1020  REM  PRINT RESULTS
1030  HTAB HT: VTAB VT
1040  PRINT "STEP "S
1050  HTAB HT
1060  PRINT 5 * T(S)" SEC"
1070  FOR J = 1 TO NV
1080  HTAB HT
1090  VTAB VT + 1 + J
1100  PRINT "V"J" ="V(J,S);
```

```
1110  NEXT J
1120  RETURN
1130  REM   LOAD DATA ------------------------------
1140  INPUT "ENTER NAME OF FILE TO LOAD: ";F$
1150  PRINT D$"MON,I"
1160  PRINT D$"OPEN "F$
1170  PRINT D$"READ "F$
1180  INPUT NS,NV: REM  #STPS,#VLVS
1190  FOR S = 1 TO NS
1200  INPUT T(S)
1210  FOR J = 1 TO NV
1220  INPUT V(J,S)
1230  NEXT J
1240  NEXT S
1250  PRINT D$"CLOSE "F$
1260  PRINT D$"NOMON,I"
1270  HOME
1280  PRINT F$" LOADED"
1290  RETURN
1300  REM
1310  REM   SAVE DATA ------------------------------
1320  INPUT "ENTER FILE NAME TO SAVE: ";F$
1330  PRINT D$"OPEN "F$
1340  PRINT D$"WRITE "F$
1350  PRINT NS","NV
1360  FOR S = 1 TO NS
1370  PRINT T(S)
1380  FOR J = 1 TO NV
1390  PRINT V(J,S)
1400  NEXT J
1410  NEXT S
1420  PRINT D$"CLOSE "F$
1430  HOME
1440  PRINT F$" SAVED
1450  RETURN
1460  REM  LIST DATA ----------------------------
1470  PRINT "FILE: "F$
1480  FOR S = 1 TO NS
1490  PRINT : PRINT "STEP "S"    "5 * T(S)" SEC
1500  FOR J = 1 TO NV
1510  PRINT V(J,S)
1520  NEXT J: NEXT S
1530  PRINT D$"PR#0"
1540  INPUT "PRINT THIS (Y/N) ";AN$
1550  IF AN$ = "Y" THEN  PRINT D$"PR#1": GOTO 1470
1560  HOME : RETURN
1570  POKE 34,0: HOME : END : REM END PROGRAM
1580  REM   INITIALIZE MOTORS **********************
1590  HOME : POKE 34,1: VTAB 2
1600  IF M >  = 0 THEN  GOTO 1920
1610  PRINT "INITIALIZING..."
1620 POFF = .1637
```

```
1630  PRINT "RESETTING MOTORS..."
1640  FOR M = 0 TO NV - 1
1650 OK(M) = 1: REM  INNOCENT UNTIL GUILTY
1660  POKE PB,M
1670  REM  CHECK LIM SW
1680  IF  PEEK (PA) < 128 THEN 1810: REM  BRN IF OPEN
1690  REM  LIM SW CLOSED, SO OPEN A LITLE
1700  PRINT "SWITCH "M" CLOSED"
1710  POKE MM,M: REM  M#
1720  POKE MD,1
1730  POKE MS,8: REM    DIST
1740  PRINT "TRYING TO OPEN..."
1750  CALL MTR: REM  SUBROUTINE
1760  IF  PEEK (PA) < 128 THEN 1810: REM  BRN IF OPEN
1770  REM  IF STILL CLOSED, ITS BROKE
1780  PRINT G$"MOTOR IS BROKE!"
1790 OK(M) = 0
1800  GOTO 1910
1810  POKE MM,M: REM  M#
1820  POKE MD,0: REM   DIR=BKWD
1830  POKE MS,230: REM  ARBIT DIST
1840  PRINT "CLOSING MOTOR "M + 1
1850  CALL MTR
1860  POKE MM,M: POKE MD,0: POKE MS,200
1870  IF  PEEK (PA) < 128 THEN  CALL MTR
1880 CUR(M) = 0
1890 CH(M) =  - 1: REM  SET HYSTERYSIS SGN
1900  IF  PEEK (MD) <  > 255 THEN OK(M) = 0
1910  NEXT M
1920  REM  MAIN PROGRAM ******************************
1930  GOSUB 4930: REM  SET FLUID TYPE
1940  HOME
1950  VTAB 1: HTAB 25: PRINT "MODE MENU   "
1960  PRINT "1) MONITOR V,T"
1970  PRINT "2) CONTROL ENTERED DATA"
1980  PRINT "3) CONTROL INTERACTIVELY"
1990  PRINT "4) CONTROL WITH TEMP INPUT"
2000  PRINT "5) INITIALIZE FLOW METERS"
2010  PRINT "6) EXIT"
2020  POKE  - 16368,0
2030  PRINT "ENTER CHOICE: ";: GET I2$:I2 =  VAL (I2$)
2040  PRINT
2050  ON I2 GOSUB 2090,2460,2670,3020,2200
2060  POKE  - 16368,0
2070  IF I2 = 6 THEN  HOME : RETURN
2080  GOTO 1940
2090  REM  JUST MONITOR P,V,T
2100  HOME
2110  VTAB 1: HTAB 25: PRINT "MONITOR V,T  "
2120  VTAB 2
2130  GOSUB 3790: REM  GET T
2140  GOSUB 3550: REM  GET P,V
```

```
2150  GOSUB 4540: REM  PRN P,V,T
2160  IF  PEEK ( - 16384) <  > 155 THEN 2120: REM  CHK 'ES
C'
2170  POKE  - 16368,0: REM  CLR KBD
2180  VTAB 1: HTAB 25: PRINT "                "
2190  RETURN
2200  REM   INIT FLOWMETERS ******************
2210  HOME : VTAB 1: HTAB 25: PRINT "INIT FLW MTRS "
2220  FOR J = 0 TO NV - 1
2230  FOR MP = 100 TO 300 STEP 20
2240  GOSUB 4660: GOSUB 3550: REM  MOVE, PV
2250  IF VA(J) < 30 THEN  NEXT MP
2260  PRINT VA(J)"ML/MIN    "PSI"PSI    THETA:"CUR(J)
2270 C1(J) = VA(J) / (CUR(J) * PSI)
2280  NEXT J
2290 PSI(1) = PSI:VSUM(1) = VSUM
2300  FOR J = 0 TO NV - 1
2310 MP = CUR(J) + 80: GOSUB 4660
2320  NEXT J
2330  GOSUB 3550: REM  PV
2340  GOSUB 4540: REM PRN PV
2350  FOR J = 0 TO NV - 1
2360  IF VA(J) < 1.1 * FB THEN MP = CUR(J) + 20: GOSUB 466
0
2370  NEXT J
2380  REM  SET FLOW CONST
2390  FOR J = 0 TO NV - 1
2400 X = .5 * (E1(J) + E2(J) * (AZ(J) - CUR(J))) / E3(J) ^
 3
2410 X =  LOG (X +  SQR (X * X + 1))
2420 X = E3(J) * ( EXP ( - X / 3) -  EXP (X / 3)) - CZ(J)
2430 EE(J) = PSI * (X / 3 / DZ(J) - AP(J) * PSI) / VA(J)
2440  NEXT J
2450  RETURN
2460  REM  ENTERED DATA SECTION *********************
2470  HOME
2480  VTAB 1: HTAB 25: PRINT "PRGRM CNTRL    "
2490  FOR S = 1 TO NS
2500  FOR J = 1 TO NV
2510 VT(J - 1) = V(J,S)
2520  NEXT J
2530 FUDGE = 0
2540  GOSUB 4100: REM  SET VALVES
2550  GOSUB 3790: REM  TEMP
2560  HOME : PRINT "STEP "S"/"NS
2570  GOSUB 4540: REM  PRN PVT
2580  IF S = 1 THEN  INPUT "<CR> TO START TIMING";J$
2590 TT = T(S) -  INT (FUDGE * .01 + .5)
2600  IF TT < 0 THEN TT = 0
2610  PRINT : PRINT "WAIT: "5 * T(S)" SEC"
2620  POKE 255,TT: CALL DS
2630  IF  PEEK ( - 16384) <  > 155 THEN  NEXT S
```

```
2640  POKE  - 16368,0
2650  VTAB 1: HTAB 25: PRINT "                    "
2660  RETURN : REM  BACK TO MAIN
2670  REM  INTERACTIVE SECTION  ******************
2680  VTAB 1: HTAB 25: PRINT "INTERACT MENU"
2690  HOME
2700  FOR J = 0 TO NV - 1
2710  PRINT J + 1"  "VT(J)" ML/MIN"
2720  NEXT J
2730  PRINT : PRINT "1) CHANGE FLOWS"
2740  PRINT "2) CONTROL FLOWS"
2750  PRINT "3) MONITOR V"
2760  PRINT "4) EXIT"
2770  POKE  - 16368,0
2780  GET I3$:I3 =  VAL (I3$)
2790  ON I3 GOTO 2900,2820,2940
2800  IF I3 = 4 THEN  HOME : RETURN
2810  PRINT : GOTO 2700
2820 VV =  PEEK (37):HH =  PEEK (36): HTAB 25: VTAB 1: PRI
NT "CONTROL FLOW "
2830  HTAB HH + 1: VTAB VV + 1
2840  GOSUB 4100: REM SET VALVES
2850  HOME
2860  GOSUB 4540: REM  PRN PVT
2870  IF  PEEK ( - 16384) <  > 155 THEN 2840
2880  POKE  - 16368,0
2890  GOTO 2670
2900  INPUT "ENTER V#, FLOW: ";J,V
2910  IF J < 1 OR J > 4 THEN 2930
2920 VT(J - 1) = V
2930  GOTO 2670
2940  REM  MONITOR FLOWS
2950  VTAB 1: HTAB 25: PRINT "MONITOR V      "
2960  HOME
2970  FOR J = 0 TO 3:TKID(J) = 0: NEXT
2980  GOSUB 3550
2990  VTAB 8: GOSUB 4540
3000  IF  PEEK ( - 16384) = 155 THEN  GOTO 2670
3010  GOTO 2980
3020  REM  TEMP INPUT SECTION  *******************
3030  VTAB 1: HTAB 25: PRINT "TEMP CONTROL "
3040  HOME : PRINT
3050  FOR J = 0 TO NV - 1
3060  PRINT "V"J + 1
3070  INPUT "ENTER FLOW @ AMB T: ";VN(J)
3080  INPUT "ENTER FLOW GAIN/DEG C: ";G(J)
3090  INPUT "ENTER RESPONSE DELAY(20-2000 SEC): ";TW(J)
3100 TW(J) =  INT (TW(J) / 20 + .5)
3110  IF TW(J) < 1 THEN TW(J) = 1
3120  IF TW(J) > 100 THEN TW(J) = 100
3130  PRINT 20 * TW(J)" SEC"
3140  NEXT J
```

```
3150 SR = 0
3160  INPUT "PRINT V,T (Y/N) ?";J$: IF J$ = "Y" THEN SR =
1: PRINT D$"PR#1"
3170  FOR J = 0 TO NV - 1: PRINT "G="G(J);
3180  PRINT "  DEL="20 * TW(J): NEXT
3190  PRINT : IF SR THEN  PRINT D$"PR#0"
3200  PRINT "SETTING AMBIENT TEMP...";
3210 TAMB = 0
3220  GOSUB 3790: REM  READ TEMP
3230  FOR J = 0 TO 3:TAMB = TAMB + TKID(J): NEXT
3240 TAMB = TAMB / 4
3250  REM  RESET DIFFERENCE ARRAY
3260  FOR J = 0 TO NV - 1: FOR I = 00 TO 102
3270 DT(J,I) = 0: NEXT : NEXT
3280  PRINT "DONE"
3290  GOSUB 3790: REM   READ  TEMP
3300  FOR J = 0 TO NV - 1
3310  FOR I = 102 TO 1 STEP  - 1
3320 DT(J,I) = DT(J,I - 1)
3330  NEXT I
3340 T = TW(J)
3350 DT(J,0) = TKID(J) - TAMB
3360  IF DT(J,0) < 0 THEN DT(J,0) = 0
3370 WT = DT(J,T)
3380 VT(J) = VN(J) + G(J) * WT
3390 VT(J) =  INT (VT(J) * 10 + .5) / 10
3400  NEXT J
3410  GOSUB 4100: REM  SET VALVES
3420  HOME
3430  GOSUB 4540: REM PRN PVT
3440  IF SR THEN  PRINT D$"PR#1": FOR J = 0 TO NV - 1: PRI
NT VT(J)", "TK(J)"C    ";: NEXT
3450  PRINT : PRINT D$"PR#0"
3460  REM  CHECK 'ESC'
3470  IF  PEEK ( - 16384) = 155 THEN 3510
3480  POKE 255,2: CALL D5: REM 10 SEC
3490  IF  PEEK ( - 16384) = 155 THEN 3510
3500  GOTO 3290
3510  POKE  - 16368,0
3520 VV =  PEEK (37):HH =  PEEK (36): HTAB 25: VTAB 1: PRI
NT "              "
3530  HTAB HH + 1: VTAB VV + 1
3540  RETURN : REM  BACK TO MAIN
3550  REM  GET PRESSURE AND FLOW SUBRT ***************
3560 VV =  PEEK (37):HH =  PEEK (36): HTAB 1: VTAB 1
3570  PRINT "READ FLOW RATE    "
3580  CALL  - 32095:FUD = FUD + 18: REM  FLOW
3590 PSI = 0
3600  FOR II = 1 TO 5
3610  CALL  - 32090
3620 PSI = PSI + ADC%(17)
3630  NEXT II
```

```
3640  FOR II = 0 TO 3
3650 VA(II) = ADC%(32 + II)
3660  NEXT II
3670 VSUM = 0
3680  FOR II = 0 TO 3
3690 VA(II) = FM * VA(II) + FB
3700  IF VA(II) < 1.1 * FB THEN VA(II) = 0
3710 VSUM = VSUM + VA(II)
3720  NEXT II
3730 PSI = 1.6667 * (PSI / (409.5 * 5) - .25)
3740 PSI =  INT (PSI * 1000 + .5) / 1000 + .1: REM OFFSET

3750  VTAB 1: HTAB 1
3760  PRINT "                    "
3770  HTAB HH + 1: VTAB VV + 1
3780  RETURN
3790  REM  READ  TEMPERATURE SUB ********************
3800 VV =  PEEK (37):HH =  PEEK (36): HTAB 1: VTAB 1
3810  PRINT "READ TEMP   "
3820  FOR I = 0 TO 15:TCUR(I) = 0: NEXT
3830  FOR JK = 0 TO 3:TKID(JK) = 0: NEXT
3840 TREF = 0
3850  FOR JT = 1 TO 4: REM  TAKE 4 AVG
3860  CALL  - 32090
3870  FOR JJ = 0 TO 15
3880 TCUR(JJ) = TCUR(JJ) + ADC%(JJ)
3890  IF JJ = 3 THEN TCUR(3) = TCUR(2): REM FIX BRKN TCPL
3900  NEXT JJ
3910 TREF = TREF + ADC%(16)
3920  NEXT JT
3930 TREF = (TREF / (4 * 8.19)) * 1.0305 - 274.6
3940 TCUR(3) = TCUR(2): REM   FIX             BRKN TC
3950  FOR JJ = 0 TO 15
3960 XQ = (TCUR(JJ) / (4095 * 995)) / 4
3970 XQ = XQ * 0.8583 - 6.2E - 6
3980 XQ = XQ + TZ(2,1) + TRF * (TZ(2,2) + TRF * (TZ(2,3) +
 TRF * TZ(2,4)))
3990 XZ = XQ - TZ(1,4)
4000 XQ = TZ(1,5) + XZ * (TZ(1,1) + XZ * (TZ(1,2) + XZ * T
Z(1,3)))
4010 XQ = XQ - TZ(3,JJ): REM  TC OFFSET
4020 TCUR(JJ) = XQ
4030 JK =  INT (JJ / 4): IF XQ > TKID(JK) THEN TKID(JK) =
XQ
4040 TKID(JK) =  INT (TKID(JK) * 100 + .5) / 100
4050  NEXT JJ
4060  VTAB 1: HTAB 1
4070  PRINT "                             "
4080  VTAB VV + 1: HTAB HH + 1
4090  RETURN
4100  REM  SET VALVES SUB  ******************************
4110 VCHK = 0
```

```
4120  FOR J = 0 TO NV - 1
4130  IF  PEEK ( - 16384) = 155 THEN  RETURN : REM CHK FOR
 'ESC'
4140 VT = VT(J):VA = VA(J)
4150  IF VT < 35 THEN  GOSUB 4450: GOTO 4220
4160 DV = VT - VA
4170  IF  ABS (DV) < 10 THEN  GOSUB 4480: GOTO 4210
4180  REM  HIGH FLOW (CUBIC FIT)
4190 X = EE(J) * VT / PSI + AP(J) * PSI
4200 MP = AZ(J) + BZ(J) * X + CZ(J) * X * X + DZ(J) * X *
X * X
4210  IF  ABS (DV) < 1.9 THEN 4280
4220  IF MP > MX THEN MP = MX
4230 MP =  INT (MP + .5)
4240  IF  ABS (MP - CUR(J)) < MP / 100 + 4 THEN 4280
4250  REM  SET VALVE
4260 VCHK = 1
4270  GOSUB 4660: REM  MOVE MOTOR
4280  NEXT J
4290  GOSUB 3550: REM  GET PV
4300  FOR J = 0 TO NV - 1
4310  IF 30 < VA(J) AND VA(J) < 40 THEN C1(J) = VA(J) / CU
R(J) / PSI
4320  NEXT J
4330 AP = PSI + BP * VSUM ^ 2
4340  GOSUB 4540: REM  PRN PVT
4350  REM  UPDATE CUBIC MULTIPLIER
4360  FOR J = 0 TO NV - 1: IF VA(J) < 35 THEN 4420
4370 X = .5 * (E1(J) + E2(J) * (AZ(J) - CUR(J))) / E3(J) ^
 3
4380 X =  LOG (X +  SQR (X * X + 1))
4390 X = E3(J) * ( EXP ( - X / 3) -  EXP (X / 3)) - CZ(J)
4400 EE = PSI * (X / 3 / DZ(J) - AP(J) * PSI) / VA(J)
4410 EE(J) = .8 * EE(J) + .2 * EE: REM  WEIGHT EE
4420  NEXT J
4430  IF VCHK THEN 4110: REM  NOT GOOD ENOUGH
4440  RETURN
4450  REM  SLOW SECTION
4460 MP = VT / (C1(J) * PSI)
4470  RETURN
4480  REM LINEARIZED CUBIC
4490 X = EE(J) * VT / PSI + AP(J) * PSI
4500 SLP = BZ(J) + 2 * CZ(J) * X + 3 * DZ(J) * X * X
4510 SLP = 0.8 * SLP / PSI
4520 MP = CUR(J) + SLP * DV
4530  RETURN
4540  REM  PRINT PVT INFO **********************
4550  PRINT : PRINT  INT (PSI * 100 + .5) / 100" PSI  "
4560  PRINT "REF TEMP: " INT (TREF * 10 + .5) / 10" C  "
4570  PRINT : PRINT "V";: HTAB 5: PRINT "TARG";: HTAB 14:
PRINT "MEASD";: HTAB 23: PRINT "TEMP"
4580  HTAB 5: PRINT "ML/MIN";: HTAB 14: PRINT "ML/MIN";: H
```

```
TAB 23: PRINT "DEG C"
4590  PRINT "=============================="
4600  FOR JJ = 0 TO NV - 1
4610  PRINT JJ + 1;: HTAB 5: PRINT VT(JJ)"    ";
4620  HTAB 14: PRINT VA(JJ)"     ";
4630  HTAB 23: PRINT TKID(JJ)"       "
4640  NEXT
4650  RETURN
4660  REM  MOTOR MOVER *******************************
4670  REM  1 REV=384 STEPS
4680  IF MP > MX THEN MP = MX
4690  IF MP < 0 THEN MP = 0
4700 MP =  INT (MP + .5)
4710 DIST = MP - CUR(J)
4720 DR =  SGN (DIST)
4730  PRINT "MOVE V"J + 1": "MP"/"CUR(J)
4740 DIST =  ABS (DIST)
4750 FUDGE = FUDGE + DIST
4760  IF DIST = 0 THEN  RETURN
4770  IF DR <  > CH(J) THEN DIST = DIST + HYST
4780 CH(J) = DR
4790  IF DR < 0 THEN DR = 0
4800 D1 = DIST:D2 = 0
4810  IF DIST > 255 THEN D1 = 255:D2 = DIST - 255
4820  IF D2 > MX - 255 THEN D2 = MX - 255
4830  POKE MM,J
4840  POKE MD,DR
4850  POKE MS,D1
4860  CALL MTR
4870  POKE MM,J
4880  POKE MD,DR
4890  POKE MS,D2
4900  CALL MTR
4910 CUR(J) = MP
4920  RETURN
4930  REM  INITIALIZE FLUID TYPE *****************
4940  RESTORE
4950  PRINT : PRINT "SELECT FLUID TYPE"
4960  PRINT : PRINT "1) WATER"
4970  PRINT "2) 80% ETHANOL"
4980  REM GETI$:I =  VAL (I$)
4990 I = 2
5000  ON I GOTO 5020,5050
5010  GOTO 4950
5020  PRINT "WATER"
5030 FM = 1.8:FB = 11.1
5040  GOTO 5070
5050  PRINT "ALCHOHOL"
5060 FM = 1.7:FB = 28.3
5070  REM  CALC FLOW-THETA COEFS
5080  FOR J = 0 TO 3
5090  READ AZ(J),BZ(J),CZ(J),DZ(J),AP(J)
```

```
5100 E1(J) = 2 * CZ(J) ^ 3 - 9 * BZ(J) * CZ(J) * DZ(J)
5110 E2(J) = 27 * DZ(J) ^ 2
5120 E3(J) =  SQR (3 * BZ(J) * DZ(J) - CZ(J) ^ 2)
5130  NEXT J
5140  PRINT "SETTING TC COEFFICIENTS"
5150  REM  TEMP COEF INFO FOR THERMOCOUPLES **************
*
5160  FOR JZ = 1 TO 2: FOR JY = 1 TO 5
5170  READ TZ(JZ,JY)
5180  NEXT : NEXT
5190  REM  READ OFFSET FOR TC'S
5200  FOR JY = 0 TO 15: READ TZ(3,JY): NEXT
5210  RETURN
5220  REM  DATA FOR FLOW-THETA FIT
5230  DATA  -20,19.9,-.682,.03223,.267
5240  DATA  -209,91.8,-6.92,.2024,.2
5250  DATA  -832,327,-37.5,1.493,.1
5260  DATA  -215,95.2,-9.45,.381,.167
5270  REM  TC COEFS
5280  DATA  25871.9,-664947,22291E3,-5.0686E-5,-1.29422
5290  DATA  1.78186E-7,3.86555E-5,3.9897E-8,1.94986E-11,0
5300  REM  OFFSET FOR TC'S FOLLOWS
5310  DATA  .85,.85,.8,0,1,1,1,1
5320  DATA  1,1,1,1,1,1.1,1.2,1.2
```

APPENDIX B

MACHINE CODE SUBROUTINES

The machine code subroutines were written in three blocks of code. First is a timer, followed by a stepper motor driver, and the last is an ADC driver program which reads the analog voltages and the frequency output of the flow meters. These programs are written for a 16 bit 'California Computer Systems' parallel interface in slot two, and a 12 bit 16 channel 'Applied Engineering' ADC in slot four. Although the programs are in three separate blocks of code, They were combined into one block of code (for ease of loading by Basic) named 'BIGN'. This block is 1204 bytes long, beginning at memory location 32768, and is simply BLOADed into place at the beginning the the Basic program.

B.1 Timer

This subroutine counts time in multiples of 5 seconds. The multiplier (POKEd by Basic before CALLing this routine) is at memory location 255 ($FF), and the elapsed time in seconds is displayed on the screen at the current cursor location. The variables LB and HB contain the low and high bytes of the current elapsed time in decimal mode. A separate subroutine, located at the end of this code, counts time in multiples of 0.05 seconds times the number at memory location 255. There is no display for this timer.

```
:ASM
                 **************************
                 *PROG 'TIMER'
                 * PROGRAM TO DELAY MULTIPLES
                 * OF 5 SEC.  MULTIPLIER IS PUT
                 * IN $255 AND STAYS INTACT.
                 *
                 * THE TIME IS OUTPUT TO THE
                 * SCREEN AT THE CURRENT CURSOR
                 * LOCATION, AND UPDATED EVERY
                 * SECOND.  THE TIMER MAY BE
                 * TERMINATED BY TYPING
                 * 'CTRL-A'
                 *
                 * A .05 SECOND TIMER IS AT
                 * THE END OF THE PROGRAM
                 **************************
                 *
                 LB       EQU  $7FF4       CURR
                 HB       EQU  $7FF5       TIME
                 TM       EQU  $FF          INPUT
                 XTMP     EQU  $7FF7
                 COUT     EQU  $FDED
                 CROUT    EQU  $FD8E
                 PRBYTE   EQU  $FDDA
                 WAIT     EQU  $FCA8
                 KBD      EQU  $C000
                 STRB     EQU  $C010
                 *
                          ORG  $8000
                          OBJ  $8000
                 *
8000: A5 FF      START    LDA  TM
8002: C9 00               CMP  #0
8004: F0 61               BEQ  END
8006: A9 00               LDA  #0
8008: 8D F4 7F            STA  LB
800B: 8D F5 7F            STA  HB
800E: A0 00               LDY  #0
8010: A2 00               LDX  #0
8012: AD F5 7F   TOP      LDA  HB          PRNT TIME
8015: 20 DA FD            JSR  PRBYTE
8018: AD F4 7F            LDA  LB
801B: 20 DA FD            JSR  PRBYTE
                 * BACK UP 4 SPACES
801E: A9 88               LDA  #$88        BK SPC
8020: 20 ED FD            JSR  COUT
8023: 20 ED FD            JSR  COUT
8026: 20 ED FD            JSR  COUT
8029: 20 ED FD            JSR  COUT
                 * CHECK FOR CTRL-A (ENDS TIMER)
```

```
802C: AD 00 C0             LDA   KBD
802F: C9 81                CMP   #129
8031: F0 34                BEQ   END
              * CHECK 5 SEC MULTIPLIER
8033: E0 05                CPX   #5
8035: D0 07                BNE   OK
8037: A2 00                LDX   #0
8039: C8                   INY
803A: C4 FF                CPY   TM
803C: F0 29                BEQ   END
803E: 8E F7 7F OK          STX   XTMP      STORE X
8041: A2 0A                LDX   #10       LOOP 10X
8043: A9 C7    WW          LDA   #199      .1 SEC
8045: 20 A8 FC             JSR   WAIT
8048: CA                   DEX
8049: E0 00                CPX   #0
804B: D0 F6                BNE   WW
804D: AE F7 7F             LDX   XTMP
              *
8050: E8                   INX
              *INCREMENT TIME COUNTER
8051: F8                   SED   SET       DEC FOR ADDIN
G
8052: 18                   CLC
8053: AD F4 7F             LDA   LB
8056: 69 01                ADC   #1
8058: 8D F4 7F             STA   LB
805B: AD F5 7F             LDA   HB
805E: 69 00                ADC   #0
8060: 8D F5 7F             STA   HB
8063: D8                   CLD   CLR       DEC
8064: 4C 12 80             JMP   TOP
8067: 20 8E FD END         JSR   CROUT
806A: A9 00                LDA   #0
806C: 8D 10 C0             STA   STRB       CLEAR KBD
806F: 60                   RTS
              *END OF 5 SECOND TIMER
              *
              *
              * PROGRAM TO DELAY MULTIPLES
              * OF .05 SEC.  MULTIPLIER IS PUT
              * IN $255 AND STAYS INTACT.
              *
8070: A6 FF    STRT        LDX   TM
8072: E0 00    TOP2        CPX   #0
8074: F0 09                BEQ   END2
8076: A9 8B                LDA   #139      .05 SEC
8078: 20 A8 FC             JSR   WAIT
807B: CA                   DEX
807C: 18                   CLC
807D: 90 F3                BCC   TOP2      ALWAY BRANCH
807F: 60       END2        RTS
```

```
--END ASSEMBLY--

ERRORS: 0

128 BYTES

SYMBOL TABLE - ALPHABETICAL ORDER:

   COUT     =$FDED      CROUT    =$FD8E      END      =$8067
   END2     =$807F      HB       =$7FF5      KBD      =$C000
   LB       =$7FF4      OK       =$803E      PRBYTE   =$FDDA
?  START    =$8000      STRB     =$C010   ?  STRT     =$8070
   TM       =$FF        TOP      =$8012      TOP2     =$8072
   WAIT     =$FCA8      WW       =$8043      XTMP     =$7FF7


SYMBOL TABLE - NUMERICAL ORDER:

   TM       =$FF        LB       =$7FF4      HB       =$7FF5
   XTMP     =$7FF7   ?  START    =$8000      TOP      =$8012
   OK       =$803E      WW       =$8043      END      =$8067
?  STRT     =$8070      TOP2     =$8072      END2     =$807F
   KBD      =$C000      STRB     =$C010      WAIT     =$FCA8
   CROUT    =$FD8E      PRBYTE   =$FDDA      COUT     =$FDED
```

B.2 Stepper Motor Driver

This program controls the movement of 8 stepper motors. It keeps track of the phase of each motor, and checks the range of all the inputs (motor number, direction of travel, and distance) before executing a movement. The motor number is at memory location 32762, the direction of travel (0=close, 1=open) at 32763, and the distance to travel at location 32764. The limit switches are continually checked as the valves are closing, but ignored when the valves are opening. Flags are set when either a limit switch closes unexpectedly (during a closing movement), or when the movement is terminated with a 'CTRL-C' from the keyboard. However, these flags are not used in the Basic program which calls this subroutine.

```
:ASM
                    ********************************
                    * PROG 'MOTOR'
                    * STEPPER MOTOR DRIVER PROGRAM
                    * MAY, 1988     JON B. ZAERR
                    *
                    * INPUT: DIRECTION, DISTANCE, MOTOR #
                    * CHECKS FOR LIM SWTCH, CTRL-C, MTR RN
G
                    *
                    *  X CONTAINS CURRENT MOTOR #
                    *  Y CONTAINS CURRENT PHASE POINTER
                    ********************************
                    *
                    MM       EQU  $7FFA       ;MOTOR #
                    MD       EQU  $7FFB       ;DIR 0=BACK,1
=FORW
                    MS       EQU  $7FFC       ;STEPS TO GO
                    PATM     EQU  $7FFD       ;PORT A TEMP
                    COUT     EQU  $FDED
                    CROUT    EQU  $FD8E
                    TMP      EQU  $7FF7
                    PA       EQU  $C0A0       ;PORT A
                    PB       EQU  $C0A2       ;PORT B
                    WAIT     EQU  $FCA8
                    PRBYTE   EQU  $FDDA
                    KBD      EQU  $C000
                    KSTRB    EQU  $C010
                    *
                    *
                             ORG  $819F
                             OBJ  $819F
                    *
819F: 80            DELT     HEX  80          ;INIT TM DEL
FOR STPING
                    *
81A0: AD FA 7F               LDA  MM
                    * CHECK RANGE OF MOTOR # (0 TO 7)
81A3: C9 00                  CMP  #0          ;M#-0
81A5: 10 04                  BPL  OKL
81A7: F0 02                  BEQ  OKL
81A9: A9 00                  LDA  #0          ;OTHERWISE SE
T TO 0
81AB: C9 08         OKL      CMP  #8          ;M#-8
81AD: 30 02                  BMI  OKH
81AF: A9 07                  LDA  #7
81B1: AA            OKH      TAX              ;SET M#
81B2: 8D FA 7F               STA  MM           ;SET MM TO A
CTUAL USED
                    *
81B5: 8D A2 C0               STA  PB          ;SELECT LIMIT
```

```
 SWITCH
81B8: 0A        46           ASL
81B9: 0A        47           ASL
81BA: 0A        48           ASL
81BB: 0A        49           ASL
81BC: 8D FD 7F  50           STA  PATM       ;UPPER NIBBLE
 IS M#
                51  * CHECK FOR 0 STEPS
81BF: AD FC 7F  52           LDA  MS
81C2: C9 00     53           CMP  #0
81C4: D0 0C     54           BNE  GOGO        ;CONTINUE IF
 <> 0 STEPS
81C6: 60        55           RTS             ;QUIT IF 0 ST
EPS
                56  *
                57  * WAIT FOR RELAY TO EQUILIBRATE
81C7: AD FD 7F  58           LDA  PATM
81CA: 8D A0 C0  59           STA  PA
81CD: A5 3D     60           LDA  61
81CF: 20 A8 FC  61           JSR  WAIT       ;10 MSEC
                62  *
                63  * GET PHASE
81D2: BD 58 82  64  GOGO     LDA  MPH,X
                65  * CHECK RANGE OF PHASE (0 TO 7)
81D5: C9 00     66           CMP  #0         ;MPH-0
81D7: 10 04     67           BPL  OK
81D9: F0 02     68           BEQ  OK
81DB: A9 01     69           LDA  #1         ;OTHERWISE SE
T TO 1
81DD: C9 08     70  OK       CMP  #8         ;MPH-8
81DF: 30 02     71           BMI  OK2
81E1: A9 01     72           LDA  #1
81E3: A8        73  OK2      TAY             ;UPDATE CURRE
NT PHASE
                74  *
                75  * DONE WITH ERROR CHECKING,
                76  * NOW MOVE MOTOR
                77  *
                78  *
81E4: AD FB 7F  79  TOP      LDA  MD         ;DIRECTION
81E7: C9 00     80           CMP  #0
81E9: F0 0A     81           BEQ  BKWD
                82  * FORWARDS
81EB: C8        83           INY             ;INC PHASE
81EC: C0 08     84           CPY  #8         ;Y-8
81EE: D0 0C     85           BNE  OUTPUT
81F0: A0 00     86           LDY  #0         ;SET BACK TO
0
81F2: 4C FC 81  87           JMP  OUTPUT
                88  *BACKWARDS
81F5: 88        89  BKWD     DEY
81F6: C0 FF     90           CPY  #$FF
```

```
81F8: D0 02     91              BNE   OUTPUT
81FA: A0 07     92              LDY   #7            ;SET BACK TO
7
81FC: B9 60 82 93   OUTPUT     LDA   PH,Y          ;PHASE
81FF: 0D FD 7F 94              ORA   PATM          ;MOTOR #
8202: 8D A0 C0 95              STA   PA            ;OUTPUT TO PO
RT A
                96   ******* DELAY SECTION   *****
8205: AD 9F 81 97              LDA   DELT          ;TIME DELAY
8208: 20 A8 FC 98              JSR   WAIT          ;DELAY
                99   *
                100  * DEC STPS TO GO, LIM SW, CTRL-C...
                101  *
820B: CE FC 7F 102             DEC   MS            ;#STPS
820E: AD 00 C0 103             LDA   KBD
8211: C9 C3    104             CMP   #$C3          ;CTRL-C??
8213: D0 0D    105             BNE   NOKEY
8215: A9 63    106             LDA   #99
8217: 8D FB 7F 107             STA   MD            ;SET FLAG
821A: A9 00    108             LDA   #0
821C: 8D 10 C0 109             STA   KSTRB         ;CLEAR KBD
821F: 4C 3E 82 110             JMP   QT            ;QUIT
8222: AD FC 7F 111  NOKEY      LDA   MS
8225: C9 00    112             CMP   #0            ;CHECK DISTAN
CE
8227: F0 15    113             BEQ   QT            ;QUIT IF AT E
ND
               114  * CHECK LIM SW   0=OPEN
               115  *                128=CLOSED
8229: AD A0 C0 116             LDA   PA
822C: 29 80    117             AND   #128
822E: C9 00    118             CMP   #0
8230: F0 B2    119             BEQ   TOP           ;CONTINUE STE
PPING
8232: AD FB 7F 120             LDA   MD            ;STOP IF GOIN
G DOWN...
8235: C9 00    121             CMP   #0
8237: D0 AB    122             BNE   TOP           ;CONTINUE IF
GOING UP
               123  *SO, THE MOTOR IS GOING DOWN, HASN'T
               124  * REACHED IT'S GOAL, AND THE LIMIT SWI
TCH IS
               125  * CLOSED.  THEREFORE, SET A FLAG @ MD
AND QUIT.
               126  *
8239: A9 FF    127             LDA   #$FF
823B: 8D FB 7F 128             STA   MD
823E: 98       129  QT         TYA                 ;SAVE PHASE
823F: AE FA 7F 130             LDX   MM            ;GET M#
8242: 9D 58 82 131             STA   MPH,X
               132  * WAIT FOR MOTOR TO SETTLE
8245: AD 9F 81 133             LDA   DELT
```

```
8248: 20 A8 FC 134          JSR  WAIT
824B: AD 9F 81 135          LDA  DELT
824E: 20 A8 FC 136          JSR  WAIT
               137  * TURN OFF PHASE $ M SELECT
8251: A9 70    138          LDA  #%01110000
8253: 8D A0 C0 139          STA  PA
8256: 60       140          RTS            ;NOW RETURN T
O MAIN PROG
8257: 00       141          BRK
               142  * PHASE POINTERS (FOR 8 MOTORS)
8258: 00 00 00
825B: 00       143  MPH     HEX  00000000  ;INITIAL PHAS
E POINTER
825C: 00 00 00
825F: 00       144          HEX  00000000
               145  * PHASE DATA (HALF STEPS)
8260: 0A 08 09
8263: 01       146  PH      HEX  0A080901
8264: 05 04 06
8267: 02       147          HEX  05040602
8268: EA       148          NOP
8269: 4C D0 03 149          JMP  $03D0     ;EXIT TO BASI
C IF GOOFED


--END ASSEMBLY--

ERRORS: 0

205 BYTES


SYMBOL TABLE - ALPHABETICAL ORDER:

   BKWD    =$81F5   ? COUT    =$FDED   ? CROUT   =$FD8E
   DELT    =$819F     GOGO    =$81D2     KBD     =$C000
   KSTRB   =$C010     MD      =$7FFB     MM      =$7FFA
   MPH     =$8258     MS      =$7FFC     NOKEY   =$8222
   OK      =$81DD     OK2     =$81E3     OKH     =$81B1
   OKL     =$81AB     OUTPUT  =$81FC     PA      =$C0A0
   PATM    =$7FFD     PB      =$C0A2     PH      =$8260
?  PRBYTE  =$FDDA     QT      =$823E   ? TMP     =$7FF7
   TOP     =$81E4     WAIT    =$FCA8


SYMBOL TABLE - NUMERICAL ORDER:

?  TMP     =$7FF7     MM      =$7FFA     MD      =$7FFB
   MS      =$7FFC     PATM    =$7FFD     DELT    =$819F
   OKL     =$81AB     OKH     =$81B1     GOGO    =$81D2
   OK      =$81DD     OK2     =$81E3     TOP     =$81E4
   BKWD    =$81F5     OUTPUT  =$81FC     NOKEY   =$8222
```

	QT	=$823E		MPH	=$8258		PH	=$8260
	KBD	=$C000		KSTRB	=$C010		PA	=$C0A0
	PB	=$C0A2		WAIT	=$FCA8	?	CROUT	=$FD8E
?	PRBYTE	=$FDDA	?	COUT	=$FDED			

B.3 ADC Driver

This code segment has two main entry points. The entry at -32090 converts all channels of the ADC inputs including the 16 extra channels from the thermocouple multiplexer. The results are loaded directly into the Basic integer array 'ADC%' in the order: Thermocouples 1-16 (ADC channel 0) in ADC%(0-15), ADC channels 1-15 in ADC%(16-30). The entry point at location -32095 is used to measure the frequency output of the flow meters. The results are loaded into the Basic integer array ADC%(32-39). The time base for the frequency measurements is 1.0 milliseconds, sampled 1000 times. At each time interval, all 8 flow meters are measured and the voltage levels compared with a dead-band (limits of dead-band at lines 217-220) to determine whether they count as high or low. The frequency counter is incremented only on low to high transitions of the flow meter output voltage. The gain for each analog input channel is in the array 'GAIN' at lines 203-206.

```
:ASM
*************************************
*PROGRAM  "CONV"
* NON-RELOCATABLE SUBROUTINES:
* TO READ ADC CHANNELS
* AND MEASURE FLOW METERS IN HZ
*   <<<<< BY JON B. ZAERR >>>>>
*           MAY, 1988
*
* "ADC" CONVERTS ALL 31 CHANNELS.
*  RESULT IS PUT IN THE BASIC
*  INTEGER ARRAY 'ADC%'(0-30)
*
* "VD" MEASURES 8 FLOW METERS.
*   RESULT IN 'ADC%'(32 TO 39)
*
* 'ADC%' MUST BE DIMENSIONED
* BEFORE THESE SUBROUTINES
* ARE USED.
************************************
*
*1ST AND 2ND LTRS OF NAME OF ARRAY FOR ADC INFO
N1       EQU  $C1
N2       EQU  $C4
PH       EQU  $6C         BSIC PNTR TO VAR
PL       EQU  $6B         " " LOW BYTE
CH       EQU  $7FFA       ;ADC BOARD PNTER
MUX      EQU  $7FFB       ;EXTERNAL MUX PNTR
* TEMP POINTER TO BASIC ARRAY
PLT      EQU  $7FFD       ;TMP LB
PHT      EQU  $7FFE       ;TMP HB
BASE     EQU  $7A00       ;BASE ADRS FOR PGO (FOR BASIC)
*
PA       EQU  $C0A0       ;PORT 2,A
PACOM    EQU  PA+1
PB       EQU  PA+2        ;PORT 2,B
PBCOM    EQU  PA+3
ADCLB    EQU  $C0C0       ;PORT 4
ADCHB    EQU  ADCLB+1     ;ADC HIGH BYTE
ADCST    EQU  ADCLB+2     ;STRT CONV ADRS
ADCCH    EQU  ADCLB+3     ;CHAN & GAIN ADRS
*
PRBYTE   EQU  $FDDA
```

```
                    RDKEY     EQU  $FDOC
                    COUT      EQU  $FDED
                    CROUT     EQU  $FD8E
                    WAIT      EQU  $FCA8
                    BLNK      EQU  $F948
                    *
                              ORG  $82A0
                              OBJ  $82A0
                    *
                    *
                    *
                    *
                    * SECTION TO PUT INFO FROM ADC
                    * INTO AN INTEGER ARRAY CALLED ADC%
                    * FIRST GET ADDRESS OF INTEGER ARRAY
                    *
82A0: 00            PROG      HEX  00           ;PROGRAM POIN
ER
                    * ENTRY POINT FOR V MEASUREMENT
                    *********   $82A1=-32095   ***********
82A1: A9 01         VD        LDA  #1
82A3: 8D A0 82                STA  PROG         ;SET PNTR FOR
 V
                    *ENTRY POINT FOR ADC CONVERSION
                    *********   $82A6=-32090   ***********
82A6: 20 9B 84      ADC       JSR  SAVE          SV BSC PNTRS
82A9: A5 6B                   LDA  PL
82AB: 8D FD 7F                STA  PLT          SV LB OF VARP
TR
                    * SET UP I/O
82AE: A9 00                   LDA  #0
82B0: 8D A2 C0                STA  PB
82B3: 8D A1 C0                STA  PACOM
82B6: A9 7F                   LDA  #127
82B8: 8D A0 C0                STA  PA
82BB: A9 24                   LDA  #36
82BD: 8D A1 C0                STA  PACOM
82C0: A9 F0                   LDA  #$F0
82C2: 8D A0 C0                STA  PA
82C5: A9 00                   LDA  #0
82C7: 8D A3 C0                STA  PBCOM
82CA: A9 FF                   LDA  #255
82CC: 8D A2 C0                STA  PB
82CF: A9 24                   LDA  #36
82D1: 8D A3 C0                STA  PBCOM
                    * DONE WITH I/O

                    * BEGIN SEARCH FOR ADC% ARRAY
82D4: A0 00         TOP       LDY  #0
82D6: B1 6B                   LDA  (PL),Y
82D8: C9 C1                   CMP  #N1
82DA: D0 09                   BNE  MORE
```

```
                 *FOUND 1ST,LOOK FOR 2ND LETTER
82DC: 20 FB 82            JSR   INC          INC POINTER
82DF: B1 6B               LDA   (PL),Y
82E1: C9 C4               CMP   #N2
82E3: F0 24               BEQ   FOUND        BRN IF HAVE
NAME
82E5: 20 FB 82   MORE     JSR   INC
82E8: A5 6B               LDA   PL
82EA: CD FD 7F            CMP   PLT          CMP W OLD
82ED: D0 E5               BNE   TOP          BACK FOR MOR
E
                 *COULDN'T FIND...
                 *SEND MSG
82EF: 20 48 F9            JSR   $F948        3BLNK
82F2: A9 B0               LDA   #$B0         A 0
82F4: 20 ED FD            JSR   COUT
82F7: 20 8E FD            JSR   CROUT
82FA: 60                  RTS
                 *INCREMENT POINTER
82FB: 18         INC      CLC
82FC: A5 6B               LDA   PL
82FE: 69 01               ADC   #1
8300: 85 6B               STA   PL
8302: A5 6C               LDA   PH
8304: 69 00               ADC   #0
8306: 85 6C               STA   PH
8308: 60                  RTS
                 *
                 *FOUND NAME, NOW OFFSET TO STRT OF INF
O
8309: 18         FOUND    CLC
830A: A5 6B               LDA   PL
830C: 69 06               ADC   #6           BASE OF ARRA
Y INFO
830E: 85 6B               STA   PL
8310: 8D FD 7F            STA   PLT         ;SAVE LB
8313: A5 6C               LDA   PH
8315: 69 00               ADC   #0
8317: 85 6C               STA   PH
8319: 8D FE 7F            STA   PHT         ;SAVE HB
                 *
                 * FOUND ARRAY, NOW CHECK WHICH PROG
831C: AD A0 82            LDA   PROG
831F: C9 01               CMP   #1
8321: D0 03               BNE   GUK         ;USE IF <>1
8323: 4C B8 83            JMP   VDB         ;USE IF = 1
                 *
                 ************************************
                 * SECTION TO CONVERT ALL CHANELS
                 * OF THE ADC
                 ************************************
```

```
8326: A9 00      GUK      LDA  #0
8328: 8D FA 7F            STA  CH
832B: 8D FB 7F            STA  MUX         ;INITIALIZE P
OINERS
                 * LOOP FOR MUX
832E: AD FB 7F   LPMUX    LDA  MUX
8331: 0A                  ASL
8332: 0A                  ASL
8333: 0A                  ASL
8334: 0A                  ASL
8335: 8D A2 C0            STA  PB          ;SET UP EXTER
NAL MUX
8338: 20 5C 83            JSR  STUFF       ;CONVERT & SA
VE DATA
833B: EE FB 7F            INC  MUX
833E: AD FB 7F            LDA  MUX
8341: C9 10               CMP  #16         ;LOOP 16 TIME
S
8343: D0 E9               BNE  LPMUX
                 *
                 * NOW LOOP FOR OTHER CHANNELS
8345: EE FA 7F            INC  CH
8348: 20 5C 83   LPCH     JSR  STUFF       ;CONV & SAVE
DATA
834B: EE FA 7F            INC  CH
834E: AD FA 7F            LDA  CH
8351: C9 10               CMP  #16
8353: D0 F3               BNE  LPCH        ;LOOP 16 TIME
S
8355: 20 A8 84            JSR  REST        ;REST BASIC P
NTRS
8358: 60                  RTS              ;ALL DONE - R
ETURN TO BASIC
8359: EA                  NOP
835A: EA                  NOP
835B: 00                  BRK
                 * SUBROUTINE TO SET UP ADC, CONVERT,
                 * AND PUT RESULT IN BASIC ARRAY
                 *
835C: AE FA 7F   STUFF    LDX  CH
835F: BD 93 83            LDA  GAIN,X
8362: 0A                  ASL
8363: 0A                  ASL
8364: 0A                  ASL
8365: 0A                  ASL
8366: 0D FA 7F            ORA  CH
8369: 8D C3 C0            STA  ADCCH       ;CH & GAIN
                 * ALLOW SETTLING TIME
836C: A9 0E               LDA  #14         ;SEEMS TO BE
STABLE
836E: 20 A8 FC            JSR  WAIT
                 *
```

```
8371: AD C2 C0 183          LDA  ADCST      ;START CONVER
SION
8374: AD C1 C0 184 REPT     LDA  ADCHB      ;CHECK STATUS
8377: 2A       185          ROL
8378: 2A       186          ROL
8379: B0 F9    187          BCS  REPT       ;REPEAT UNTIL
 DONE
               188 * NOW STUFF INTO ARRAY
837B: A0 00    189          LDY  #0
837D: AD C1 C0 190          LDA  ADCHB
8380: 29 0F    191          AND  #$0F       ;CLEAR UPPER
NIBBLE
8382: 91 6B    192          STA  (PL),Y
8384: 20 FB 82 193          JSR  INC        ;INCREMENT AR
RAY PNTR
8387: AD C0 C0 194          LDA  ADCLB
838A: 91 6B    195          STA  (PL),Y
838C: 20 FB 82 196          JSR  INC        ;INC TO NEXT
HB
838F: 60       197          RTS             ;DONE
               198 *
8390: EA       199          NOP
8391: EA       200          NOP
8392: 00       201          BRK
               202 *GAIN ARRAY FOR DIGITAL CONVERTER
8393: 03 04 00
8396: 00       203 GAIN     HEX  03040000
8397: 00 00 00
839A: 00       204          HEX  00000000
839B: 00 00 00
839E: 00       205          HEX  00000000
839F: 00 00 00
83A2: 00       206          HEX  00000000
               207
               208
               209 ***************************************
****
               210 * SECTION TO CALCULATE
               211 * HZ OF EACH OF 8 FLOW METERS
               212 ***************************************
****
               213
               214 * HYSTERYSIS LEVELS FOR
               215 * FREQUENCY MEASUREMENT
               216 * OF FLOW METER OUTPUT
               217 VDHBH    EQU  $07        ;UPPER LIM, H
B
               218 VDLBH    EQU  $8A        ;UPPER LIM, L
B
               219 VDHBL    EQU  $07        ;LOWER LIM, H
B
               220 VDLBL    EQU  $7A        ;LOWER LIM, L
```

B

```
                        *
83A3: 0A                TIME      HEX   0A          ;TIME DELAY I
N 1 MSEC LOOP
83A4: 00                CNL       HEX   00          ;LOOP COUNTER
 LB
83A5: 00                CNH       HEX   00          ;" " HB
83A6: 00                FLAG      HEX   00          ;INIT. FLAG
83A7: 00                NEW       HEX   00          ;NEW STATUS
83A8: 00 00 00
83AB: 00                OLD       HEX   00000000    ;OLD STATUS A
RRAY
83AC: 00 00 00
83AF: 00                          HEX   00000000
83B0: 00 00 00
83B3: 00                CNTR      HEX   00000000    ;TEMP V (HZ)
83B4: 00 00 00
83B7: 00                          HEX   00000000
                        *
83B8: A9 0A             VDB       LDA   #10         ;INIT LP COUN
TER
83BA: 8D A5 83                    STA   CNH
83BD: A9 64                       LDA   #100
83BF: 8D A4 83                    STA   CNL

                        * INIT STATE OF EACH VALVE
83C2: A9 01                       LDA   #1
83C4: 8D A6 83                    STA   FLAG        ;SET FLAG
83C7: 20 DB 83                    JSR   LPV
83CA: A9 00                       LDA   #0
83CC: 8D A6 83                    STA   FLAG        ;RESET FLAG

                        * INITIALIZE CNTR
83CF: A2 00                       LDX   #0
83D1: A9 00             LLL       LDA   #0
83D3: 9D B0 83                    STA   CNTR,X
83D6: E8                          INX
83D7: E0 08                       CPX   #8
83D9: D0 F6                       BNE   LLL

                        * LOOP 1000 TIMES, 1 MSEC EACH
83DB: A2 00             LPV       LDX   #0          ;INIT V INDEX
                        *LOOP FOR 8 METERS
83DD: 18                VTOP      CLC
83DE: 8A                          TXA
83DF: 69 08                       ADC   #8          ;SET ADC CHAN
EL
83E1: 8D C3 C0                    STA   ADCCH       ;CH & GAIN
83E4: AD C2 C0                    LDA   ADCST       ;START CONV
83E7: AD C1 C0          REPT2     LDA   ADCHB
83EA: 2A                          ROL
83EB: 2A                          ROL
```

```
83EC: B0 F9     263             BCS   REPT2        ;REP TIL ADC
DONE
                264   *
                265   * CHECK V>VDH & VOLD=0
83EE: AD C1 C0  266             LDA   ADCHB
83F1: 29 0F     267             AND   #$0F         ;CLEAR UPPER
NIBBLE
83F3: C9 07     268             CMP   #VDHBH
83F5: F0 04     269             BEQ   EQ1
83F7: 30 17     270             BMI   FAIL1
83F9: 10 08     271             BPL   GOOD1
                272   *HB MUST BE EQUAL
83FB: AD C0 C0  273   EQ1       LDA   ADCLB
83FE: 38        274             SEC
83FF: C9 8A     275             CMP   #VDLBH
8401: 90 0D     276             BCC   FAIL1
8403: A9 01     277   GOOD1     LDA   #1           ;CHECK OLD ST
ATUS
8405: 8D A7 83  278             STA   NEW          ;NEW STATUS
8408: DD A8 83  279             CMP   OLD,X
840B: D0 25     280             BNE   UPDATE       ;ONLY COUNT L
O-HI XITION
840D: 4C 3B 84  281             JMP   ANOTHER      ;NO CHANGE
                282
                283
                284   * CHECK V<VDL & VOLD =1
8410: AD C1 C0  285   FAIL1     LDA   ADCHB
8413: 29 0F     286             AND   #$0F         ;CLEAR UPPER
NIBBLE
8415: C9 07     287             CMP   #VDHBL
8417: F0 04     288             BEQ   EQ2
8419: 10 14     289             BPL   FAIL2
841B: 30 08     290             BMI   GOOD2
                291   *HB MUST BE EQUAL
841D: AD C0 C0  292   EQ2       LDA   ADCLB
8420: 38        293             SEC
8421: C9 7A     294             CMP   #VDLBL
8423: B0 0A     295             BCS   FAIL2
8425: A9 00     296   GOOD2     LDA   #0
8427: 8D A7 83  297             STA   NEW          ;UPDATE STATU
S
842A: DD A8 83  298             CMP   OLD,X
842D: D0 06     299             BNE   UPDATE2      ;DON'T COUNT
HI-LO XITION
842F: 4C 3B 84  300   FAIL2     JMP   ANOTHER
                301
                302
8432: FE B0 83  303   UPDATE    INC   CNTR,X       ;INC HZ
8435: AD A7 83  304   UPDATE2   LDA   NEW          ;UPDATE STATU
S
8438: 9D A8 83  305             STA   OLD,X
843B: E8        306   ANOTHER   INX
```

```
843C: E0 08                CPX   #8
843E: D0 9D                BNE   VTOP         ;LOOP 8X

           * CHECK IF CALLED BY INIT ROUTINE
8440: AD A6 83             LDA   FLAG
8443: C9 01                CMP   #1
8445: D0 01                BNE   DD
8447: 60                   RTS                ;RET IF CALLED BY INIT.
8448: EA           DD      NOP                ;GO HERE IF IN LOOP

           *
           * DONE WITH THAT ROUND, DELAY 1 MSEC
           *  & DO IT AGAIN 1000 TIMES
           *
8449: AD A3 83             LDA   TIME
844C: 20 A8 FC             JSR   WAIT         ;ABOUT 1 MSEC

           *UPDATE COUNTER
844F: CE A4 83             DEC   CNL
8452: AD A4 83             LDA   CNL
8455: C9 00                CMP   #0
8457: D0 3F                BNE   ABC
8459: A9 64                LDA   #100
845B: 8D A4 83             STA   CNL          ;RESET CNL
845E: CE A5 83             DEC   CNH
8461: AD A5 83             LDA   CNH
8464: C9 00                CMP   #0
8466: D0 30                BNE   ABC          ;LOOP 10X
           * DONE LOOPING

           * GET BASE ADDRESS OF V (DIGITAL)
           *  ELEMENTS 32-39 OF ADC% ARRAY

8468: 18                   CLC
8469: AD FD 7F             LDA   PLT          ;PNTR TO 0TH ELE.
846C: 69 40                ADC   #64          ;INC TO 32ND ELE.
846E: 85 6B                STA   PL
8470: AD FE 7F             LDA   PHT
8473: 69 00                ADC   #0
8475: 85 6C                STA   PH

8477: A2 00                LDX   #0           ;V INDEX
8479: A0 00                LDY   #0
847B: A9 00        ZZ      LDA   #0
847D: 91 6B                STA   (PL),Y       ;HB
847F: 20 FB 82             JSR   INC          ;INC TO LB
8482: BD B0 83             LDA   CNTR,X
8485: 91 6B                STA   (PL),Y
```

```
8487: 20 FB 82 355          JSR  INC          ;INC TO NEXT
HB
848A: E8       356          INX
848B: E0 08    357          CPX  #8
848D: D0 EC    358          BNE  ZZ
848F: A9 00    359          LDA  #0
8491: 8D A0 82 360          STA  PROG         ;RESET PROG P
NTR
8494: 20 A8 84 361          JSR  REST         ;RETRIEVE BAS
IC PNTRS
8497: 60       362          RTS               ;DONE *******
               363
8498: 4C DB 83 364 ABC      JMP  LPV          ;TO AVOID BRN
 > 1PG
               365
               366
               367
               368 * SUBROUTINES TO SAVE AND RETRIEVE
               369 * BASIC POINTERS TO HELP OUT BASIC
               370 * WHEN CALLING THESE PROCEDURES
               371 * SAVE PG ONE
849B: A2 60    372 SAVE     LDX  #$60
849D: B5 00    373 LPS      LDA  0,X
849F: 9D 00 7A 374          STA  BASE,X
84A2: E8       375          INX
84A3: E0 00    376          CPX  #0
84A5: D0 F6    377          BNE  LPS
84A7: 60       378          RTS
               379 *
               380 *RETRIEVE PG ONE
84A8: A2 60    381 REST     LDX  #$60
84AA: BD 00 7A 382 LPR      LDA  BASE,X
84AD: 95 00    383          STA  0,X
84AF: E8       384          INX
84B0: E0 00    385          CPX  #0
84B2: D0 F6    386          BNE  LPR
84B4: 60       387          RTS

--END ASSEMBLY--

ERRORS: 0

533 BYTES


SYMBOL TABLE - ALPHABETICAL ORDER:

   ABC      =$8498   ?  ADC      =$82A6      ADCCH    =$C0C3
   ADCHB    =$C0C1      ADCLB    =$C0C0      ADCST    =$C0C2
   ANOTHER  =$843B      BASE     =$7A00   ?  BLNK     =$F948
   CH       =$7FFA      CNH      =$83A5      CNL      =$83A4
```

```
  CNTR     =$83B0     COUT     =$FDED     CROUT    =$FD8E
  DD       =$8448     EQ1      =$83FB     EQ2      =$841D
  FAIL1    =$8410     FAIL2    =$842F     FLAG     =$83A6
  FOUND    =$8309     GAIN     =$8393     GOOD1    =$8403
  GOOD2    =$8425     GUK      =$8326     INC      =$82FB
  LLL      =$83D1     LPCH     =$8348     LPMUX    =$832E
  LPR      =$84AA     LPS      =$849D     LPV      =$83DB
  MORE     =$82E5     MUX      =$7FFB     N1       =$C1
  N2       =$C4       NEW      =$83A7     OLD      =$83A8
  PA       =$C0A0     PACOM    =$C0A1     PB       =$C0A2
  PBCOM    =$C0A3     PH       =$6C       PHT      =$7FFE
  PL       =$6B       PLT      =$7FFD   ? PRBYTE   =$FDDA
  PROG     =$82A0   ? RDKEY    =$FD0C     REPT     =$8374
  REPT2    =$83E7     REST     =$84A8     SAVE     =$849B
  STUFF    =$835C     TIME     =$83A3     TOP      =$82D4
  UPDATE   =$8432     UPDATE2  =$8435   ? VD       =$82A1
  VDB      =$83B8     VDHBH    =$07       VDHBL    =$07
  VDLBH    =$8A       VDLBL    =$7A       VTOP     =$83DD
  WAIT     =$FCA8     ZZ       =$847B
```

```
SYMBOL TABLE - NUMERICAL ORDER:

  VDHBH    =$07       VDHBL    =$07       PL       =$6B
  PH       =$6C       VDLBL    =$7A       VDLBH    =$8A
  N1       =$C1       N2       =$C4       BASE     =$7A00
  CH       =$7FFA     MUX      =$7FFB     PLT      =$7FFD
  PHT      =$7FFE     PROG     =$82A0   ? VD       =$82A1
? ADC      =$82A6     TOP      =$82D4     MORE     =$82E5
  INC      =$82FB     FOUND    =$8309     GUK      =$8326
  LPMUX    =$832E     LPCH     =$8348     STUFF    =$835C
  REPT     =$8374     GAIN     =$8393     TIME     =$83A3
  CNL      =$83A4     CNH      =$83A5     FLAG     =$83A6
  NEW      =$83A7     OLD      =$83A8     CNTR     =$83B0
  VDB      =$83B8     LLL      =$83D1     LPV      =$83DB
  VTOP     =$83DD     REPT2    =$83E7     EQ1      =$83FB
  GOOD1    =$8403     FAIL1    =$8410     EQ2      =$841D
  GOOD2    =$8425     FAIL2    =$842F     UPDATE   =$8432
  UPDATE2  =$8435     ANOTHER  =$843B     DD       =$8448
  ZZ       =$847B     ABC      =$8498     SAVE     =$849B
  LPS      =$849D     REST     =$84A8     LPR      =$84AA
  PA       =$C0A0     PACOM    =$C0A1     PB       =$C0A2
  PBCOM    =$C0A3     ADCLB    =$C0C0     ADCHB    =$C0C1
  ADCST    =$C0C2     ADCCH    =$C0C3   ? BLNK     =$F948
  WAIT     =$FCA8   ? RDKEY    =$FD0C     CROUT    =$FD8E
? PRBYTE   =$FDDA     COUT     =$FDED
```

APPENDIX C

ELECTRONIC COMPONENTS AND CONNECTIONS

The connections between the computer interface cards and the external circuits described in sections 2.3, 2.4, 2.5.1, and 2.7 are given in this section. The physical location on the circuit board of each component is shown in Figure C-1. These components had the following functions:

Component ID	Description
IC1	AD590JH (Temperature sensor)
IC2	SN74150N (Data Selector)
IC3	MUX-16 (Analog multiplexer)
IC4	D75FY4D (PNP array)
IC5	AMP01FX (Instrumenation amplifier)
IC6	μA79M05C (-5V regulator)
IC7	LS138N (Decoder)
IC8	74F150N (Inverter)
R1	10k SIP
R2	1k, 1%
R3	4.7k SIP
R4	22k SIP
R5	10k DIP
R6	4.7k DIP
R7	380 Ω
R8	33 Ω
R9	10k DIP
RS	10k
RG	200 Ω
C1	10 μF
C1	0.1 μF
Q1	2N2222 (NPN array)
Q2	D41K (PNP)
Q3	2N2222 (NPN)
D1	1N4001
J1	25 Pin, Sub-D connector
J2	25 Pin, Sub-D connector
J3	25 Pin, Sub-D connector
J4	15 Pin, Sub-D connector
J5	24 Pin, Molex connector
J6	4 Pin, Molex connector

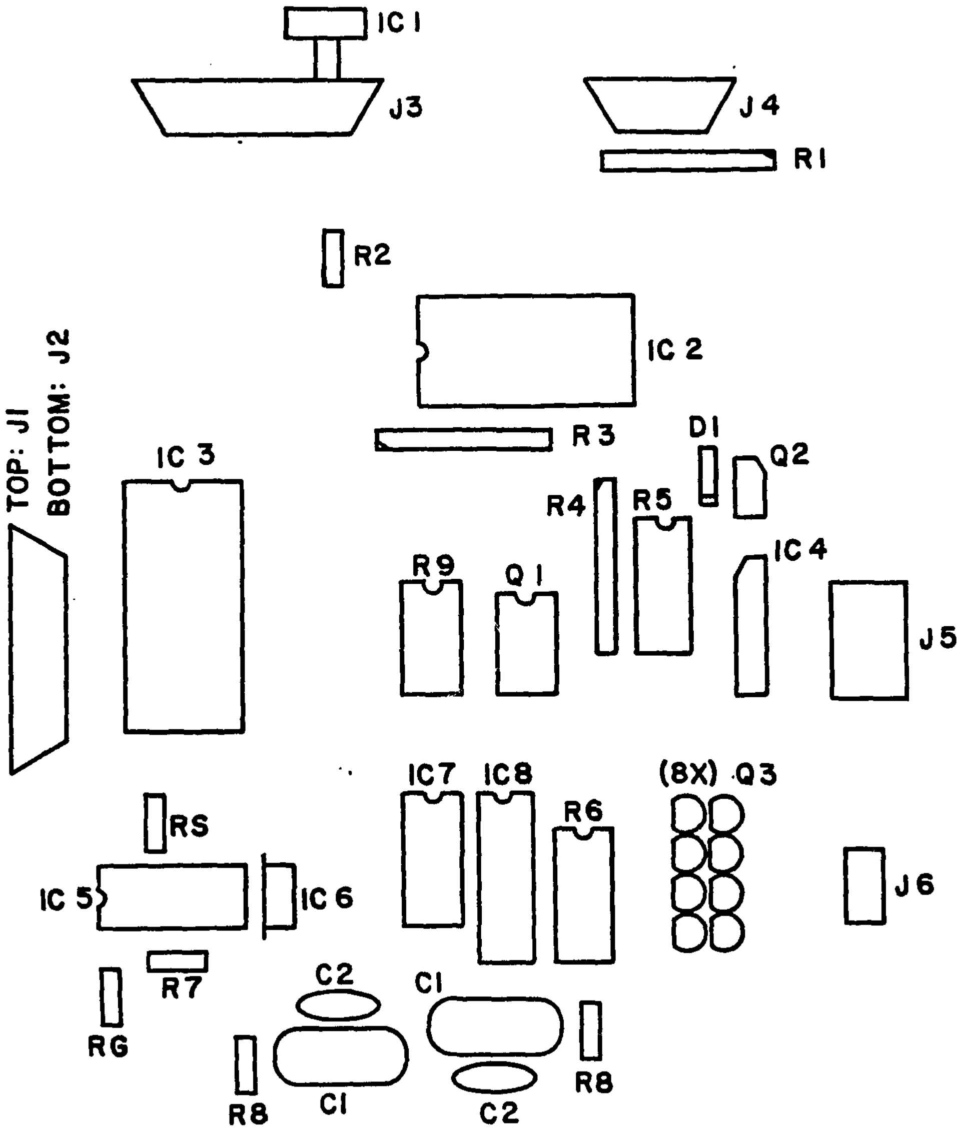


Figure C-1 Component Location. The physical location of the electronic components and interface connections in the external interface.

Several connectors, labeled J1 through J6 were used to provide power to and interface with the external interface. The function of each pin of these connectors follows.

The analog cable connected signals from the external interface to the ADC card in the Apple computer.

J1: Analog (25 Pin, Sub-D)

Pin number	Function
1	ADC channel 0, thermocouples
2	ADC channel 2, pressure
3	ADC channel 4
4	ADC channel 6
5-12	Gnd
13	nc
14	ADC channel 1, IC temperature sensor
15	ADC channel 3
16	ADC channel 5
17	ADC channel 7
18	ADC channel 8, flow meter 1
19	ADC channel 9, flow meter 2
20	ADC channel 10, flow meter 3
21	ADC channel 11, flow meter 4
22	ADC channel 12, flow meter 5
23	ADC channel 13, flow meter 6
24	ADC channel 14, flow meter 7
25	ADC channel 15, flow meter 8

The Parallel cable connected the parallel I/O card in the Apple computer with the external interface.

J2: Parrallel I/O (25 Pin, Sub-D)

Pin number	Function
1	Gnd
2	nc
3	Port B, Bit 6, Thermocouple MUX 2
4	Port B, Bit 4, Thermocouple MUX 0
5	Port B, Bit 2, Data Select 2
6	Port B, Bit 0, Data Select 0
7	Gnd
8	nc
9	Port A, Bit 6, Motor Select 2
10	Port A, Bit 4, Motor Select 0
11	Port A, Bit 2, Motor Phase $\phi 3$
12	Port A, Bit 0, Motor Phase $\phi 1$

13 .... Gnd
14 .... Gnd
15 .... nc
16 .... Port B, Bit 7, Thermocouple MUX 3
17 .... Port B, Bit 5, Thermocouple MUX 1
18 .... Port B, Bit 3, Data Select 3
19 .... Port B, Bit 1, Data Select 1
20 .... Gnd
21 .... nc
22 .... Port A, Bit 7, Data Out
23 .... Port A, Bit 5, Motor Select 1
24 .... Port A, Bit 3, Motor Phase ϕ4
25 .... Port A, Bit 1, Motor Phase ϕ2

The thermocouple cable connected the bare-wire thermocouples in the phantom to the analog multiplexer in the external interface.

J3: Thermocouples (25 Pin, Sub-D)

Pin number	Function
1	Thermocouple 1
2	Thermocouple 2
3	Thermocouple 3
4	Thermocouple 4
5	Thermocouple 5
6	Thermocouple 6
7	Thermocouple 7
8	Thermocouple 8
9	Thermocouple 9
10	Thermocouple 10
11	Thermocouple 11
12	Thermocouple 12
13	Thermocouple 13
14	Thermocouple 14
15	Thermocouple 15
16	Thermocouple 16
17	5V (to temperature sensor)
18	Temperature sensor output
19-24	nc
25	Gnd

The pressure and flow cable connected the flow meters and pressure transducer on the phantom to the external interface.

J4: Flow Meters, Pressure (15 Pin, Sub-D)

Pin number	Function
1	Flow Meter 1
2	Flow Meter 2
3	Flow Meter 3
4	Flow Meter 4
5	Flow Meter 5
6	Flow Meter 6
7	Flow Meter 7
8	Flow Meter 8
9	+14 Volts (to pressure transducer)
10	Gnd
11	Pressure transducer output
12	Gnd
13	nc
14	nc
15	+5 Volts (to flow meters)

The stepper motors and limit switches were connected to the external interface with the stepper motor cable.

J5: Stepper Motors (24 Pin, Molex)

Pin number	Function
1	Motor $\phi 1$
2	Motor $\phi 2$
3	Motor $\phi 3$
4	Motor $\phi 4$
5	Motor Select 1
6	Motor Select 2
7	Motor Select 3
8	Motor Select 4
9	Motor Select 5
10	Motor Select 6
11	Motor Select 7
12	Motor Select 8
13	Limit Switch 1
14	Limit Switch 2
15	Limit Switch 3
16	Limit Switch 4
17	Limit Switch 5
18	Limit Switch 6
19	Limit Switch 7
20	Limit Switch 8
21	+14 Volts
22	nc
23	nc
24	Gnd

The power cable provided 3 voltages to the external interface from a power supply.

J6: Power (4 Pin, Molex)

Pin number	Function
1	Gnd
2	+5 Volts
3	+14 Volts
4	-14 Volts

APPENDIX D

PARTS SOURCE

A list of specialized components in the phantom is given here. All parts not listed here are readily available at electronics, plumbing or hardware stores.

Part	Source
Computer Interface Cards:	
12 Bit, 16 Channel ADC	Applied Engineering, Carrollton, TX
16 Bit Parallel I/O	California Computer Systems, Sunnyvale, CA
Electronic Components:	
D75FY4D,	
AMP-01,	
MUX-16,	
AD590JH	Newark Electronics
General Items:	
PVC (3/8", Clear sheet)	Antex Plastics, Phoenix, AZ
PVC tubing (1/4" ID,	
reinforced, clear)	Cole-Parmer
Gear Pump (Variable Speed)	Cole-Parmer
Pressure Transducer	SenSym, Sunnyvale, CA
12 Tooth Pulley,	
48 Tooth Pulley,	
60 Tooth Belt	King Bearing, Tucson, AZ
Flow Meter	Digital Precision Flowmeter, Inc., Largo, FL

APPENDIX E

PID PARAMETER OPTIMIZATION

E.1 Procedure for PID Parameter Optimization

In this study, the operating parameters of the PID controller developed by Lin, Roemer and Hynynen (1989) were investigated. The effect of gain and sampling interval on controller stability were studied at gains of 0.1, 0.3, 0.7, 1.0 and 1.5 at sampling intervals of 10, 15, 25, and 40 seconds. This entire set of permutations was performed at perfusions of 0, 2.3 and 4.5 kg/m^3 sec. The maximum power available to the controller was 24 Watts acoustic power for perfusions less than 4.5, and 38 Watts for perfusion equal to 4.5. The nine PID controller regions were located as shown in Figure 3-8 and the scan pattern of Figure 3-1 was used for heating, as in Chapter Three.

For each combination of gain, sampling time and perfusion, the following procedure was performed. The prescribed flow rates were attained by the Apple computer,and the phantom temperature was allowed to equilibrate. Then a 6 °C (nominal) step in target temperature was applied to the controller. Temperature measurements were recorded simultaneously with the sampling interval used by the controller. Heating continued until the temperature achieved steady state, or until the temperature fluctuations were regular (as in the case of marginally stable controllers).

From these temperature profiles, the percent overshoot, rise time, settling time and rms (root mean square at steady state) temperature were calculated and used to evaluate the optimal operation point of the controller. The rms temperature was calculated based on the average temperature over the interval from the settling time to the end of the heating time. The percent overshoot was based on the actual temperature step from steady state to the target temperature. For over-damped profiles, the rise time was calculated as the first point at which the temperature rise was within 5% of the target temperature step. For under-damped profiles, the rise time was the last point before the peak where the temperature rise was 5% below the target temperature step. The settling time was simply the time at which the temperature subsequently stayed within 5% of the target temperature step.

One of the difficulties in determining rise time and settling time was large fluctuations in some temperature profiles. These fluctuations were due in part to the 'seasoning' effect of asynchronous power scanning and temperature measurement (Moros, Roemer and Hynynen 1988, and Lin, Roemer and Hynynen 1989), but also, in some instances, to the instability of the controller. To alleviate this problem, a smoothing filter was applied to the data before calculating the rise and settling times. An example of this is shown in Figure E-1, where the raw temperature data corresponded to a perfusion of 2.3 kg/m^3 sec, a controller gain of 0.7, and the sampling

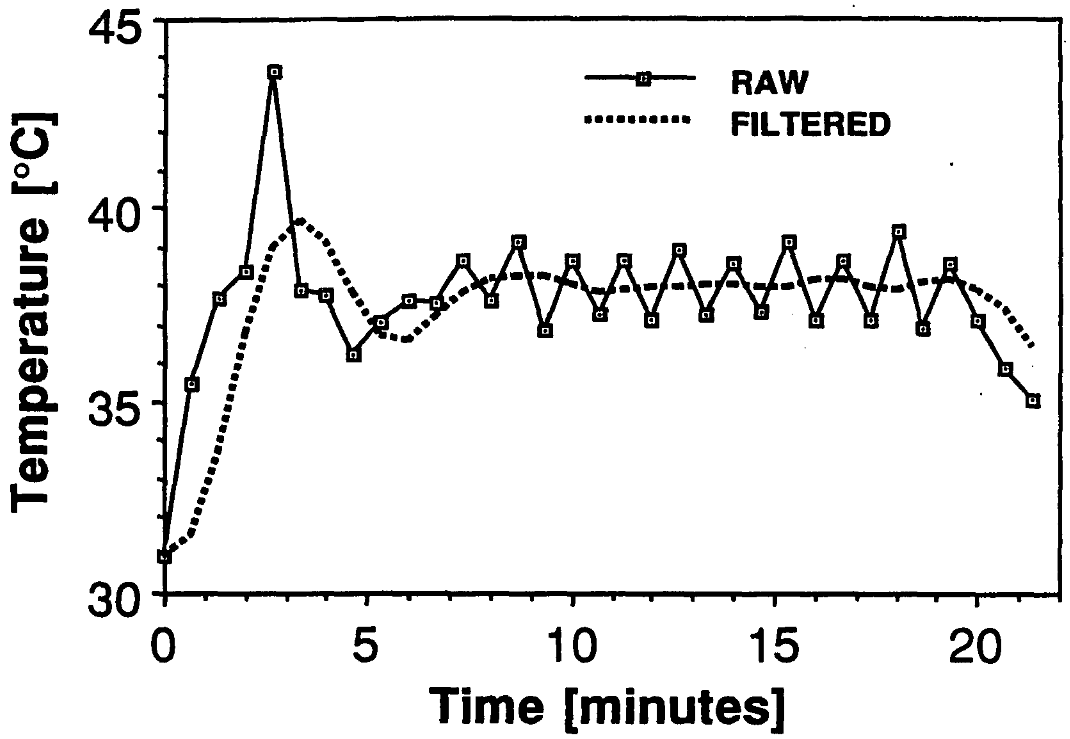


Figure E-1 Effect of Filtering. A digital 3rd order Butterworth filter was used to smooth the temperature responses produced by the PID controller prior to analysis. The PID parameters here were Gain = 0.7 and sampling interval = 40 seconds.

interval at 40 seconds. The filtered version exhibits a smoother response, enabling a more consistent evaluation of the settling time. The filter was a digital IIR (infinite impulse response) version of a third order low pass Butterworth filter (see Appendix F for development). It had 3 recursive terms and 4 nonrecursive terms as follows.

$$T_f(n) = c_0T(n) + c_1T(n-1) + c_2T(n-2) + c_3T(n-3) + d_1T_f(n-1) + d_2T_f(n-2) + d_3T_f(n-3)$$

where:

$T(n)$ = original temperature data at time n
$T_f(n)$ = filtered temperature data at time n
c_i = non-recursive filter coefficients
d_i = recursive filter coefficients

The coefficients c and d were determined by the sampling interval and the desired cut-off frequency. This digital filtering technique is reasonable since the hyperthermia temperature controller program itself implemented a low pass digital filter when calculating the required power intensity. This filter, denoted ARMA (auto regressive moving average), was used to smooth variations in the temperature to yield a more stable signal. The ARMA filter had the following form:

$$T_f(n) = .35T(n) + .3T(n-1) + .2T(n-2) + .15T(n-3)$$

Since the sampling interval did not enter into this equation, the cut-off frequency (attenuation=.707) was fixed at about 0.124 times the sampling frequency. The Butterworth filter implemented in this analysis had a variable cut-off frequency as shown in Table E-1. The cut-off frequencies for the Butterworth filter were chosen as high as possible to retain the characteristics of the original data, but low enough to filter out the harmonics produced by asynchronous scanning and temperature sampling. Note that in all cases the cut-off for the Butterworth filter was higher (f_c/f_s is higher) than the ARMA filter, which means the former is affecting the data less at low frequencies, and thus preserving more of the desired characteristics. A comparison of the normalized frequency response for these two filters is shown in Figure E-2 for a case where f_c/f_s=.130 for the Butterworth filter. On the frequency scale, 1 corresponds to the Nyquist sampling rate (the highest frequency obtainable with the given sampling rate). Since the Butterworth filter was derived using recursive sampling, it has a sharper transition between the pass and stop bands.

E.2 Results for PID Parameter Optimization

The qualities sought in an optimal controller were a minimal percent overshoot, a rapid settling time, and a small RMS temperature. Figure E-3 shows the percent overshoot (POS) at all 3 perfusion levels. At perfusions of zero and 2.5 kg/m^3 sec, the minimum value of POS was obtained at a gain of 0.3 and a sampling interval of 10 seconds. At 5 kg/m^3 sec, the minimum

filter type	sampling interval [sec]	sampling frequency f_s [Hz]	filter cut-off f_c [Hz]	f_c/f_s
ARMA	10	.1	.0124	.124
ARMA	15	.0667	.00826	.124
ARMA	25	.04	.00496	.124
ARMA	40	.025	.00310	.124
Btwth	10	.1	.013	.130
Btwth	15	.0667	.011	.165
Btwth	25	.04	.009	.225
Btwth	40	.025	.0055	.220

Table E-1 Comparison of Filter Cut-off Frequency. The ratio of cut-off frequency to sampling frequency was constant for the ARMA filter, but variable for the Butterworth filter.

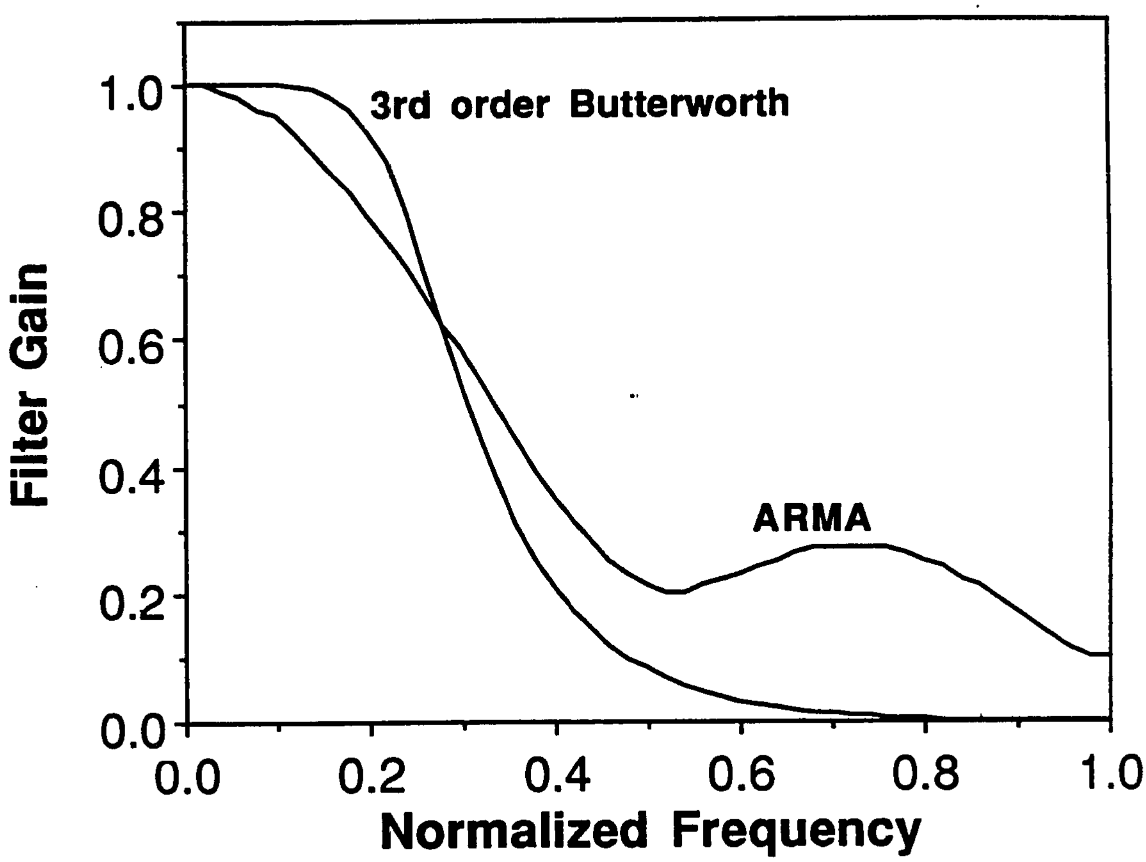


Figure E-2 Comparison of Filter Frequency Response. The PID controller applied an Auto Regressive Moving Average filter to temperature data before computing the required power. The Butterworth filter used in the analysis here filtered at a higher frequency than the ARMA filter, thus preserving more characteristics of the data.

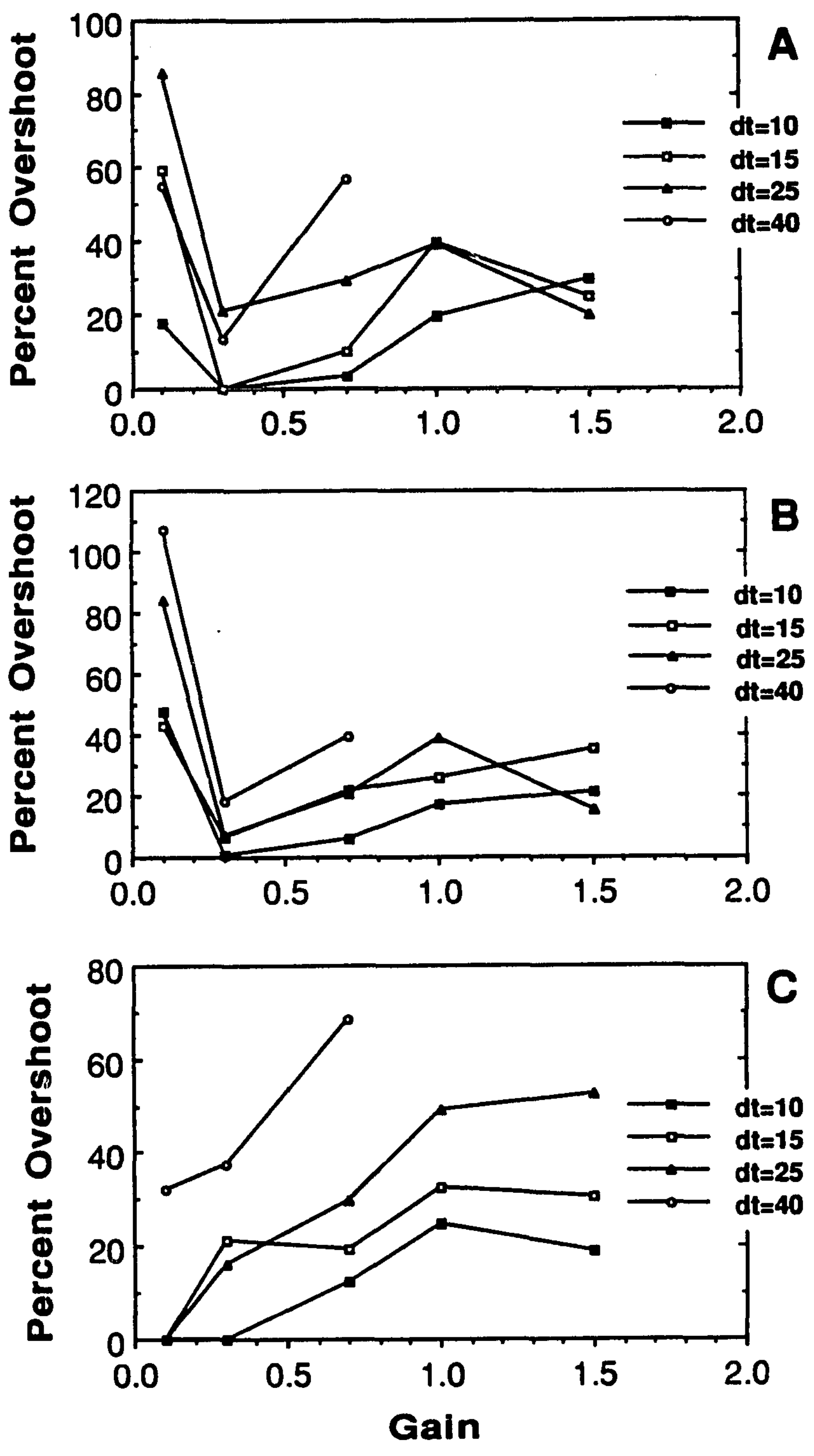


Figure E-3 Percent Overshoot of the PID Hyperthermia Controller. A) W = 0. B) W = 2.5 kg/m3 sec. C) W = 5 kg/m3 sec.

POS was at a gain of 0.1 and a sampling interval of 10, 15 or 25 seconds. The settling time (t_s) is shown in Figure E-4. At zero perfusion, the t_s had a broad minimum at a gain of 0.7, and a sampling interval of 10 seconds. At 2.5 kg/m^3 sec, the minimum t_s was at a gain of 0.3 and a sampling interval of 10 seconds. At 5 kg/m^3 sec, the minimum t_s was at a gain of 0.1, and a sampling interval of 25 seconds. The third criterion, RMS temperature at steady state (RMST), is shown in Figure E-5. At zero perfusion, the lowest RMST was at a gain of 1.0 and a sampling interval of 25 seconds. At 2.5 and 5 kg/m^3 sec, the minimum RMST was at a gain of 0.1 and a sampling interval of 25 seconds.

Clearly, the optimal controller paramters derived from the three criteria are not in agreement. The optimal gain ranged from 0.1 to 1.0 and the optimal sampling interval ranged from 10 to 25 seconds. One of the reasons of for these inconsistencies was experimental variations. Due to time constraints, only one repetition was performed for each case. Thus random factors were not well known. One way of diminishing the effect of this experimental error was to combine the criteria together in a weighted fashion. The weights for this scheme were as follows.

$$w_1 = .5/L_1 = 0.010$$
$$w_2 = 1/L_2 = 0.00182$$
$$w_3 = 1/L_3 = 2.0$$
$$wc = w_1*(\%os) + w_2*(t_{settle}) + w_3*(\text{rms temp})$$

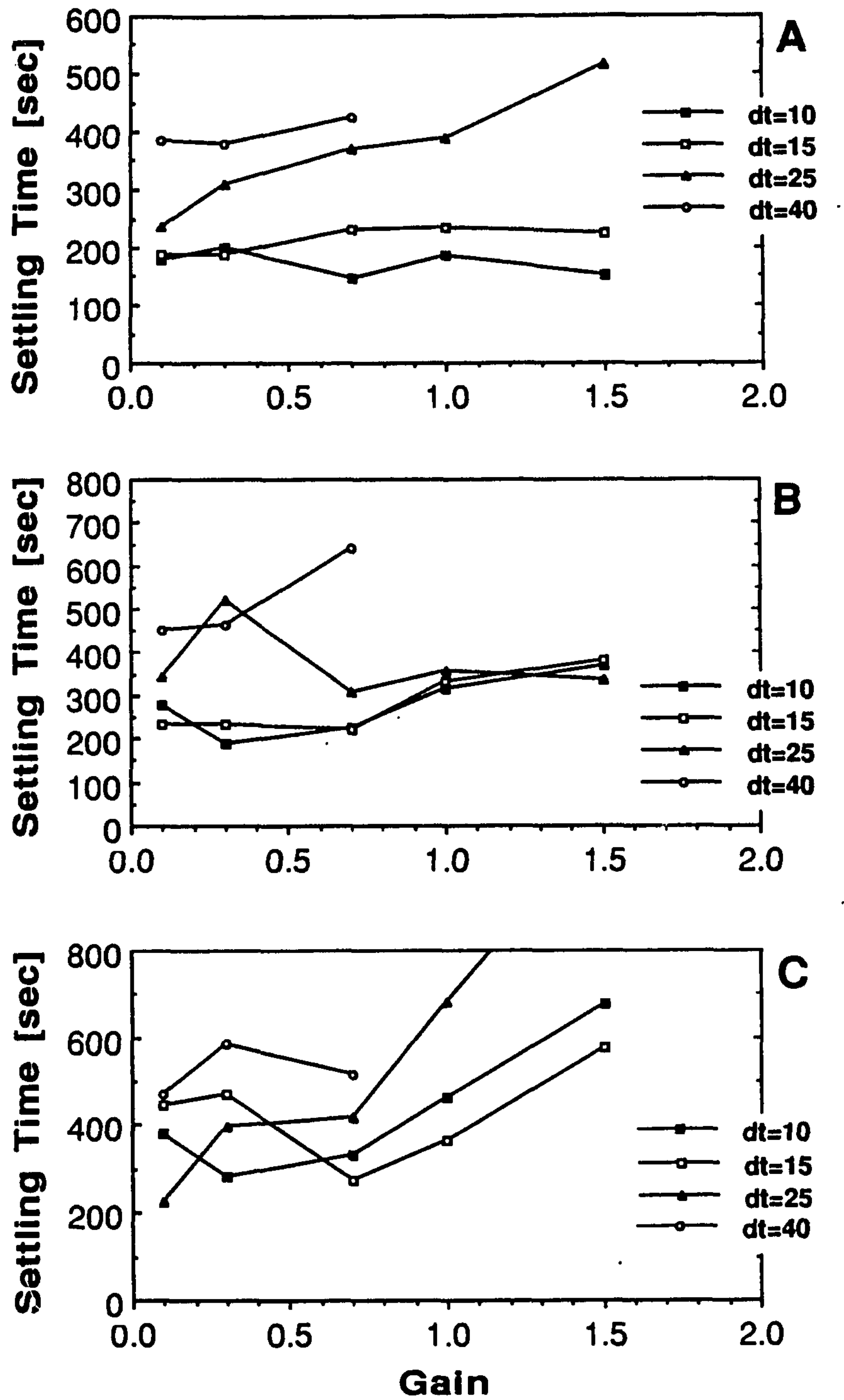


Figure E-4 Settling Time of the PID Hyperthermia Controller. A) W = 0. B) W = 2.5 kg/m³ sec. C) W = 5 kg/m³ sec.

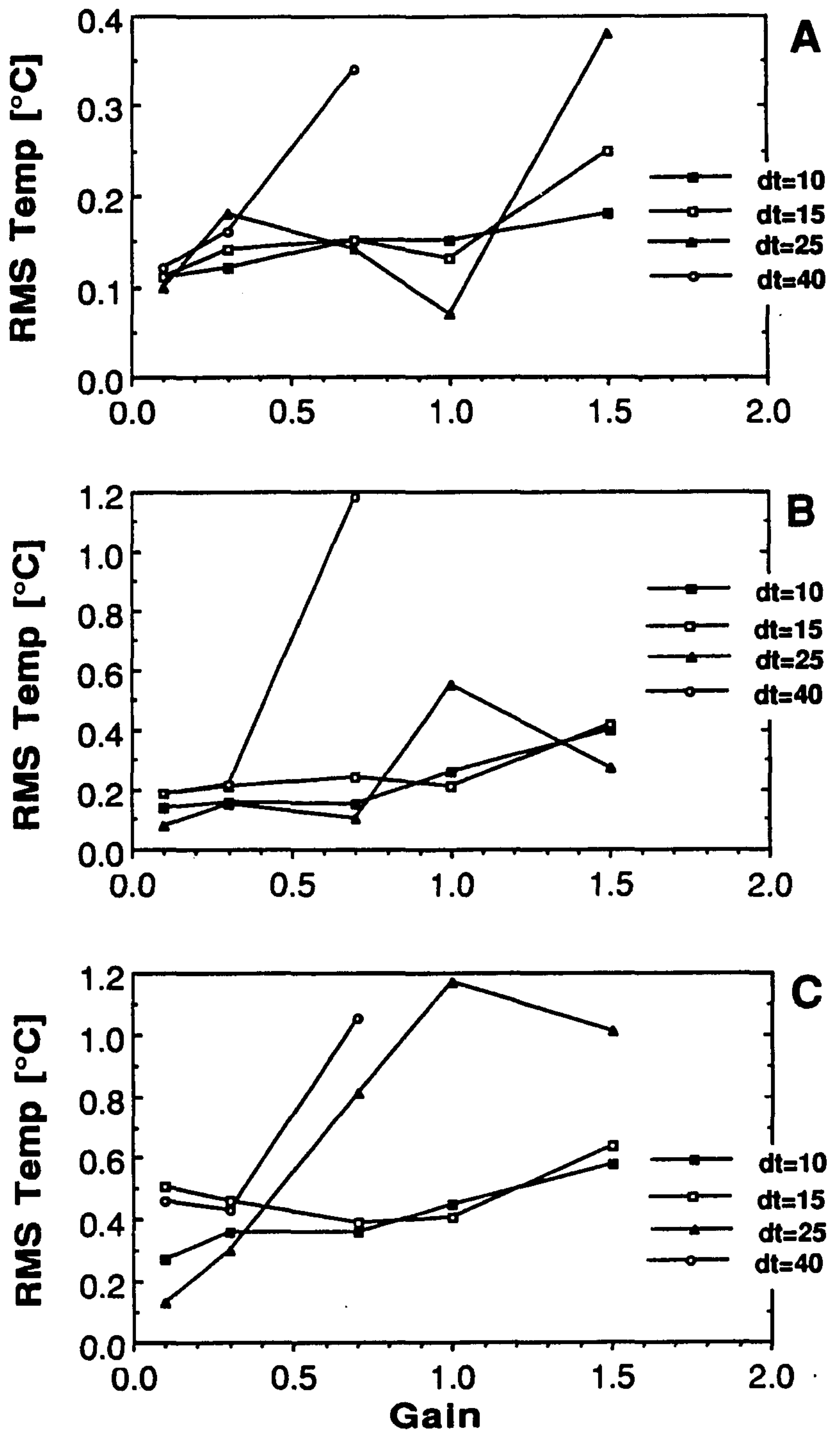


Figure E-5 RMS Temperature of the PID Hyperthermia Controller.
A) W = 0. B) W = 2.5 kg/m³ sec. C) W = 5 kg/m³ sec.

where:

L_1 = maximum acceptable limit for percent overshoot
L_2 = maximum acceptable limit for settling time
L_3 = maximum acceptable limit for rms temperature
wc = weighted criteria

The weight for the percent overshoot was given half the weight of the others since it is not only a function of the controller parameters, but also of the available power. Thus, the settling time and rms temperature were normalized to 1, and the percent overshoot was normalized to 1/2. Rise time was not used at all since it is only a function of available power and perfusion and not of the controller parameters.

A plot of the weighted criteria vs. controller gain at various sampling intervals is shown in Figure E-6 for the three perfusion levels. From these combined criteria, the optimal controller parameters were gain = .3, and sampling interval = 10 seconds. For the high perfusion case, the parameters gain = .1 and sampling interval = 25 seconds minimized the weighted criteria, but that point was probably an anomaly since it does not follow the trend of increasing weighted criteria with increasing sampling interval at constant gain. These values are close to the result obtained by Lin and Hynynin (1989), which was a gain of 0.3 and a sampling interval of 12 seconds. Thus the in vitro results obtained with the dynamic phantom substantiate the computer simulations for optimal controller parameters.

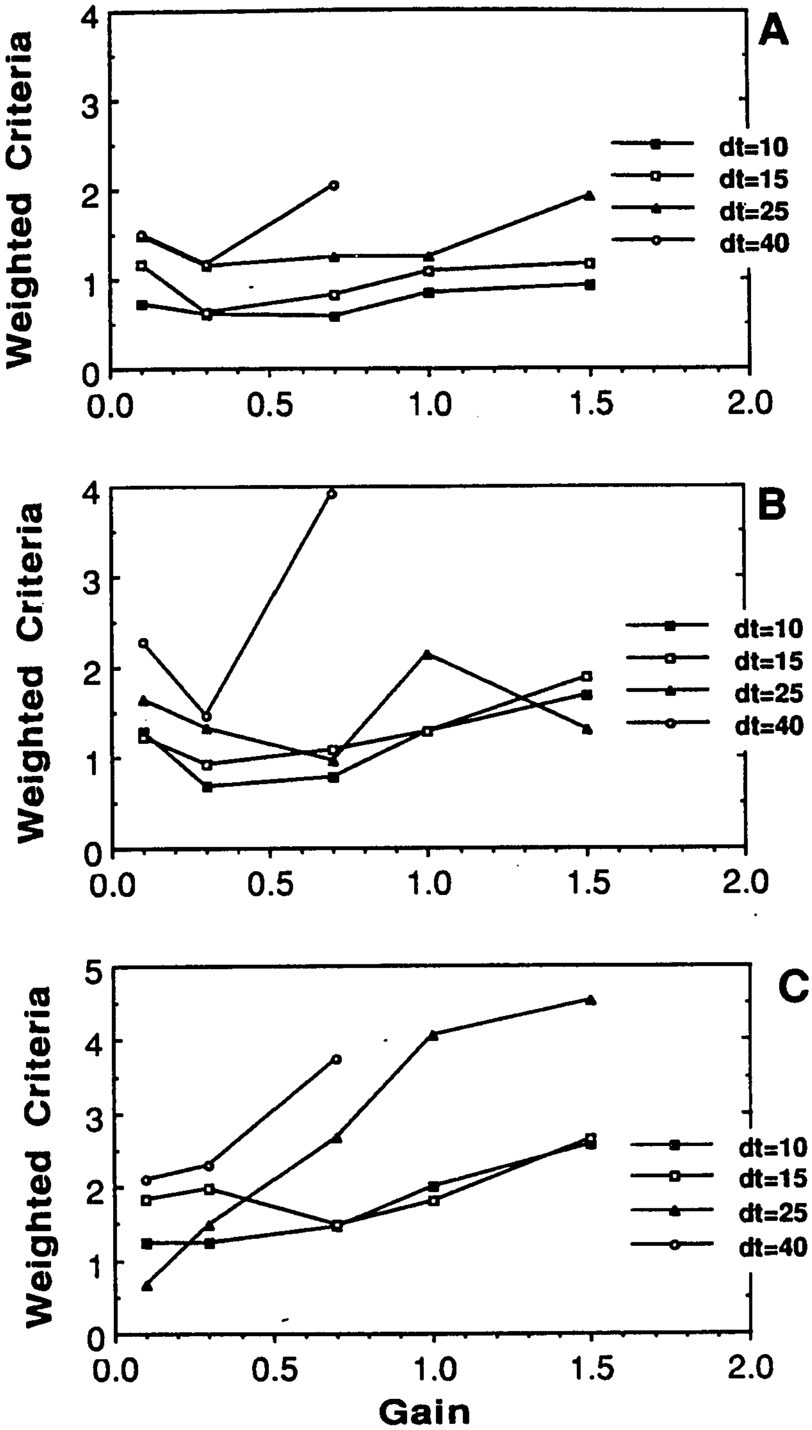


Figure E-6 Weighted Criteria. The percent overshoot, settling time and RMS temperature were summed with weighting factors to form the weighted criteria. A) W = 0. B) W = 2.5 kg/m^3 sec. C) W = 5 kg/m^3 sec.

APPENDIX F

DEVELOPMENT OF A DIGITAL FILTER

The purpose of the filter used in this work was to remove periodic fluctuations in the temperature data due to asynchronous power scanning and thermocouple reading (Moros, Roemer and Hynynen 1988, and Linn, Roemer and Hynynen 1989). This was accomplished by implementing a low pass filter with a cut-off frequency (the frequency at which the attenuation = 0.707) below that of the periodic fluctuations. With this procedure, unwanted components of the signal were attenuated, leaving a relatively smooth temperature profile which could easily be analyzed. Although the filter shifts the phase, the data is all treated the same, and thus the phase shift is irrelevant.

As a means of attaining a reasonably high quality filter (sharp transition from pass band to stop band), an IIR (infinite impulse response) filter was selected (Williams 1986, p. 127). This type of filter uses previous inputs as well as previous outputs to form the next output and is of the form:

$$y_k = \sum_{l=0}^{M} c_l x_{k-l} + \sum_{l=1}^{N} d_l y_{k-l} \qquad \text{(F-1)}$$

where:

x_k = k^{th} component of the input data
y_k = k^{th} component of the filtered data
c_l = nonrecursive filter coefficients
d_l = recursive filter coefficients

M = number of nonrecursive coefficients - 1

N = number of recursive coefficients

As a check on the computation of the coefficients for a unity gain low pass filter, the sum of the filter coefficients should be 1. This is expressed as follows:

$$\sum_{i=0}^{M} c_i + \sum_{i=1}^{N} d_i = 1 \qquad \text{(F-2)}$$

The filter coefficients were derived by converting an analog filter to a digital filter using the Tustin derivative approximation (Palm III 1983, p. 261), and frequency prewarping (Parks & Burrus 1987, p. 209-213). A 3rd order low pass Butterworth filter was used since it is a reasonably simple and common choice for digital filters (Peled and Liu 1976, p. 47).

The original analog filter (normalized to 1 rad/sec) was

$$T(s) = \frac{1}{s^3 + 2s^2 + 2s + 1}$$

The Tustin derivative approximation with frequency prewarping yielded the following s- to z-plane substitution:

$$s = \frac{u_c}{\tan(\omega_c T/2)} \frac{(z-1)}{(z+1)}$$

where:

u_c = analog cut-off frequency [rps]

ω_c = desired digital cut-off frequency [rps]

T = digital sampling period [sec]

Performing the substitution and solving for the coefficients of equation F-1 yielded the following result:

$$c_0 = 1/d_0$$
$$c_1 = 3/d_0$$
$$c_2 = 3/d_0$$
$$c_3 = 1/d_0$$
$$d_1 = (3c^3 + 2c^2 - 2c - 3)/d_0$$
$$d_2 = (-3c^3 + 2c^2 + 2c - 3)/d_0$$
$$d_3 = (c^3 - 2c^2 + 2c - 1)/d_0$$

where:

$$d_0 = c^3 + 2c^2 + 2c + 1$$
$$c = u_c/\tan(\omega_c T/2)$$

These equations were used to calculate the coefficients of a filter with any given cut-off, ω_c, and sampling period, T. For example, a filter with a cut-off at 0.020 Hz operating on data sampled every 10 seconds had the following coefficients:

$$c_0 = c_3 = 0.0985$$
$$c_1 = c_2 = 0.2956$$
$$d_1 = 0.5772$$
$$d_2 = -0.4218$$
$$d_3 = 0.0563$$

Note that the sum of the coefficients adds up to 1, thereby satisfying equation F-2.

REFERENCES

Benkesser, P.J., L.A. Frizzell, K.R. Holmes, and S.A. Goss, "A Perfused Tissue Phantom for Ultrasound Hyperthermia", Submitted for Publication in IEEE Transactions On Biomedical Engineering, 1989.

CRC Handbook of Chemistry and Physics, 66th edition, R.C. Weast, ed., CRC Press, 1985.

Dally, J.W., W.F. Riley and K.G. McConnell, Instrumentation for Engineering Measurements, John Wiley and Sons, Inc., 1984.

Diederich, C., The Implementation and Evaluation of two Thermal Techniques for Measuring Local Tissue Perfusion, M.S. Thesis, Department of Electrical and Computer Engineering, University of Arizona, 1986.

Holmes, K.R., W. Ryan, P. Weinstein, and M. Chen, "A Fixation Technique for Organs to be Used as Perfused Tissue Phantoms in Bioheat Transfer Studies", ASME WAM, 1984.

Hynynen, K., R.B. Roemer, D. Anhalt, C. Johnson, Z. Xu, W. Swindell, and T.C. Cetas, "A Scanned Focussed Multiple Transducer Ultrasonic System for Localized Hyperthermia Treatments", Int. J. of Hyperthermia, vol. 3, p. 21, 1987.

Hynynen, K., D. DeYoung, M. Kundrat, and E. Moros, "The effect of blood perfusion rate on the temperature distributions induced by multiple, scanned and focused ultrasonic beams in dogs' kidneys in vivo", Int. J. of Hyperthermia, vol. 5, pp. 485-497, 1989.

Kress, R.L., Adaptive Model-Following Control for Hyperthermia Treatment Systems, Ph.D. Thesis, Department of Aerospace & Mechanical Engineering, University of Arizona, 1988.

Lin, W., R.B. Roemer and K. Hynynen, "Theoretical and Experimental Evaluation of a Temperature Controller for Scanned Ultrasound Hyperthermia", Submitted for Publication in Medical Physics, 1989.

Morros, E.G., R.B. Roemer, and K. Hynynen, "Simulations of Scanned Focussed Ultrasound Hyperthermia: The Effects of Scanning Speed and Pattern on the Temperature Fluctuations at the Focal Depth", IEEE Transactions on Ultasonic, Ferroelectrics, and Frequency Control, vol. 35, p. 552, 1988.

Palm III, Modeling, Analysis, and Control of Dynamic Systems, John Wiley and Sons, Inc., 1983.

REFERENCES--Continued

Parks, T.W., and C.S. Burrus, Digital Filter Design, John Wiley & Sons, Inc, 1987.

Pelled, A., and B. Liu, Digital Signal Processing, John Wiley & Sons, 1976.

Roemer, R.B., J.R. Oleson, and T.C. Cetas, "Oscillatory Temperature Response to Constant Power", American Journal of Physiology, vol. 249, 1985.

Song, C.W., A. Lokshina, J.G. Rhee, M. Patten, and S.H. Levitt, "Implication of Blood Flow in Hyperthermic Treatment of Tumors", IEEE Transactions On Biomedical Engineering, vol. 31, pp. 9-16, 1984.

Valvano, J.W., and P.A. Pat, "Self-heated Thermistors to Measure Perfusion", Proceedings of the 8th Annual Conference of the IEEE/Engineering in Medicine and Biology Society, 1986.

Williams, C.S., Designing Digital Filters, Prentice-Hall, Inc., 1986.