

---


Retrospective Theses and Dissertations

---

1988

## Direct digital control of a liquid level control unit

Barbara Viehman Lytle  
*University of Central Florida*

 Part of the [Computer Engineering Commons](#)  
Find similar works at: <https://stars.library.ucf.edu/rtd>  
University of Central Florida Libraries <http://library.ucf.edu>

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact [STARS@ucf.edu](mailto:STARS@ucf.edu).

---

### STARS Citation

Lytle, Barbara Viehman, "Direct digital control of a liquid level control unit" (1988). *Retrospective Theses and Dissertations*. 4309.  
<https://stars.library.ucf.edu/rtd/4309>

DIRECT DIGITAL CONTROL OF A LIQUID LEVEL  
CONTROL UNIT

BY

BARBARA VIEHMAN LYTLE  
B.A., University of Central Florida, 1972

THESIS

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science  
in the Graduate Studies Program  
of the College of Engineering  
University of Central Florida  
Orlando, Florida

Summer Term  
1988

## ABSTRACT

A system was developed to allow undergraduate engineering students the capability of simulating various types of control modes in a laboratory environment. The use of digital control and common industrial components such as transducers, recorders, and valves was implemented.

Modification to an existing bench scale process control system is outlined and demonstrated. A computer program was written which interfaces with the hardware and data acquisition software allowing the user many options to study control processes under a variety of conditions. A wide range of parameters can be selected. System parameters can be saved on disk or printed directly. Data are also sent to a graphics program.

Three control modes, Proportional, Proportional - Integral and Proportional - Integral - Derivative, were tested. Proportional mode had a damped voltage, but the level was offset from the reference level. Adding integral control eliminated the offset, but the voltage was less damped. The inclusion of derivative decreased the amount of overshoot and undershoot.

## ACKNOWLEDGEMENTS

I would like to thank my thesis advisor, Dr. Harold Klee, for his excellent advice and support during my thesis project; and my husband Steve, for his help in organizing and helping me to complete my masters.

## TABLE OF CONTENTS

LIST OF FIGURES.....	vi
LIST OF TABLES.....	viii
NOMENCLATURE.....	ix
INTRODUCTION.....	1
CHAPTER	
I.    ORIGINAL SYSTEM DESCRIPTION.....	2
Description of Original Process....	2
Panel Components.....	6
II.   MODIFIED SYSTEM DESCRIPTION.....	10
Description of Modified Process....	10
Adaptations of System Configuration	17
III.  DIGITAL CONTROL PROGRAM.....	22
Introduction.....	22
BASICA to LabSoft Interface.....	22
Program Structure.....	23
Program Explanation.....	24
Main Program.....	24
Execution Subroutine.....	25
Trap Subroutine.....	34
Setpoint Subroutine.....	39
Conversion Subroutine.....	40
Control Algorithm Subroutine....	44
IV.  RESULTS.....	55
1.  Comparison of Experimental Results to the Three Control Algorithms....	55
2.  Comparison of the Experimental Results to the Values of the Control Parameters.....	57
V.  CONCLUSIONS.....	78

APPENDICES

A. SYSTEM CONFIGURATION.....	79
B. PROGRAM LISTING.....	87
LIST OF REFERENCES.....	124

## LIST OF FIGURES

1.	Technovate Model 9030 Front View.....	3
2.	Liquid Level Tank System.....	4
3.	Control System Block Diagram.....	11
4.	Inside the Digital Computer.....	15
5.	Relationship Between Discrete and Continuous Signals.....	16
6.	Main Program Flowchart.....	26
7.	Main Menu.....	27
8.	Execution Subroutine Flowchart.....	28
9.	Current Value Screen.....	31
10.	Trap Subroutine Flowchart.....	37
11.	Setpoint Subroutine Flowchart.....	41
12.	Setpoint Menu.....	42
13.	Conversion Subroutine Flowchart.....	45
14a.	Control Algorithm Subroutine Flowchart.....	49
14b.	Proportional Control Subunit of Control Algorithm Flow Chart.....	50
14c.	Proportional Integral Control Subunit of Control Algorithm Flowchart.....	51
14d.	Proportional Integral Derivative Control Subunit of Control Algorithm Flowchart.....	52
15.	Main Control Algorithm Menu.....	53
16.	Proportional Parameter Menu.....	53
17.	Proportional Integral Parameter Menu.....	54

18.	Proportional Integral Derivative Parameter Menu.....	54
19.	Level Comparisons of C.A. Attaining 7.5 in...	63
20.	C.A. Comparison of Voltages to Attain 7.5 in.	64
21.	Level Attained by C.A. at Setpoint = 3 in....	65
22.	Comparison of Voltages to Attain 3.0 in.....	66
23.	Comparison of C.A. to Attain 10 in.....	67
24.	Voltages to Attain Setpoint = 10.0.....	68
25.	System at Maximum Level.....	69
26.	Level With Diverse $K_P$ Values.....	70
27.	Voltages With Diverse $K_P$ Values.....	71
28.	P-I Levels With Diverse $K_P$ Values.....	72
29.	Voltages With Various $K_P$ Values $K_I = 1$ .....	73
30.	Level Fluctuations With Various $K_I$ Values....	74
31.	Voltages With Various $K_I$ Values.....	75
32.	Processor Levels With Various $K_D$ Values.....	76
33.	Voltages With Various $K_D$ Values.....	77
34.	Block Diagram of Instruments in the System...	83
35.	ISAAC 91-I Connections.....	84



## LIST OF TABLES

1. Hardware and Software Requirements.....	21
2. Height in Processor vs. Voltage of Pressure Transducer.....	38
3. Control Algorithm Comparisons.....	61
4. Actual Air Pressure vs. Simulated Air Pressure.....	62
5. Switch Settings for ISAAC Hardware.....	82
6. Instrument Connections.....	85
7. Hewlett-Packard Triple Output Power Supply Connections and Settings.....	86

## NOMENCLATURE

### SYMBOLS

- $H_s$  - setpoint or reference point
- $V_T^*(k)$  - discrete voltage from pressure transducer
- $V_T$  - continuous voltage from pressure transducer
- $P$  - pressure in Processor or tank
- $F_i$  - input flow
- $O$  - load variable
- $F_o$  - output flow
- $P_o$  - water supply pressure
- $U$  - control voltage
- $P_v$  - control pressure
- $E(k)$  - error
- $V_T(k)$  - voltage sequence from pressure transducer
- $U_o(k)$  - output voltage sequence
- $U^*(k)$  - output voltage sequence in range of 0-4095
- $U(k)$  - output voltage sequence in range of 0-10 volts

## INTRODUCTION

The undergraduate Computer Engineering curriculum at the University of Central Florida will offer a course in Digital Control Systems. Students will obtain laboratory experience with the principles of digital controls using the system to be described in this work. A description of the process without computer control is given in Chapter I. Modifications to the system necessary to implement computer control are explained in Chapter II.

Software developed to execute system control is discussed in Chapter III. A variety of menus to select and set different parameters are included in the program. Essential system variables are graphed on the screen and on a flat-bed X-Y recorder. System parameters are saved on floppy disk and can be directed to a printer or used in conjunction with a graphics program.

Different types of control algorithms can be selected. Proportional (P) Control is effective but produces a steady-state offset. Proportional Integral (P-I) Control achieves setpoint and maintains it. Proportional Integral Derivative (P-I-D) Control achieves setpoint quicker and will anticipate changes in the system.

## CHAPTER I - ORIGINAL SYSTEM DESCRIPTION

### Description of Original Process

The TECHNOVATE Model 9030 trainer (shown in Figure 1) allows the operator to experiment with a continuous system. A schematic representation of the system including the significant components is shown in Figure 2 (Klee 1988). Water level,  $H$ , in the sealed tank is controlled by regulating the input flow,  $F_1$ , through a pneumatically operated control valve. The tank is initially uncharged (zero gauge pressure and tank pressure,  $P$ , is uniquely related to the liquid level). Due to this relationship between  $P$  and  $H$ , the tank pressure serves as an indicator of the actual liquid level. This is fed back to a pneumatic controller which also accepts an air signal setpoint pressure,  $P_s$ . When the fluid level reaches the desired setpoint level,  $H_s$ , the tank pressure  $P$  and setpoint pressure  $P_s$  equalize and the error, measured in either level  $e_H = H_s - H$  or pressure  $e_P = P_s - P$ , is zero.

Load variables or disturbances tend to disrupt the liquid level. The load variables are the position,  $\theta$ , measured in percent opening of a manually adjustable valve in the discharge line and the supply pressure of the fluid,

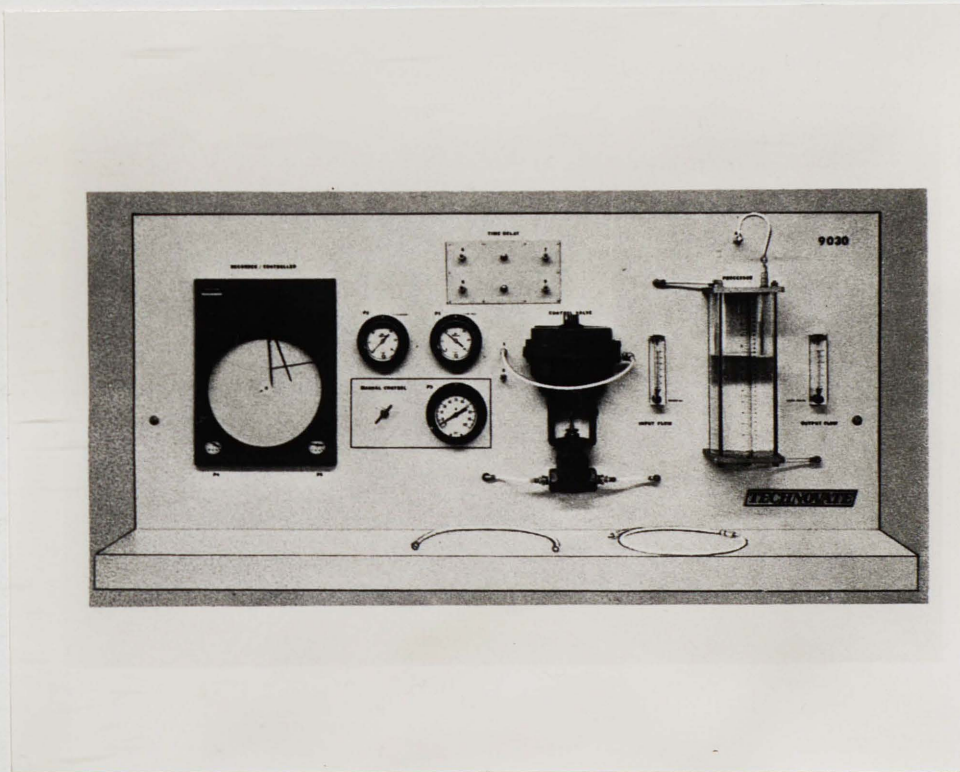


Figure 1. TECHNIVATE Model 9030 Front View.

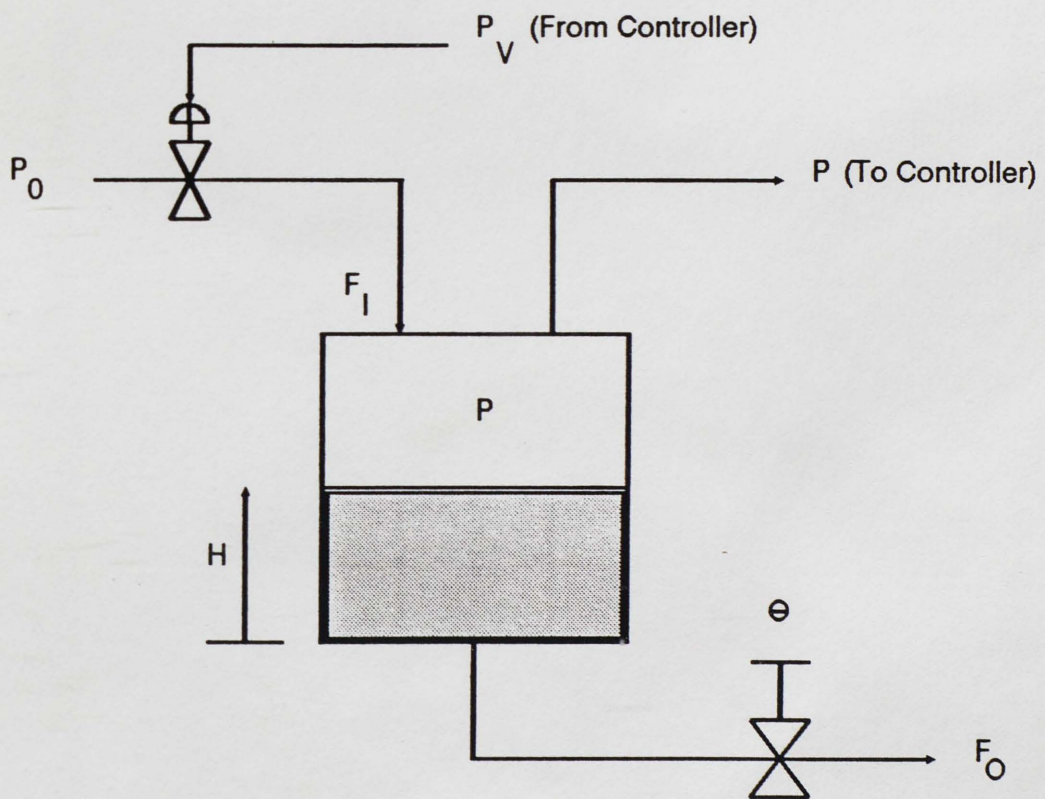


Figure 2. Liquid Level Tank System.

$P_o$ . The load valve position determines the effective fluid resistance in the line, and in combination with the liquid level influences the outflow,  $F_o$ . The inflow,  $F_i$ , is a function of the valve pressure  $P_v$  coming from the controller and the pressure drop across the control valve,  $P_o - P$ .

The controller serves two functions. First, it compares the pressures  $P_s$  and  $P$ . It also adjusts the output pressure  $P_v$  according to a control algorithm when the two input pressures are different. If the controller is effectively tuned (control algorithm parameters are chosen properly), the flow in  $F_i$  will change in the appropriate direction necessary to reduce the deviation,  $H_s - H$  (Klee 1985).

The TECHNOVATE Model 9030 can be operated manually or automatically. Automated control systems generally operate using a combination of mechanical, electrical, hydraulic, and fluidic devices. While the exact mechanism of these individual devices is different, the principles of automation and control systems are similar (Hale 1988). The Model 9030 utilizes a pneumatic circuit which has a slow response time compared with electronics. However, it is useful in training applications so that changes can be made and traced within the circuit. Pneumatic circuitry used in the Model 9030 contains common industrial hardware which is reliable and safe. The Model 9030 allows the operator to

simulate control, process operation, measurement, and adjustment.

Control activity includes manual regulation of flow, water level, and pressure. Automatic control of pressure and water level is possible under a variety of control modes. Automatic control modes include: on/off, proportional, proportional plus reset, proportional plus rate, and proportional plus reset plus rate.

#### Panel Components

Panel components of the Model 9030 are illustrated in Figure 1, which shows the front of this unit. The different panel components are described below.

##### 1. Controller

The controller is pneumatic and has provisions for proportional, reset and rate actions. It functions in the automatic control circuit to compare the value of the variable (air pressure in the Processor) against the set point, record it and pass a corrective signal to the control valve.

##### 2. Control Valve

The control valve is a pneumatic, diaphragm actuated, air-to-open valve. A pressure of three pounds per square inch (psi) is required to initiate valve opening.



Fifteen psi completely opens the valve. The valve functions in the process system as the final control actuator in either manual or automatic control modes.

### 3. Processor

The Processor is the vessel where the physical changes (water level or air pressure) in supply and demand occur. Physical changes affect the process variable. The vessel acts as a transducer to transmit air pressure changes in the process variable to the controller by pneumatic means. The Processor has a visual scale to measure the water level in either inches or centimeters. A valve in the outlet rotameter (see input flow) is used to "load" the system.

### 4. Input Flow

Input flow,  $F_1$ , is regulated by a ball-float type rotameter calibrated in percent of flow and is used to indicate the flow of water passing through the control valve and into the Processor. The valve on the rotameter should normally be fully open. The input flow valve may be used to manually override the control valve to reduce the flow into the Processor.

### 5. Output Flow

Specifications for the output flow rotameter are the same as the input flow rotameter. The function of the output flow rotameter is to implement a load on the

Processor and to indicate the magnitude of the load by exhibiting the flow out of the Processor to the drain.

#### 6. Manual Control

Manual control consists of a standard air pressure regulator and gauge (P3). Its function is to allow the following: manual operation of the control valve, transfer switching operations, and manual setting of water level in the Processor.

#### 7. Pressure Gauges

Seven standard Bourdon-tubes are used as pressure gauges. Three gauges (P<sub>1</sub>, P<sub>2</sub> and P<sub>3</sub>) are for instructional use and two (P<sub>4</sub> and P<sub>5</sub>) are integral parts of the controller. Two more, P<sub>6</sub> and P<sub>7</sub> for operational convenience, are located behind the panel. (See Figure 1)

P<sub>1</sub> and P<sub>4</sub> - indicate the pressure of controlled air coming from the controller to the control valve.

P<sub>3</sub> - indicates the pressure of regulated air going to other components and coming from manual control.

The manual regulator is completely open when P<sub>3</sub> reads the same as P<sub>5</sub> and P<sub>6</sub>.

P<sub>2</sub> - indicates the pressure of the feedback signal being transmitted to the controller from the Processor. A safety relief valve in this line protects the system against unintentional high

pressures. Pressure can also be relieved by disconnecting the hose from QC 4 at the top of the Processor.

$P_5$  and  $P_6$  - indicate the pressure of the filtered air to the controller and the manual control panel from an external supply.

$P_7$  - indicates the pressure of the water supply (25 psi is recommended)

#### 8. Quick Connects

Seven female quick connects (QC) are located on the panel, one on the control valve, QC2, and one on the Processor, QC3. All operations and circuits use these connections. Quick connects allow fast circuit changes and are joined by lengths of flexible pneumatic pressure hose.

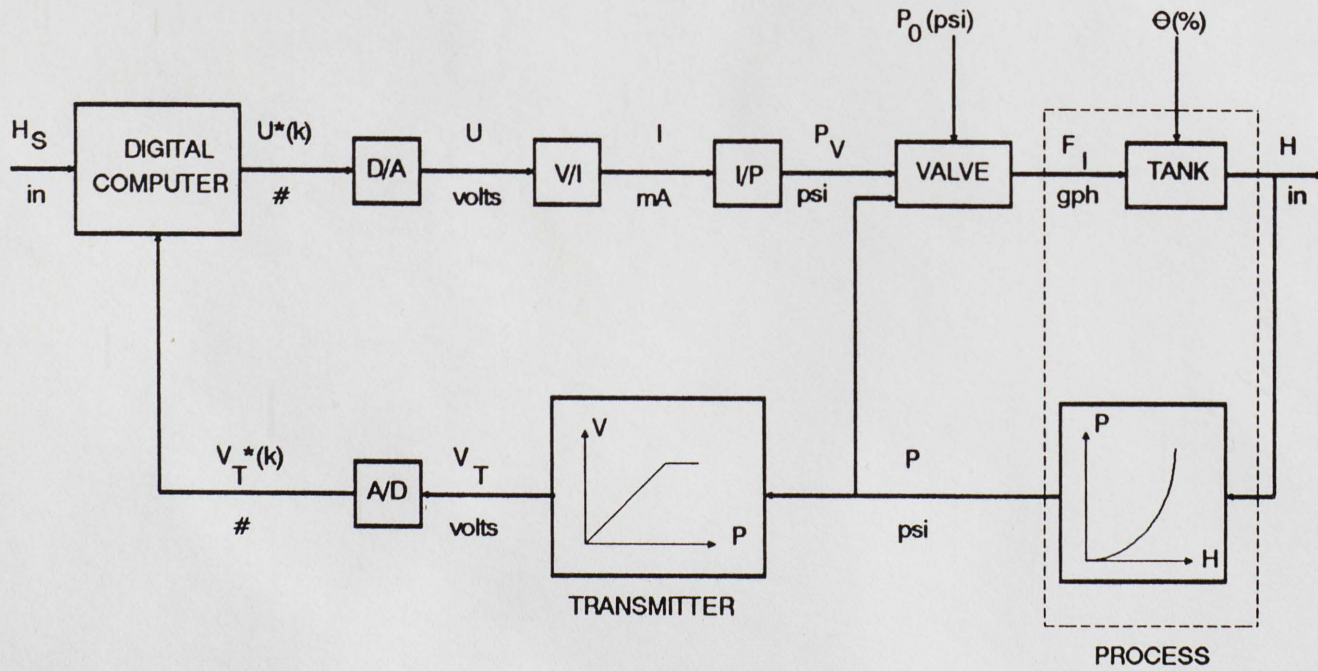
## CHAPTER II - MODIFIED SYSTEM DESCRIPTION

### Description of Modified Process

The modified system consists of three main components. Other instruments are involved to integrate the subunits. TECHNOVATE PROCESS CONTROL TECHNOLOGY LEARNING SYSTEM, Model 9030 is a system designed to teach effective analog control of a continuous process by manual or automatic means. ISAAC 91-I by Cyborg allows the computer, IBM PC-XT, to interface with the system for data acquisition. Three units are integrated for the purpose of controlling the Processor. Water level is monitored and the voltage necessary to maintain or achieve the setpoint is calculated. The control valve is adjusted to achieve steady state (Wong 1987).

Using the following modifications, the system can operate under direct digital control. A block diagram of the overall control system with the process parameters is shown in Figure 3. Reference level,  $H_s$ , (setpoint) is input and stored in memory. An additional input of a discrete sequence of numbers,  $V_T^*(k)$ , results from sampling the continuous voltage,  $V_T$ , coming from the pressure to voltage transducer (P/V) which converts the air pressure in the

Figure 3. Control System Block Diagram.



tank,  $P$ , to a voltage for the A/D conversion. The voltage corresponds to a unique relationship between the air pressure and the liquid level,  $H$ . As the liquid level goes higher, the air is compressed and the air pressure also increases. A relationship between the air pressure and the liquid level exists which serves as the level transmitter in the system. During calibration of the system, the liquid level, air pressure and voltage were noted at each level to document the correspondence.

The Processor has a regulated input flow,  $F_i$  and a load valve in the output flow for introducing disturbances (Claudio 1986). While the system is running, the output flow is not under direct digital control. The load variable is  $O$ , the valve position, which in addition with the pressure head at the bottom of the tank controls the output flow,  $F_o$ .

The pneumatic control valve is the final control element. Flow through the valve is determined by the control pressure,  $P_v$ , which opens the valve, and the pressures on either side. Pressures on either side of the control valve are the water supply pressure,  $P_o$ , and the air pressure in the tank  $P$ . Water supply pressure is a secondary load variable in the system. Signal conditioning is used to convert the control voltage  $U$ , coming from the D/A converter, to the control pressure  $P_v$ . The D/A signal is passed

to the voltage to current (V/I) converter. A current to air (I/P) converter receives the milliampere signal and transforms it to a pneumatic signal. This pneumatic signal is then sent to the pneumatic control valve for regulation of the liquid level in the Processor.

Events which occur inside the IBM PC merit a closer inspection. Figure 4 describes the conversions inside the IBM PC. Setpoint is entered directly from the keyboard and converted from inches to voltage via an array manipulation and interpolation. Error,  $E(k)$ , is calculated at discrete sampling intervals.  $E(k)$  is a measure of the difference in voltages between the setpoint and the liquid level.

The discrete sequence of numbers,  $V_T^*(k)$ , is converted to a voltage sequence,  $V_T(k)$ , within the computer for use in calculating the error sequence,  $E(k)$ . Figure 5 shows the relationship of changing a continuous signal to a discrete one. The discrete control algorithm generates the output voltage sequence,  $U_0(k)$ .  $U_0(k)$  is checked to ensure that it falls within the limits of the D/A converter which is 0 to 10 volts and becomes sequence,  $U(k)$ .  $U(k)$  is converted to a sequence of numbers in the range of 0 to 4095 and designated  $U^*(k)$ .  $U^*(k)$  is sent to the D/A converter which produces a piecewise continuous voltage sequence,  $U$ . Figure 5 shows the relationship between the voltage sequence,  $U(k)$ , inside the computer and the corresponding signal,  $U$ , which

controls the pneumatic control valve position. A three channel recorder is used to document the reference input, liquid level and the control output voltage signals (Clark 1977).



Figure 4. Inside the Digital Computer.

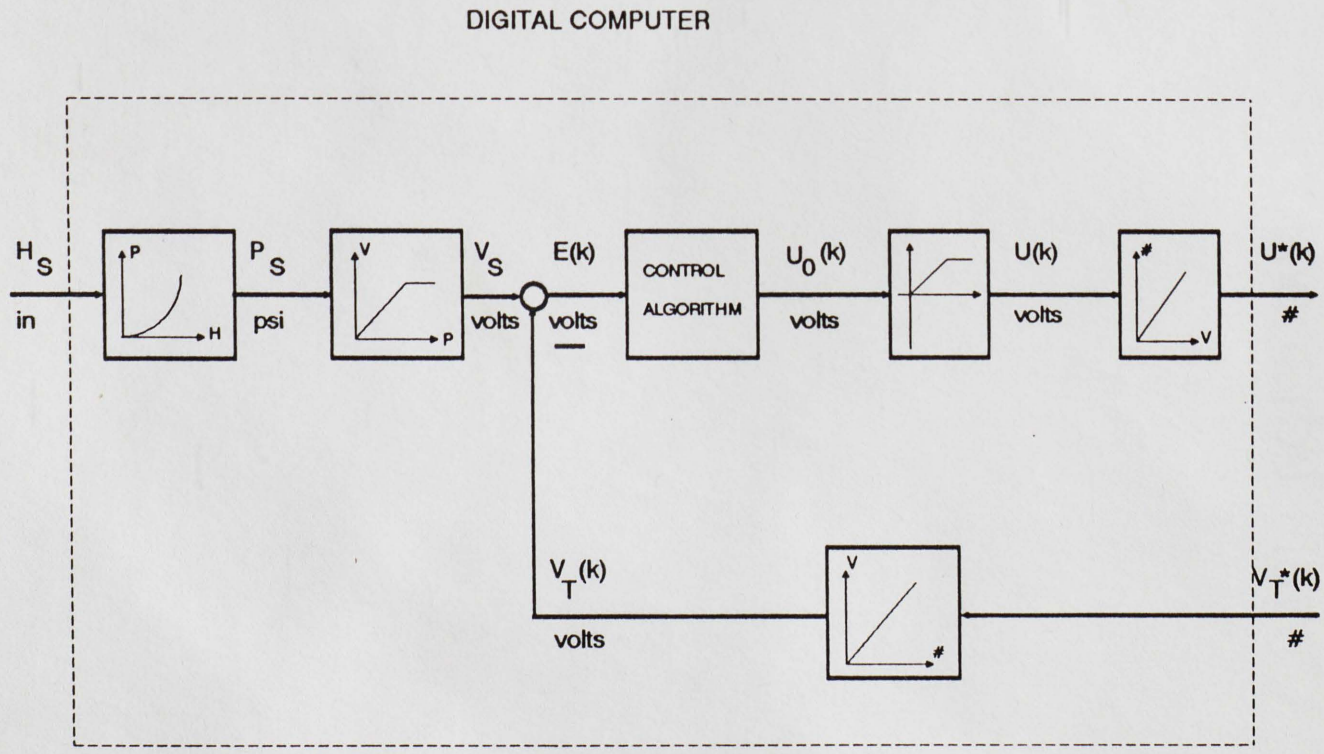
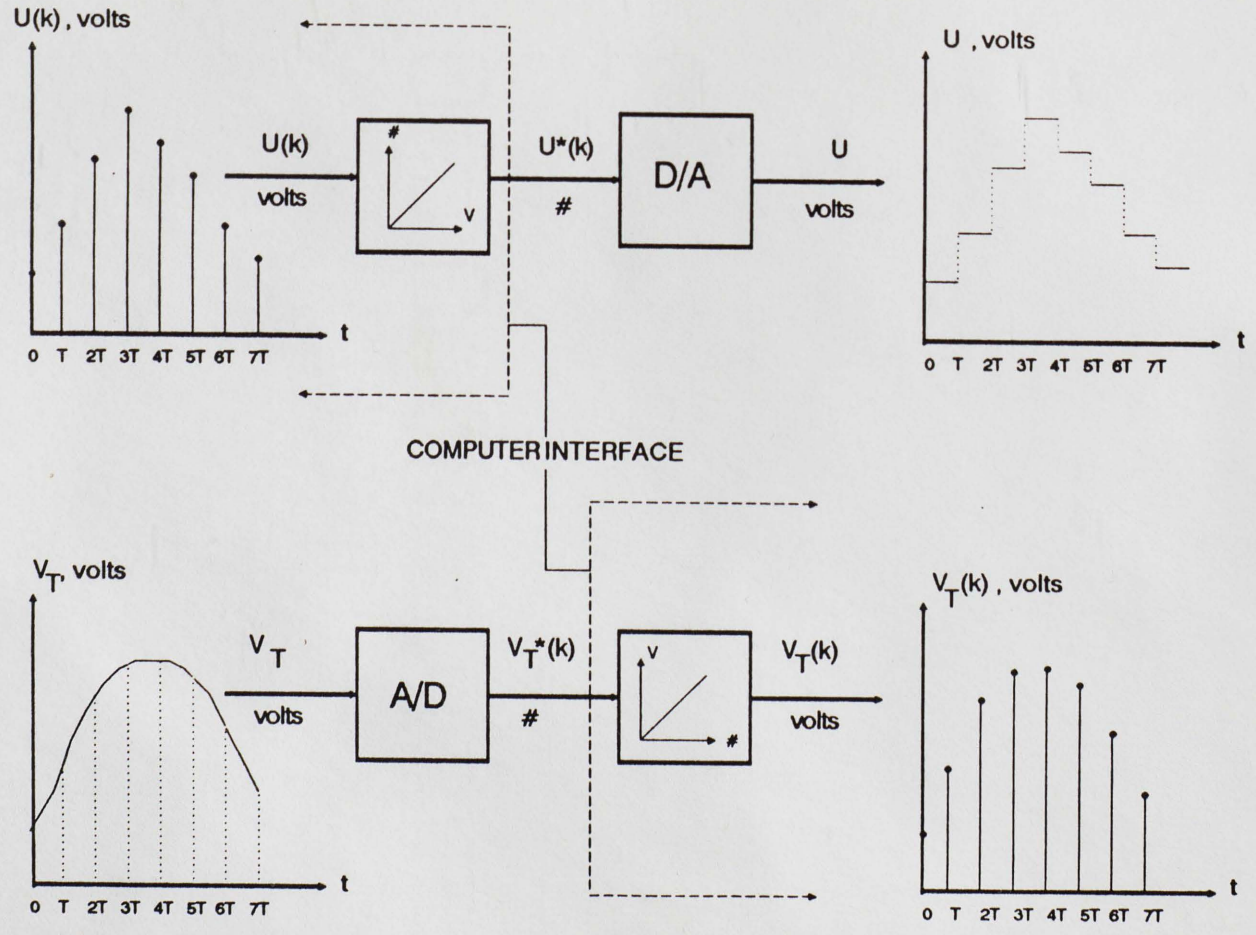


Figure 5. Relationship Between Discrete and Continuous Signals.



decreased the air volume and will now shut down the system before the Processor fills completely. Pressure gauges  $P_6$  and  $P_7$  are now used. All other pressure gauges have been bypassed.

Due to the above modifications, the controller does not receive the air pressure signal and is not used. Since the controller is not used, QC's for manual and automatic control are not needed. The air supply was adapted to respond to computer control by way of the ISAAC 91-I. The program calculates the voltage needed to open/close the control valve to achieve or maintain setpoint. A signal is transmitted via the ISAAC 91-I interface to the voltage to current converter. The current to pneumatic signal converter then outputs the air supply to the control valve (QC 2). Figure 2 shows the adapted system.

A flat-bed recorder is used to plot the following variables: the actual level in the Processor; the setpoint level and the volts calculated to achieve or maintain setpoint, U(1).

#### 1. ISAAC 91-I by CYBORG

The ISAAC 91-I is a modular data acquisition and control system which serves to link the IBM PC and the other instruments together. ISAAC 91-I has built-in acquisition and control hardware, consisting of the following : 16 -channel, 12-bit analog-to-digital (A/D)

converter; 4-channel, 12-bit digital-to-analog (D/A) converter; four programmable Schmitt triggers; 16 binary outputs and 16 binary inputs; 16-bit counter; and binary and analog handshaking lines. ISAAC 91-I also has eight expansion slots for I-series expansion devices. The 91-I system's hardware consists of the 91-I chassis and on-board data acquisition hardware, the ISAAC/IBM PC interface card and cables. Conversion via the ISAAC interface card translates the internal signals of the IBM PC into signals that the ISAAC 91-I system can process.

ISAAC 91-I's power supply provides  $\pm 12$  volts from which the ISAAC 91-I reference voltage subsystem derives a reference voltage of  $\pm 5$  V. This reference voltage gives analog I/O devices installed in the ISAAC 91-I a voltage range (exclusive of input gain or input dividing) of  $\pm 5$ V. Voltage range can be increased to  $\pm 10$  volts depending on the switch settings inside the ISAAC 91-I chassis. Settings for the switches in the ISAAC chassis and the interface are located in Appendix A and "User's Guide to the Direct Digital Controlled TECHNOVATE System."

## 2. HEISE Model 620 Series Pressure Transducer

A HEISE transducer converts air pressure to a voltage of 0 to 10 volts. A 40 volt DC power supply is required.

Instrument accuracy is  $\pm .15\%$  over the operating range of 0 to 60 psi and operates over a range of 0 to 150 degrees Fahrenheit.

3. LOVE Model 106 Process Signal Converter-Isolator

The LOVE Model 106 (V/I) is a general purpose device which can convert one process signal to any other. Conversion in this example is voltage to milliamperes. It is also useful as a signal isolator (input from output). Both inputs and outputs are selectable by switches located on the front of the unit.

4. FOXBORO Model E69F Current-To Pneumatic Signal Converter

This device changes a direct current input signal to a pneumatic output proportional to the signal. Output signals can be used to operate a variety of devices such as dampers and solenoids. The range of input current is 4 to 20 milliamperes while the output is 3 to 15 psi. Normal environmental conditions are  $-30$  to  $+60$  degrees Celsius.

5. Additional equipment consists of a standard power supply and a flat-bed recorder. The flat-bed recorder is used to plot the following variables: the actual level in the Processor,  $V(k)$ , the setpoint level,  $H_s$ , and the volts calculated to achieve/maintain setpoint,  $U_0(1)$  (Clark 1977).

TABLE 1

## HARDWARE AND SOFTWARE REQUIREMENTS

---

HARDWARE	SOFTWARE
CYBORG ISAAC 91-I with External Power Supply	CYBORG LabSoft ISAAC Software
IBM XT with High Resolution Monitor 256 K Memory	IBM DOS 3.0 IBM BASICA

---

## CHAPTER III - DIGITAL CONTROL PROGRAM

### Introduction

BASICA, the IBM version of BASIC for personal computers, was chosen as the language to develop the control program. LabSoft is the language that provides specific instructions to interface BASICA with the ISAAC 91-I interface card and ISAAC 91-I chassis.

### BASICA to LabSoft Interface

LabSoft software is made of two machine code device drivers which communicate with the ISAAC 91-I. LabSoft serves as an interface between the ISAAC driver and BASICA. Device drivers are loaded into memory by booting the configuration file. Support of all ISAAC system capabilities, ISAAC expansion interface cards and the IBM Personal Computer Data Acquisition and Control Adapter are provided.

BASICA is provided with the correct addresses of the machine coded subroutines by a LabSoft header which is the initial part of the program. LabSoft header, using line numbers 1 to 99, is a fragment of a BASICA program, stored

in ASCII format. The header assigns values to all the LabSoft function name variables used with the CALL statement. LabSoft function names should be treated as reserved words and not used for any other purpose in the program. The program header queries the Disk Operating System (DOS) to obtain the starting address of the BASICA interface driver.

### Program Structure

A top down procedure was used to write the control program. Dummy line numbers for subroutines were used initially to test the main program decision tree. Subroutines were added as they were tested and debugged. Figure 6 shows the main decision tree of the program.

When numeric inputs are requested, values are first stored in a string variable and then converted to a numeric variable. This prevents letters from being entered accidentally and a prompt from BASICA appearing on the screen asking "Redo from start?". The prompt could cause confusion for the students using this program. Use of numeric values converted from string values solves this problem. For example, the variable can be 0.0 (as the setpoint) then a check for zero is made. A string value prompts error message so another value can be input.



Program Explanation

## Main Program

The main program (MAIN) calls other subroutines to perform the tasks throughout the program. A flowchart of MAIN is shown in Figure 6. MAIN begins by printing an introductory screen informing the user of the steps necessary to set up the system, where to place the formatted data disk and to ensure the printer is on line. MAIN continues by initializing the E and U arrays to zero. A menu (see Figure 7) is then displayed asking the user whether they wish to select one of four choices : A. Set/Change setpoint; B. Set/Change control algorithms; C. Begin execution of system; and D. Shut down the system. One of the four options is selected. MAIN then branches to the corresponding subroutine handler. After returning from the subroutine's control, MAIN branches back to the main menu screen waiting for another decision. If an invalid selection has been made an error message is printed on the screen and control branches back to the main menu for another input.

Only three of the selections from the main menu have subroutine handlers and will be discussed later. If "D. Shut Down System" is chosen, it is handled in the main program. Users are queried if exiting the program is still

desired. Response of "N" returns the user to the main menu for another decision. An answer of "Y" determines that the output voltage to the V/I converter is set to 0.0 activate closure of the control valve. No more water will enter the Processor and it will begin to drain by way of the output flow. The user is then asked if the data files should be sent to the printer. A reply of "Y" sends data in data files on disk to the printer and then the program ends. An answer of "N" ends the program. Control returns to BASICA and the screen shows the BASICA prompt, "OK".

An input of "C" from the main menu causes program branching to the execution block. If the setpoint and control algorithm have not been set, they would be zero. The voltage output to the V/I converter to be zero. Consequently, the air pressure from the I/P converter would not open the control valve and the Processor would not fill.

#### Execution Subroutine

The execution subroutine (EXECUTION) sets the values of the parameters needed in the graphics subroutine and the Trap Subroutine (TRAP). It also contains the call to TRAP. EXECUTION is called by MAIN and returns there when the user desires. Figure 8 illustrates the flowchart of this subroutine.

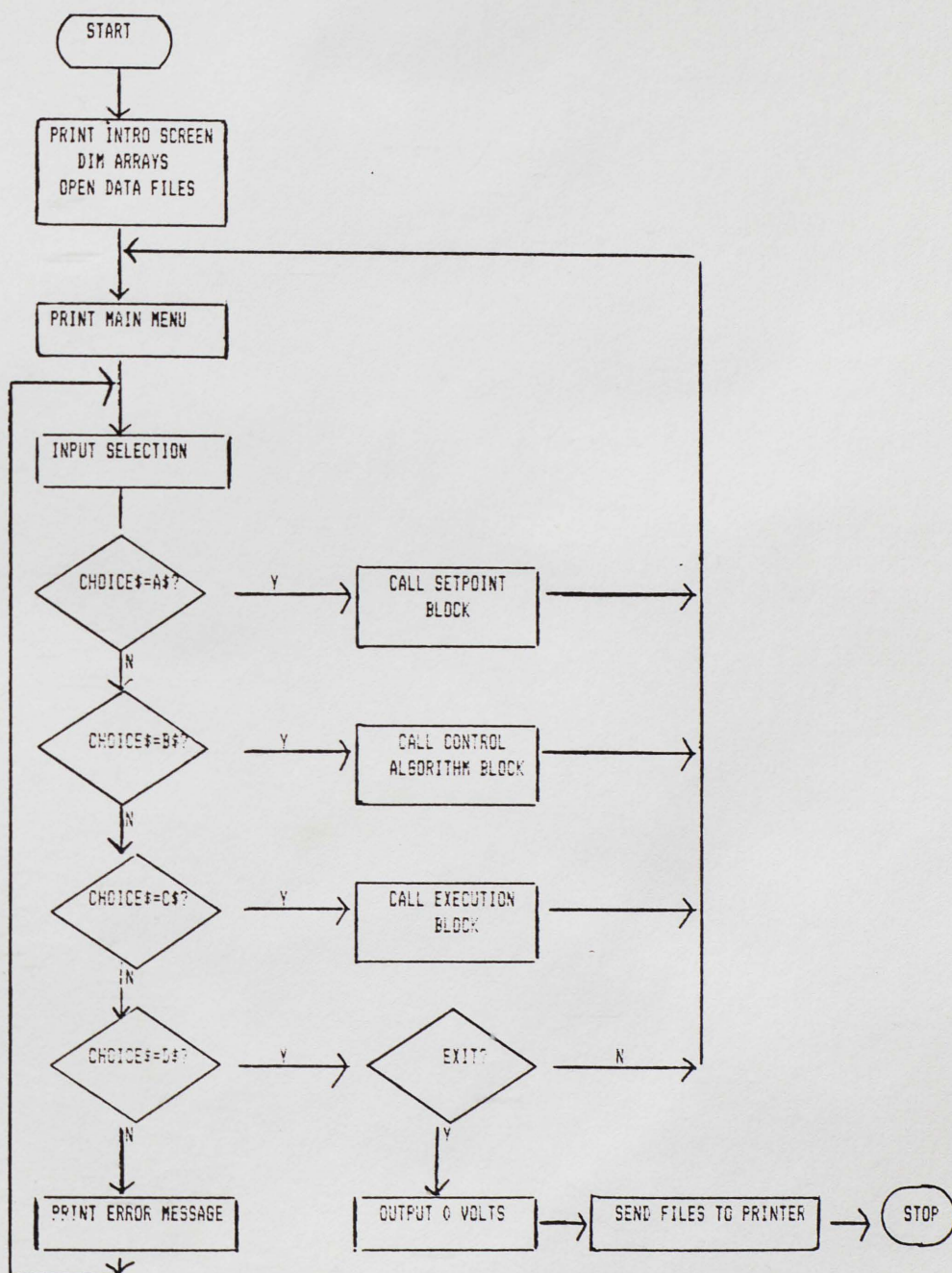


Figure 6. Main Program Flowchart.

```
*****  
*                                                                 *  
*   WELCOME TO THE TECHNOVATE SYSTEM   *  
*             MAIN MENU                 *  
*                                                                 *  
*           PLEASE SELECT ONE.         *  
*   A  SET/CHANGE SETPOINT VALUE.     *  
*   B. SET/CHANGE CONTROL ALGORITHMS. *  
*   C. BEGIN EXECUTION OF SYSTEM.     *  
*   D. SHUT DOWN SYSTEM.              *  
*                                                                 *  
*****
```

INPUT SELECTION.

Figure 7. Main Menu.

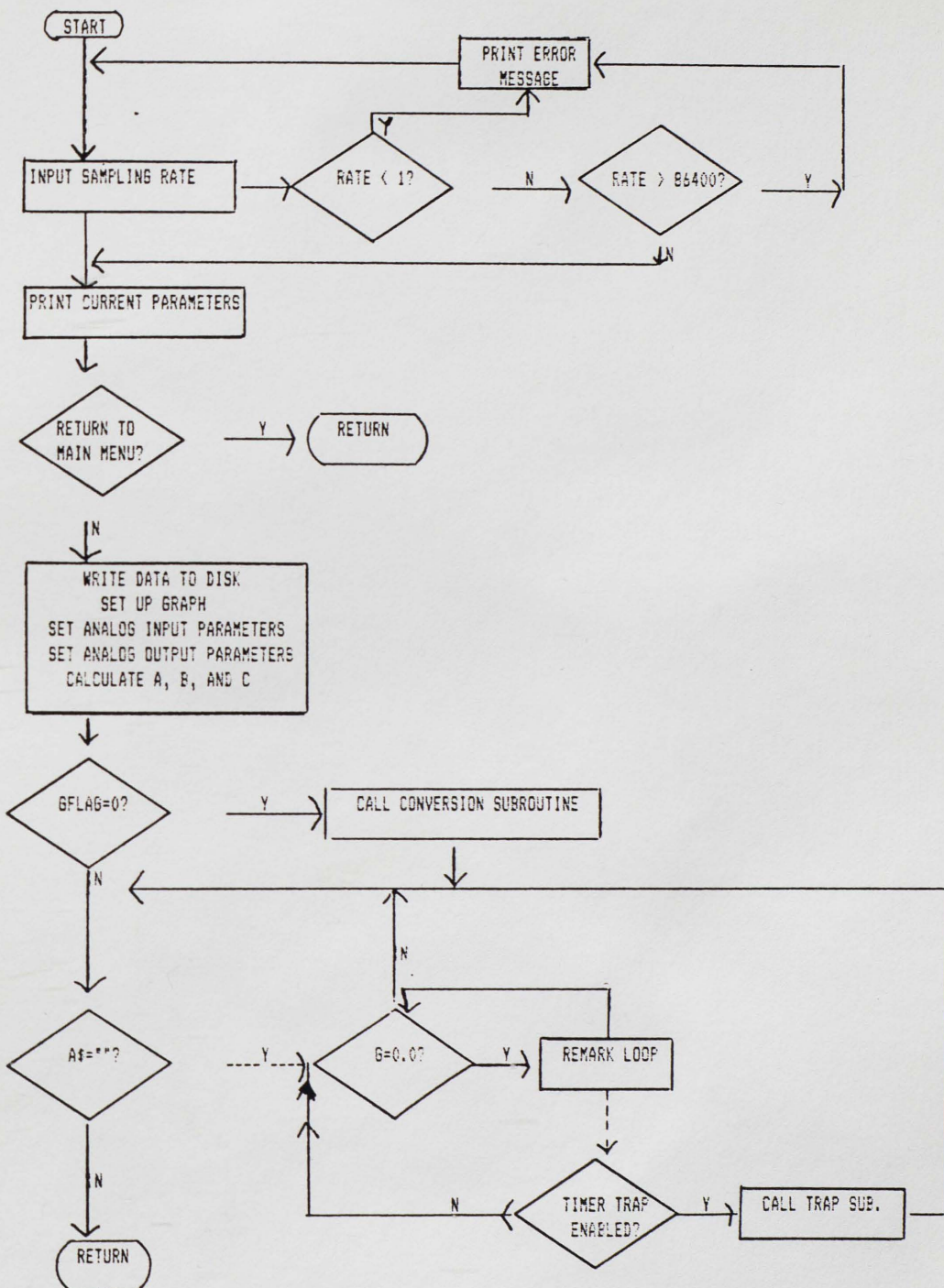


Figure 8. Execution Subroutine Flowchart.

The first event in EXECUTION is to set the sampling time. One of the limits of using BASICA is that the sampling time cannot be less than one second or greater than 86,4000 seconds. A checking procedure has to be implemented to ensure that a valid sampling time is input. The instruction "ON TIMER" is used later in the subroutine to implement the timing. This instruction invokes a trap to occur at the passing of each sampling time which branched to TRAP to handle the testing of the system.

After a valid sampling time is achieved, a screen appears to show the current values of all the parameters the user has input (See Figure 9). User can either return to the main menu or continue with the execution of the system. Values of the parameters (KP, KI, KD, TIME and CONTROL\$) are then written on the disk in drive B in the file "B:VARVAL".

The variable to show the number of samples taken for this set of parameters is set to 1. X is used to distinguish between the samples of different sets of parameters. X is increased after each sampling and is reset each time the user returns to the main menu from EXECUTION.

All the variable initialization is accomplished before the ON TIMER call to ensure that the trap subroutine does not exceed one second.

Graphic variables are set next. LabSoft provides a variety of graphing possibilities. A scrolling dot graph was considered for this program. It presents a taller graph (128 points high) and would illustrate the fluctuations around setpoint better. The drawback was the time required to reset all the points for the scrolling effect.

LabSoft also provides a continuous graph. This was impossible to use for graphing more than one variable. The continuous graph connects each consecutive value. In this case, the values of the three variables being graphed (setpoint, level and voltage to the pressure transducer) would have been connected to one another. Instead of the previous value of the setpoint being connected to the present value of the setpoint, it would connect the past values of the setpoint to the past values of the level and voltage. Therefore, the alternating dot graph was implemented.

The alternating dot graph does not scroll nor does it connect each successive dot. The graph is half as high (64 points high) and the fluctuations are not as evident as in the scrolling graph, which is a disadvantage. Alternating dot graphics consist of two windows. When both windows are filled, the top window is erased and the graphing continues. The bottom window is cleared after the top is filled again.

CURRENT VALUES ARE

KP = 1.0    KI = 1.0    KD = 1.0

SETPOINT = 7.5

SAMPLING TIME = 1

CONTROL ALGORITHM = P-I-D

PRESS 'Y' AND ENTER KEY TO RETURN TO MAIN MENU.

PRESS 'N' AND ENTER KEY TO BEGIN/CONTINUE EXECUTION OF  
THE SYSTEM.

ENTER CHOICE. (Y/N)

Figure 9. Current Value Screen.



When the graphing parameters have been set, LabSoft subroutine INITGRAF is called. This subroutine initializes a graphics screen. Number of plots to be displayed, colors to plot in, background color, and graph type to use are all processed. The graphics subroutine also allows a title, X-label, Y-label and comment to appear on the screen. High and low Y-axis are also set.

Variables for the simple analog outputs and multiple samples from one channel input are set. Simple analog outputs in the range of 0 to 4095 are: channel 0 - voltage to V/I converter; channel 1 - setpoint voltage to the flat-bed recorder; and channel 2 - P/V converter voltage is converted to units in inches and then to the 0 to 4095 range. Multiple samples from one channel use the following values: rate of 8,000 samples per second; number of samples taken is 10; all samples are stored in the array RAW% starting at RAW%(0) and the channel for input is analog to digital input channel 0.

Coefficients used in the equation for voltage to achieve or maintain setpoint are calculated. These coefficients are based on the discrete form of the P-I-D algorithm (Jacquot) as follows:

$$A = K_P + K_I * (TIME/2) + K_D/TIME$$

$$B = K_I * (TIME/2) - K_P - 2 * (K_D/TIME)$$

$$C = K_D/TIME.$$

These values are used in the Control Algorithm (CONTROLALG) subroutine and TRAP. These values are computed before the call to TRAP for efficiency reasons.

A flag is checked to ascertain if the setpoint has converted to volts. If not, the setpoint has not been set and by default is zero. Therefore, voltage of the setpoint is 0.0 volts and GFLAG is set to 1.0 to show that conversion has occurred.

This completes all the set-up needed before entering the loop which contains the timed trap call. Variable A\$ is set to "" and while it remains "", execution of the system continues. When a keyboard input occurs, the program returns to MAIN and the main menu (Figure 7) appears. The ON TIMER(TIME) GOSUB next transfers control to a given line number when a defined period of time has elapsed. When TIME seconds have elapsed, BASICA performs a GOSUB to the specified line. The event trap occurs and BASICA starts counting again from 0. While waiting for the next event trap, the program goes into a remark loop. When control returns from TRAP, the remark loop is exited and A\$ is checked. IF A\$ does not equal "" the subroutine returns to MAIN. Otherwise, the WHILE loop continues and the event trap repeats. The event trap is disabled in the trap subroutine.

## TRAP Subroutine

TRAP subroutine is to facilitate input of the P/V converter voltage, calculate voltage required for achieving setpoint, graph three values and output the voltage to the V/I converter. Values for the level of the Processor, setpoint and voltage needed to maintain setpoint are stored two ways, on disk and by a flat-bed recorder. Figure 10 shows the flowchart for this subroutine.

The first event in TRAP is to set the variable, G, to 0 and enable control to exit the remark loop EXECUTION. Next, the multiple samples from a single channel function call is made. Ten samples at a rate of 8,000 samples per second are taken from the pressure transducer. The continuous signal is changed to a digital signal as follows:

$$V_T(k) = V_e \quad @ \quad t = KT \quad (k=0,1,2,\dots).$$

Samples are averaged to guard against false readings and remove noise. The average is computed by simply adding the ten samples and dividing by ten. The average signal is used in the calculations. Conversion of the signal from a range of 0 to 4095 is performed as follows:

$$U(k) = U^*(k) / 409.6$$

This voltage corresponds to the height (in inches) of the water in the Processor (see Table 2). Conversion subroutine (CONVERSION) to convert the voltage into inches and graph

units is called. Conversions are as follows:

Voltage to graph units -

$$\text{graph units} = U(k) * 409.6$$

Inches to graph units -

$$\text{graph units} = \text{inches} * 341.25$$

These conversions are needed for the graph subroutine, to store the data on disk and to output to the flat-bed recorder.

Next, variables used in the control algorithms are calculated.  $E(2)$  is the difference between the setpoint and level voltages ( $E(2) = V_S(k) - V_T(k)$ ). The voltage for the V/I converter ( $U(k)$ ) is computed as follows:

$$\begin{aligned} U_0(k) &= f[U_0(k-1), E(k), E(k-1), E(k-2)] \\ &= U_0(k-1) + A * E(k) + B * E(k-1) + C * E(k-2) \end{aligned}$$

$$\text{where } A = K_P + K_I * \text{TIME}/2 + K_D/\text{TIME}$$

$$B = K_I * \text{TIME}/2 - K_P - 2 * K_D/\text{TIME}$$

$$C = K_D/\text{TIME}$$

$$\begin{aligned} U(k) &= 0.0 && \text{for } U_0(k) < 0.0 \\ &= U_0(k) && 0.0 < U_0(k) < 10.0 \\ &= 9.998 && U_0(k) \geq 10.0 \end{aligned}$$

The voltage is multiplied by a scaling factor of 409.6 to convert to graph units (Jacquot, 1981). The graph unit range is 0 to 4095. Since  $409.6 * 9.998 = 4095.1$ , voltage greater than or equal to 10 is set to 9.998.

Arrays are updated in preparation for the next event trap. The data file named "B:VALSAMP.01" receives the values for the Processor level in inches, setpoint in inches, voltage to the V/I converter to correct the level and number of the sample for that set of parameters. The sample counter is incremented by one.

The simple analog output function is called to pass the voltage to the V/I converter. Values of the graph units for Processor level, setpoint and voltage to the V/I converter are relayed to the graphic subroutine. These three values are also sent to a flat-bed recorder. The setpoint and Processor level are output via simple analog output function calls. The voltage to the V/I converter is simply tapped.

A character may be read from the keyboard at this time. If a keyboard input occurs, the event trap is disabled. After returning to EXECUTION the program returns to MAIN for the next menu selection. If no keyboard occurs, TRAP returns and EXECUTION will be interrupted again.

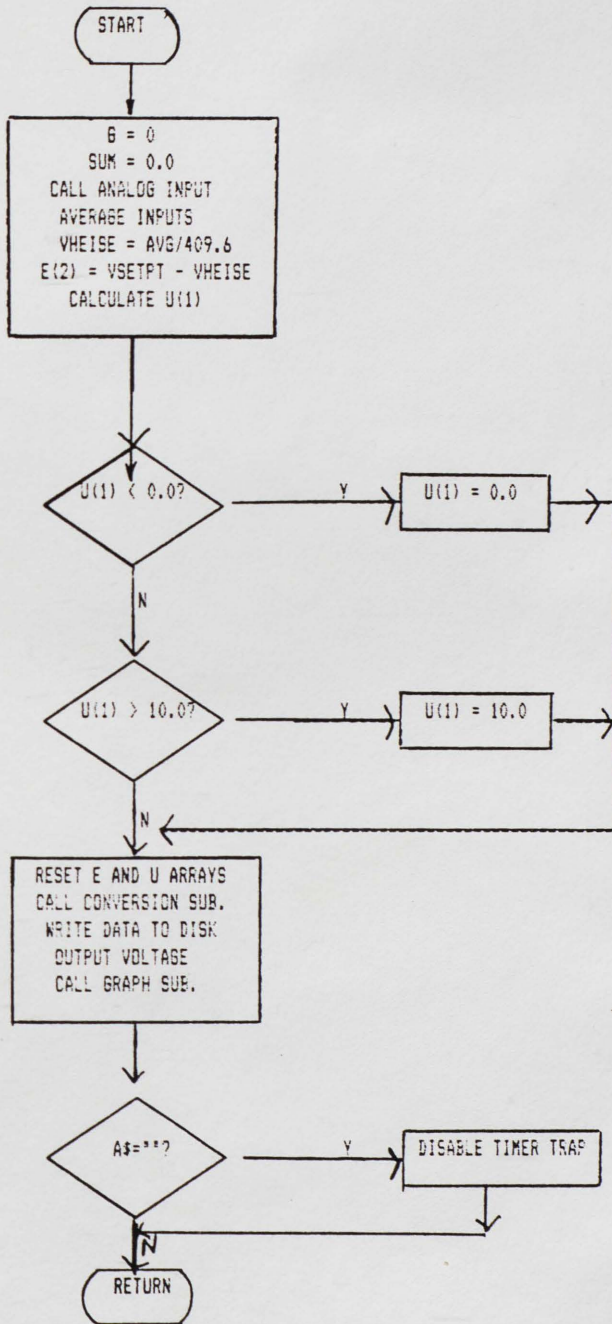


Figure 10. Trap Subroutine Flowchart.

## Setpoint Subroutine

Setpoint (SETPOINT) Subroutine allows the user to input a setpoint. Logic for this subroutine is described in Figure 11. SETPOINT displays a menu (see Figure 12) and after either selecting the default value or inputting one, the setpoint menu is again displayed. The setpoint is converted to the corresponding voltage and graph units and also stored as a string value for use in the title of the graph that appears on the screen.

Users have the option of picking the default value of 7.5 inches, inputting a different setpoint or returning to the MAIN. If default is chosen, the conversion of the setpoint to voltage and graph units is completed in SETPOINT.

When the user wishes to input a value other than 7.5, the second selection is made. A value is entered and is checked to ascertain if it is valid (between 0 and 12.0 inches). If an invalid input is made, an error message appears with the input which was made. Another input is then taken. When a valid input has been made, CONVERSION is called to convert the setpoint into a voltage corresponding to that level in the Processor. The setpoint is also converted to graphics units.

After a valid setpoint has been reached, a flag is set

to show the CONVERSION that setpoint is translated into both graphing and voltage units. This enables CONVERSION to alter both, setpoint and voltage from the pressure transducer. Each time the setpoint menu is displayed the current value of the setpoint appears on the screen. Control of the subroutine returns to MAIN when "C" is selected.

#### Conversion Subroutine

Conversion (CONVERSION) subroutine is responsible for changing the setpoint into voltage and graphics units. It also converts the P/V converter voltage into graphing units and into the corresponding inch values. The flowchart for this subroutine is shown in Figure 13. CONVERSION is called by the SETPOINT, EXECUTION and TRAP and returns to the calling subroutine when done.

The array "V" is read into memory. This array was formed by filling the Processor in half inch increments and recording the voltage from the pressure transducer at each height (see Table 2). LabSoft software includes a program called Signal Screen which facilitates this. Signal Screen displays the voltage coming in or going out of each channel which simplified the calibration of the water height in the Processor to the voltage from the P/V converter.



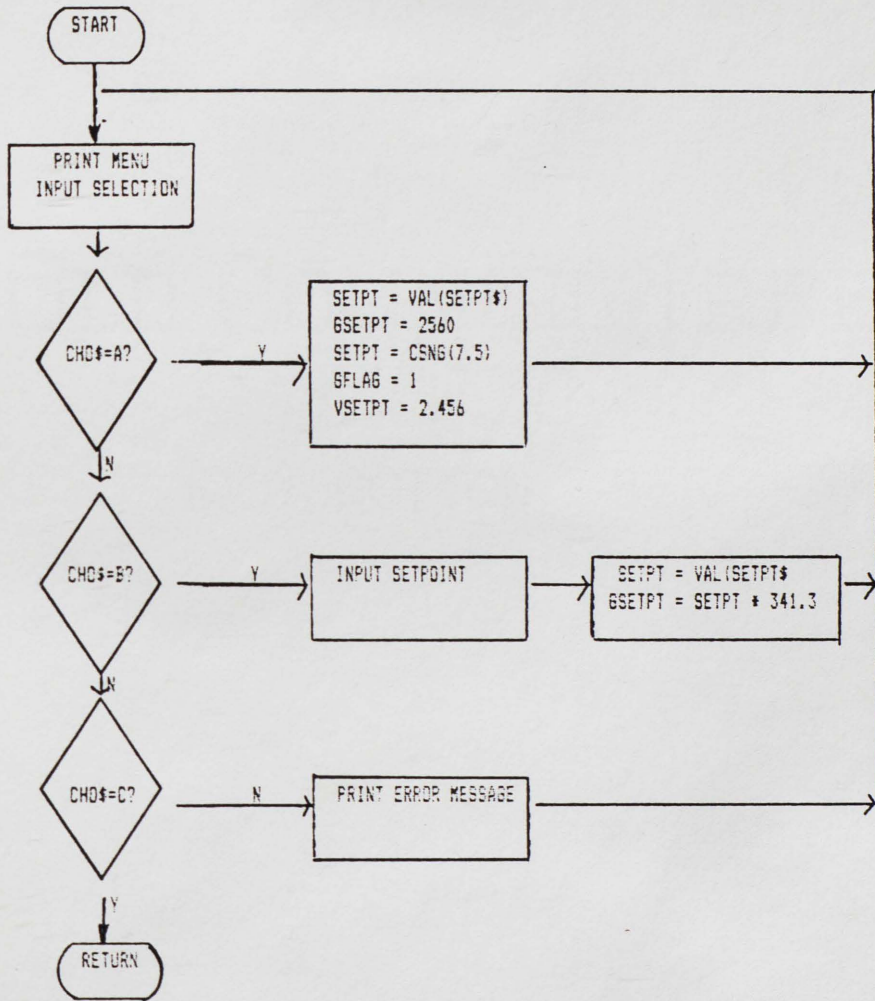


Figure 11. Setpoint Subroutine Flowchart.

SETPOINT MENU

PLEASE MAKE SELECTION.

A. DEFAULT SETPOINT (7.5 INCHES)

B. ENTER SETPOINT IN INCHES.

RETURN TO MAIN MENU.

SETPOINT = 7.5

ENTER MENU SELECTION.

Figure 12. Setpoint Menu.

The corresponding voltage of a twelve inch height was approximated using the previous data since the Processor cannot be filled to twelve inches. This enables the user to demonstrate the system reaching a maximum level.

Next, a flag in the subroutine is checked to ascertain if the setpoint or the pressure transducer voltage needs to be converted. If the flag is zero, the setpoint needs to be converted. If the flag is one, the pressure transducer voltage requires conversion and the subroutine branches to that portion of CONVERSION.

Setpoint is converted by subscript manipulation. If the setpoint is divided by 0.5, the integer result (I) is the lower subscript of the interval containing the setpoint. A check is made to see if the setpoint was divisible by 0.5 by this formula,

$$Q = \text{SETPT} - (I * 0.5).$$

If Q is 0.0, then I is the subscript needed and the voltage is the corresponding array value. The flag is set to one and the subroutine returns. If the voltage was not divisible by 0.5, interpolation is necessary using the following equation:

$$V_{\text{SETPT}} = ((V(I+1) - V(I)) * Q) / 0.5 + V(I)$$

where VSETPT is the voltage corresponding to the setpoint. After interpolating for the corresponding voltage, the flag is set to 1 and the subroutine returns.

The P/V converter voltage is converted by means of a loop to find the interval that contains the voltage. When the voltage in the array is larger than the P/V voltage, the loop is exited. The P/V voltage is first converted to the corresponding inch value by this formula:

$$IHEISE = ((VHEISE - V(J-1))/(V(J)-V(J-1))) * 0.5 + (J-1)/2$$

where VHEISE is the P/V voltage and IHEISE is the corresponding level in inches in the Processor.

The graphing units are computed by multiplying the inch units by a scaling factor of 341.25. The scaling factor is found by dividing 4096 by twelve. Twelve was used instead of 11.5 as the highest possible height that the Processor could reach. CONVERSION returns to the calling subroutine.

#### Control Algorithm Subroutine

Three different types of control are available in the control algorithm (CONTROLALG) subroutine. Proportional control is used when the actuating signal is proportional to the error. Proper selection of the gain,  $K_P$ , is necessary to eliminate or minimize the amount of offset and overshoot. Gain which is too large will impose instability on the system. Addition of integral control is desirable to completely eliminate offset. The addition of integral control has a destabilizing effect. If the system was

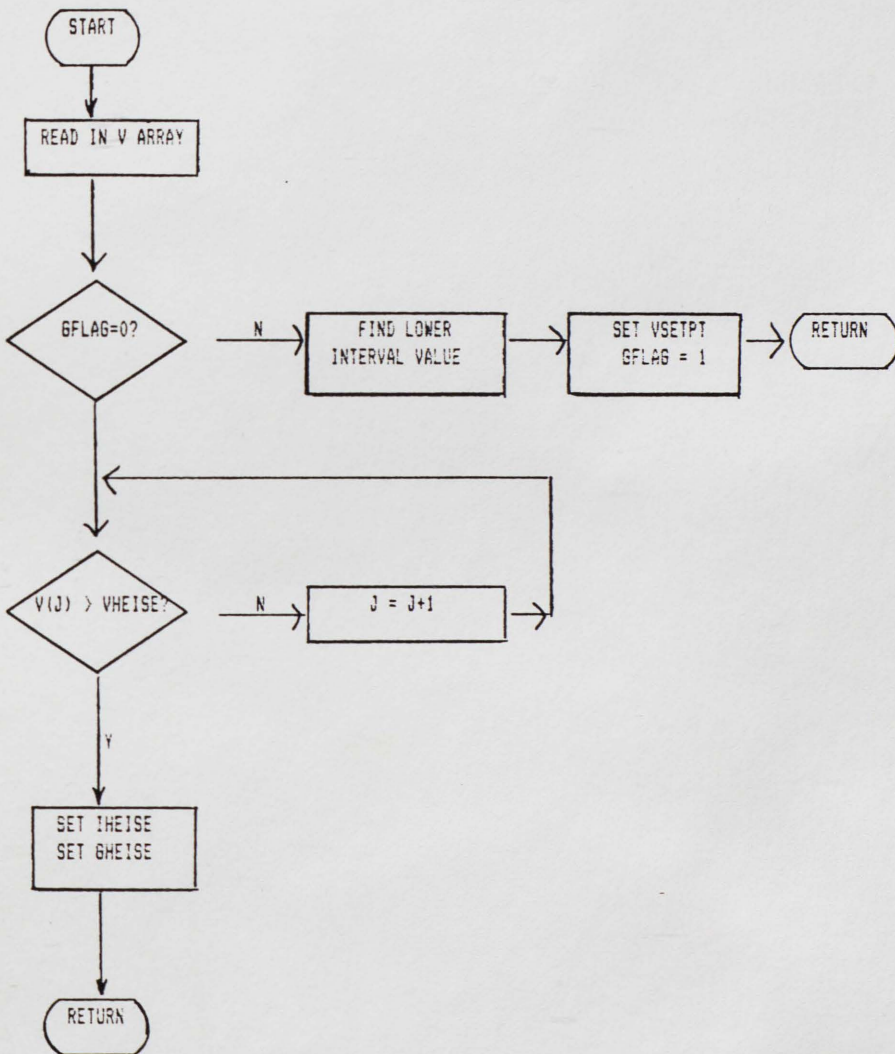


Figure 13. Conversion Subroutine Flowchart.

marginally stable, the addition of integral control may result in instability. Proportional control will tolerate a higher gain than P-I. Proportional integral derivative control will add a component to the controller output which is proportional to the time derivative of the error signal. This form of control is useful to increase system damping. Overshoot of the reference level is decreased with P-I-D control (Anand 1974).

CONTROLALG contains four menus, a main menu for the choice of control algorithms and three submenus for setting the values of KP, KI and KD. The flowchart for the CONTROLALG is displayed in Figures 14a, 14b, 14c and 14d.

A main control algorithm menu appears on the screen (see Figure 15). This menu offers four options A - Proportional Control, P; B - Proportional Integral Control, P-I; C - Proportional Integral Derivative Control, P-I-D; and D - return to the Main Menu. The current control algorithm and parameter values are also shown on the screen. CONTROLALG branches to the appropriate handler after a selection has been made. If an invalid entry is encountered, the user must enter again.

When a control algorithm has been selected, a submenu appears on the screen (Figure 16 - Proportional Parameter Menu, Figure 17 - Proportional Integral Parameter Menu, Figure 18 - Proportional Integral Derivative Parameter Menu)

offering three choices: A - to set the parameters; B - to return to the Control Algorithm Menu; and C - to return to the Main Menu. Current values of the parameters appear on the screen below each submenu.

Each of the submenu blocks employs the same logic. The control algorithm variable is set to the appropriate string ("P" for Proportional Control, "P-I" for Proportional Integral Control and "P-I-D" for Proportional Integral Derivative Control). For Proportional Control KI and KD are set to zero. For Proportional Integral Control KD is set to zero. The user is asked for a selection. The input is checked for validity. An invalid entry prompts an error message to appear and the program branches back to the submenu.

If the user wants to set the parameter(s), then the first selection of a submenu is chosen. The prompt queries for an input greater than 0.0. A check is made to ensure a valid entry is entered each time. If the entry is invalid, an error message is shown with the user's input and the proper range on the screen. The user then enters another value. This segment of the program cannot be exited until a valid entry is encountered. The program branches back to the submenu for another selection.

If the user wishes to return to the control algorithm menu, the second selection is chosen and the program

branches there. When the user indicates that return to the main menu is desired, the third choice is made and CONTROLALG returns.



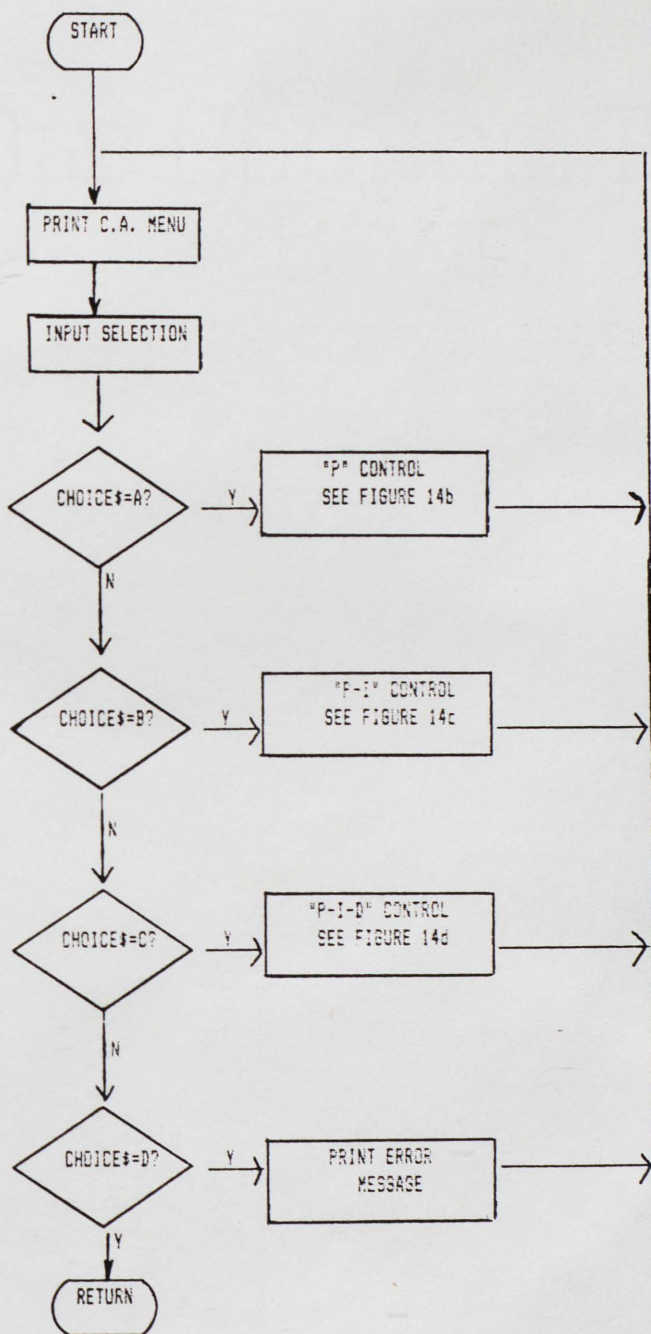


Figure 14a. Control Algorithm Subroutine Flowchart.

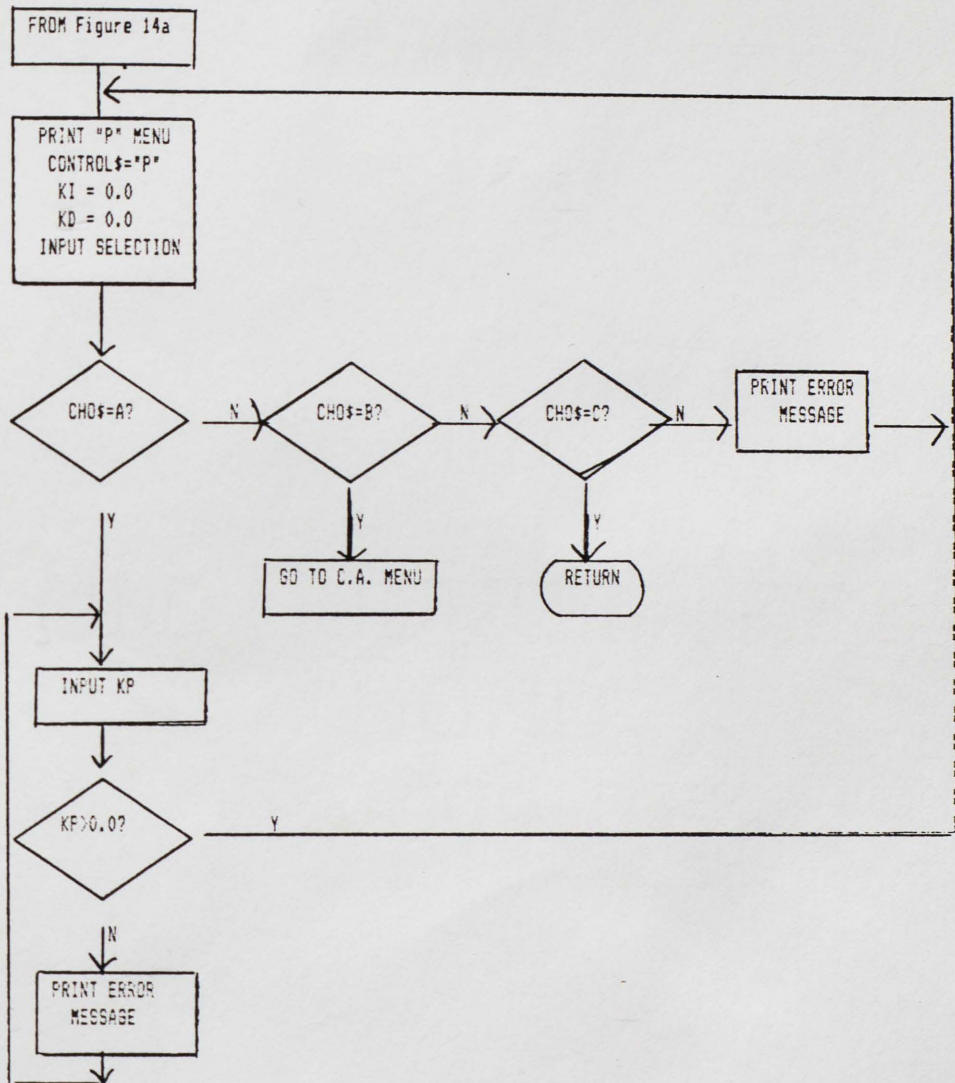


Figure 14b. Proportional Control Subunit of Control Algorithm Subroutine Flowchart.

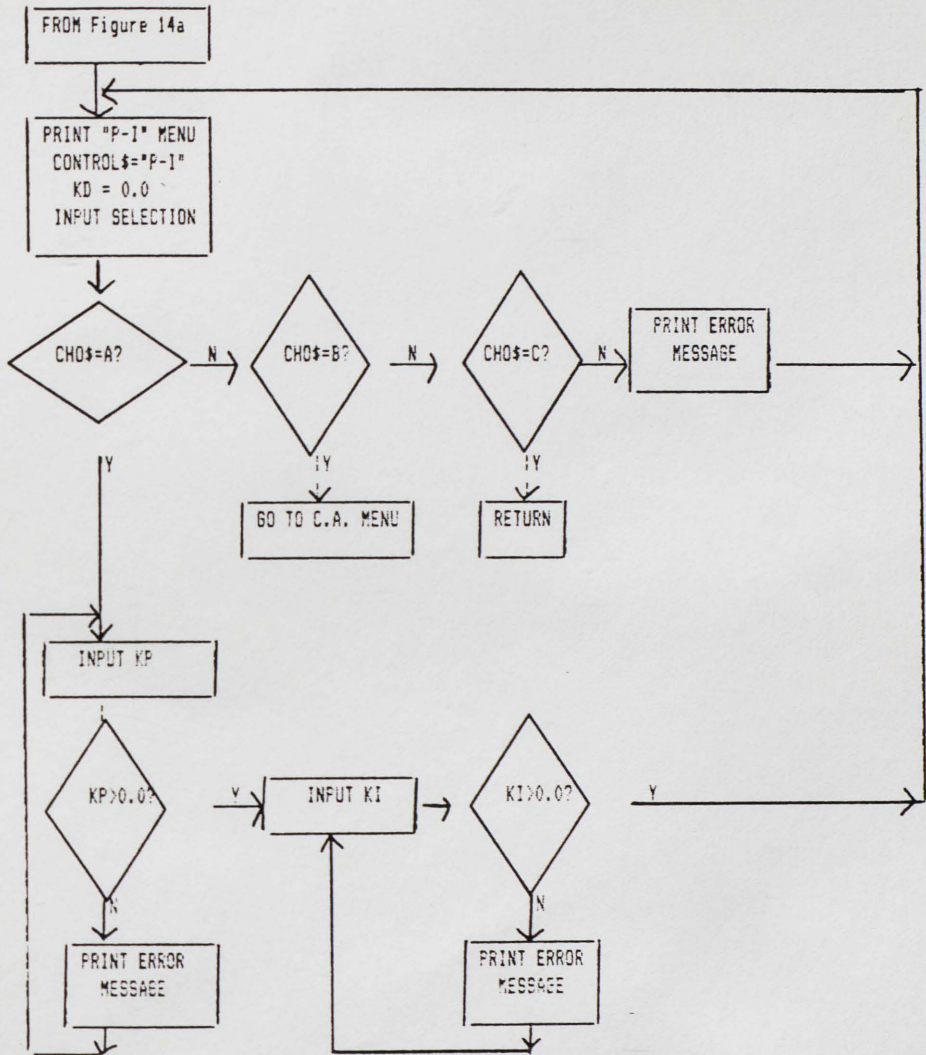


Figure 14c. Proportional Integral Control Subunit of Control Algorithm Subroutine Flowchart.

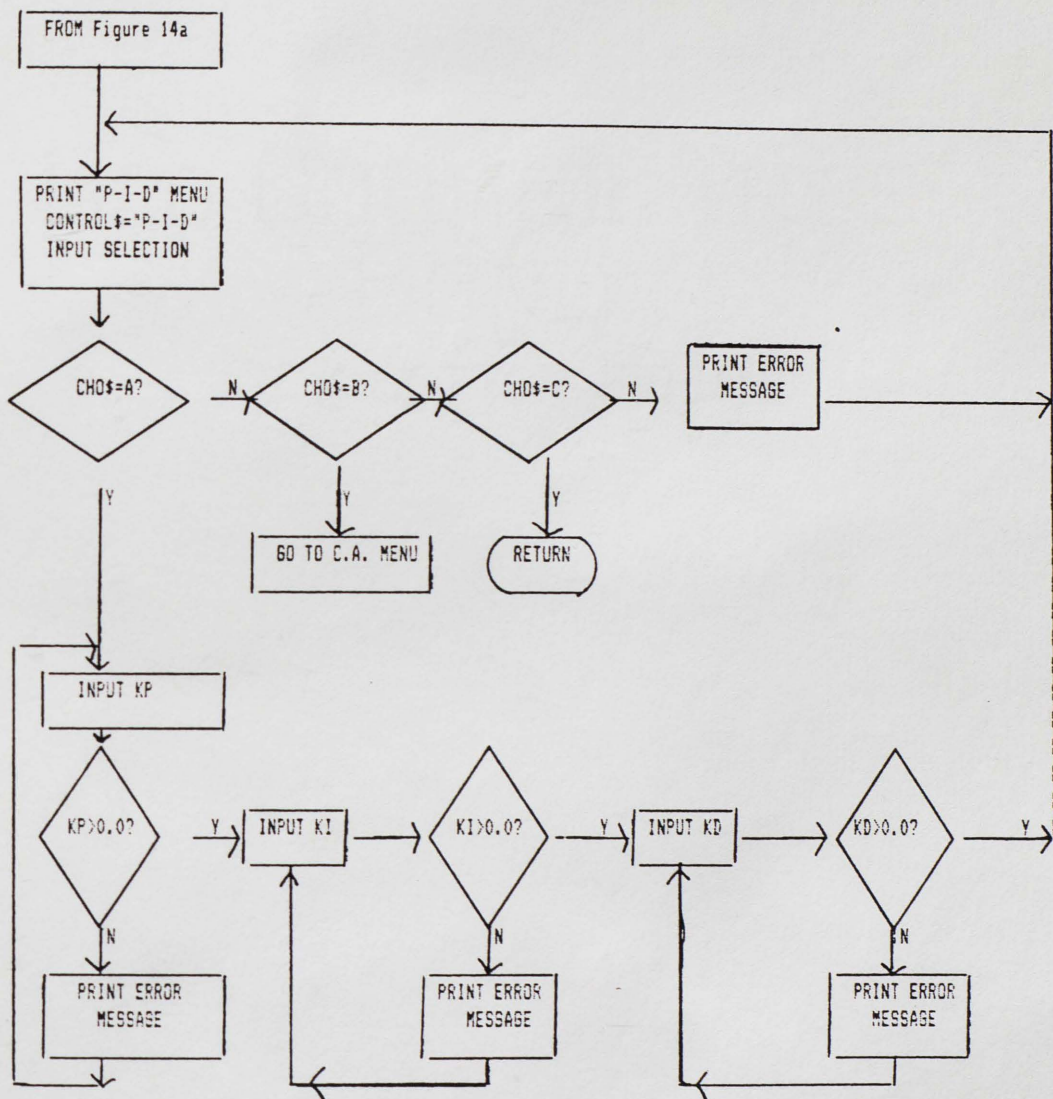


Figure 14d. Proportional Integral Derivative Subunit of Control Algorithm Subroutine Flowchart.

## CONTROL ALGORITHM MENU

PLEASE SELECT ONE.

- A. DEFAULT -- PROPORTIONAL CONTROL, P
- B. PROPORTIONAL INTEGRAL CONTROL, P-I
- C. PROPORTIONAL INTEGRAL DERIVATIVE CONTROL, P-I-D
- D. RETURN TO MAIN MENU.

CURRENT CONTROL ALGORITHM IS

KP = 0.0    KI = 0.0    KD = 0.0

ENTER SELECTION.

Figure 15. Main Control Algorithm Menu.

## PROPORTIONAL PARAMETER MENU

PLEASE MAKE SELECTION.

- A. ENTER GAIN,  $K_P > 0$ .
- B. RETURN TO CONTROL ALGORITHM MENU.
- C. RETURN TO MAIN MENU.

CURRENTLY  $K_P = 0.0$ 

ENTER MENU SELECTION.

Figure 16. Proportional Parameter Menu.

## PROPORTIONAL INTEGRAL PARAMETER MENU

- A. ENTER GAIN AND INTEGRAL CONSTANT.
- B. RETURN TO CONTROL ALGORITHM MENU.
- C. RETURN TO MAIN MENU.

CURRENTLY  $K_P = 0.0$       $K_I = 0.0$

ENTER MENU SELECTION.

Figure 17. Proportional Integral Parameter Menu.

---

## PROPORTIONAL INTEGRAL DERIVATIVE CONTROL MENU

- A. ENTER PARAMETERS  $K_P$ ,  $K_I$ , AND  $K_D$ .
- B. RETURN TO CONTROL ALGORITHM MENU.
- C. RETURN TO MAIN MENU.

CURRENTLY  $K_P = 0.0$       $K_I = 0.0$       $K_D = 0.0$

ENTER MENU SELECTION.

Figure 18. Proportional Integral Derivative Control Menu.

## CHAPTER IV - RESULTS OF EXPERIMENTS

### Comparison of Experimental Results of the Three Control Algorithms

Experimental runs of the system under direct digital control were done for the three different control algorithms: Proportional (P), Proportional-Integral (P-I), and Proportional-Integral-Derivative (P-I-D). The Processor tank was drained before each run and the input flow was completely open. The output flow was approximately the same for each run. Table 3 shows the setpoint, actual level at steady state conditions and time to achieve steady state with the three different control algorithms. The control valve was not activated using P control until a setpoint of 6.5 inches was input. The offset generated by P control decreased as the reference level increased.

Processor levels and voltages to attain setpoint were compared. A setpoint of less than 6.25 inches will not activate the control valve using proportional control. Figures 19 and 20 demonstrate the level fluctuation and control voltage oscillation of the three controls to attain 7.5 inches. The level reached with P control was 2.18 in 151 seconds. P-I and P-I-D controls both reached the

setpoint of 7.5 inches. P-I-D control reached setpoint in 110 seconds, three seconds faster than P-I control. Figures 21 and 22 illustrate the level and the control voltage of the three controls with a setpoint of three inches. The tank never filled with P control. The system ran for 104 seconds without any change in the level. P-I control attained setpoint in 41 seconds, one second more than P-I-D control. Figures 23 and 24 demonstrate the control voltage and level of the three controls reaching a setpoint of ten inches. P control took 320 seconds to reach a level of 5.672. P-I-D control attained setpoint in 318 seconds, forty seconds faster than P-I control. Figure 25 illustrates the three controls trying to attain a setpoint of twelve inches. The control voltage for each type of control was 9.998 for the tank cannot be filled to twelve inches. P control attained a level of 8.965 inches in 211 seconds. P-I control reached a level of 11.157 inches in 278 seconds. P-I-D control had a level of 11.4 inches in 287 inches.

P-I-D control reached setpoint first. Voltage oscillation was less with P-I-D control than with P-I control, but the voltage was damped using P control. P-I-D control would react to correct the voltage quicker than P-I control. This explains the differences in the voltage oscillation and level fluctuation between P-I and P-I-D



controls. These results agree with the hypothesis that proportional control has a damped voltage, but the level is offset from the setpoint. The results also show that by adding integral control the offset is eliminated, but the voltage is no longer damped. The addition of derivative control did reduce the reaction time of the system to adjust the level.

#### Comparison of the Experimental Results of the Values of the Control Parameters

Effects of using diverse values of the control parameters were explored. Experimental values of the gain,  $K_P$ , illustrated the role of  $K_P$  in the control algorithm using proportional control. A value change from one to fifty decreased the time for the level to reach its maximum under proportional control. The higher the  $K_P$  value the more the voltage would fluctuate.  $K_P$  equal to 1,000 created total instability. The voltage would swing from zero to ten. The processor level was unchecked and the Processor continued to fill until reaching the maximum level. Figures 26 and 27 show the level and voltage variations.

The gain variable was also studied under P-I control. A setpoint of 7.5 inches was used in all the experimental runs. Figures 28 and 29 illustrate the levels and control voltages of the different values of  $K_P$ . A  $K_P$  value of 100

and 1000 resulted in an unstable system. The level was unchecked. The frequency of the voltage curves was higher at  $K_P$  equal to 100 or 1000. The amplitude of the control voltage at  $K_P$  equal to 100 was three and at  $K_P$  equal to 1000 was ten. The control voltage was damped more at  $K_P$  equal to ten than at  $K_P$  equal to one.

Diverse values of  $K_I$  were used to study the system using P-I control. After a setpoint of 7.5 inches was attained, the system was allowed to run. Processor level did not vary greatly, but the voltage to maintain setpoint did. As  $K_I$  increased, the voltage fluctuations became more pronounced and the time to reach maximum level decreased. Even a value of ten caused voltage instability. Figure 30 and 31 illustrate the level and voltage curves.

The value of  $K_D$  on the control algorithm was studied using P-I-D control. The system maintained a steady level for  $K_D$  up to 100. When  $K_D$  was equal to 1,000, the system was destabilized and the level was not controlled. Voltage became extremely erratic at  $K_D$  equal to 100.  $K_D$  equal to 1,000 caused the voltage to jump from zero to ten and back to zero. An increase in  $K_D$  did not substantially alter the time to reach setpoint. Figures 32 and 33 show the level and voltage values after setpoint was reached.

The sampling time for the system can be a critical factor. If the system is checked frequently, output voltage

is updated to maintain the level. A system which is checked less frequently can become unstable due to a lack of correction. The system can either fill or drain between samplings. The system would then be overcorrected and remain unstable. The voltage would fluctuate in a piecewise amount.

Time needed to change from a higher setpoint to a lower setpoint was much greater. The output flowmeter is located higher than the relief pressure valve at the bottom of the Processor which accounts for this observation. Better results would be possible if the output flowmeter was moved lower than the relief pressure valve. The computer has no control over the output flowmeter; therefore it is the user's option to adjust the flowmeter.

The output flow was tested for different reference levels. Higher reference levels needed a lower output flow to attain setpoint. The system did not become unstable levelwise when the output flow was increased if the sampling time was small. The control algorithm would compensate and increase the output voltage.

The pressure in the Processor was measured and compared with a dynamic model equation for simulating the pressure in a tank. Table 4 lists the results. The predicted air

pressure is very close to the actual Processor pressure.

The equation is as follows:

$$P_E(t) = (14.7 H_E(t)) / (H_T - H_E(t)).$$

TABLE 3

## CONTROL ALGORITHM COMPARISONS

CONTROL ALG.	SETPOINT/STEADY STATE	LEVEL	TIME IN SEC.
P, $K_P=1$	12	8.965	211
P, $K_P=1$	3	0.00	104
P, $K_P=1$	10	5.672	320
P, $K_P=1$	7.5	2.18	151
P, $K_P=10$	7.5	4.405	92
P, $K_P=100$	7.5	0.888	100
P, $K_P=1000$	7.5	11.2	1092
P-I, $K_P=K_I=1$	12	11.157	278
P-I, $K_P=K_I=1$	3	3.056	41
P-I, $K_P=K_I=1$	10	10.016	358
P-I, $K_P=K_I=1$	7.5	7.511	113
P-I, $K_P=100, K_I=1$	7.5	7.332	287
P-I, $K_P=1000, K_I=1$	7.5	7.512	253
P-I, $K_P=10, K_I=1$	7.5	7.553	87
P-I, $K_P=1, K_I=10$	7.5	7.55	68
P-I, $K_P=1, K_I=100$	7.5	7.53	73
PID, $K_P=K_I=K_D=1$	12	11.4	287
PID, $K_P=K_I=K_D=1$	3	3.035	40
PID, $K_P=K_I=K_D=1$	10	10.007	318
PID, $K_P=K_I=1, K_D=10$	7.5	7.55	119
PID, $K_P=K_I=1, K_D=100$	7.5	7.504	122
PID, $K_P=K_I=1, K_D=1000$	7.5	7.556	116

TABLE 4

ACTUAL AIR PRESSURE IN TANK  
VS. SIMULATED AIR PRESSURE

HEIGHT IN INCHES	TANK PRESSURE	SIMULATION PRESSURE
0.0	0.0	0.0
1.5	1.6	1.633
2.0	2.2	2.26
2.5	2.9	2.94
3.0	3.6	3.675
3.5	4.4	4.473
4.0	5.2	5.345
4.5	6.2	6.3
5.0	7.3	6.68
5.5	8.4	8.51
6.0	9.7	9.8
6.5	11.2	11.24
7.0	12.8	12.86
7.5	14.7	14.7
8.0	17.0	16.8
8.5	19.7	19.22
9.0	22.7	22.05
9.5	26.1	25.34
10.0	30.3	29.4

Stopped sampling due to limitations of pneumatic indicator.

LEVEL COMPARISONS OF C.A.  
AT SETPOINT = 7.5 INCHES.

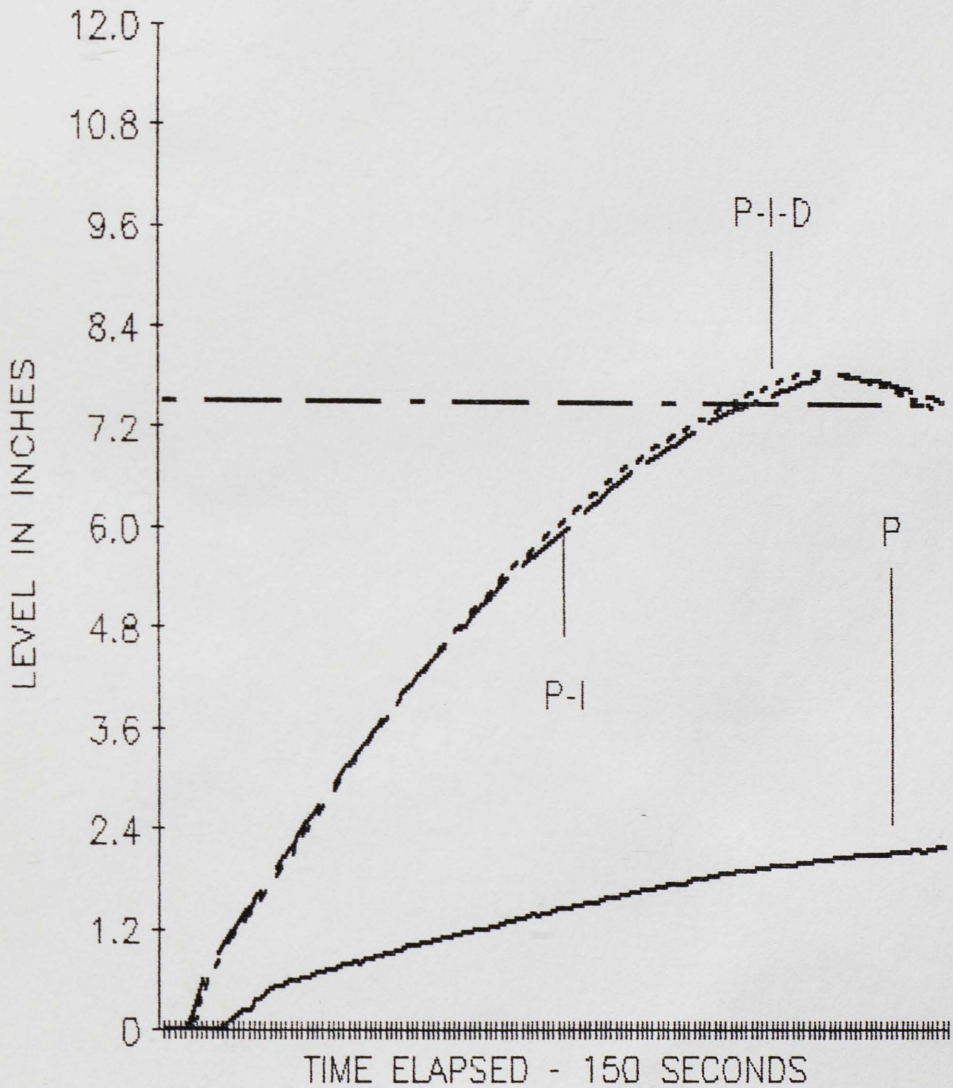


Figure 19. Level Comparisons of C.A. Attaining 7.5 Inches.

COMPARISON OF CONTROL VOLTAGES  
USING 3 CONTROL ALGORITHMS  
AT SETPOINT = 7.5 INCHES

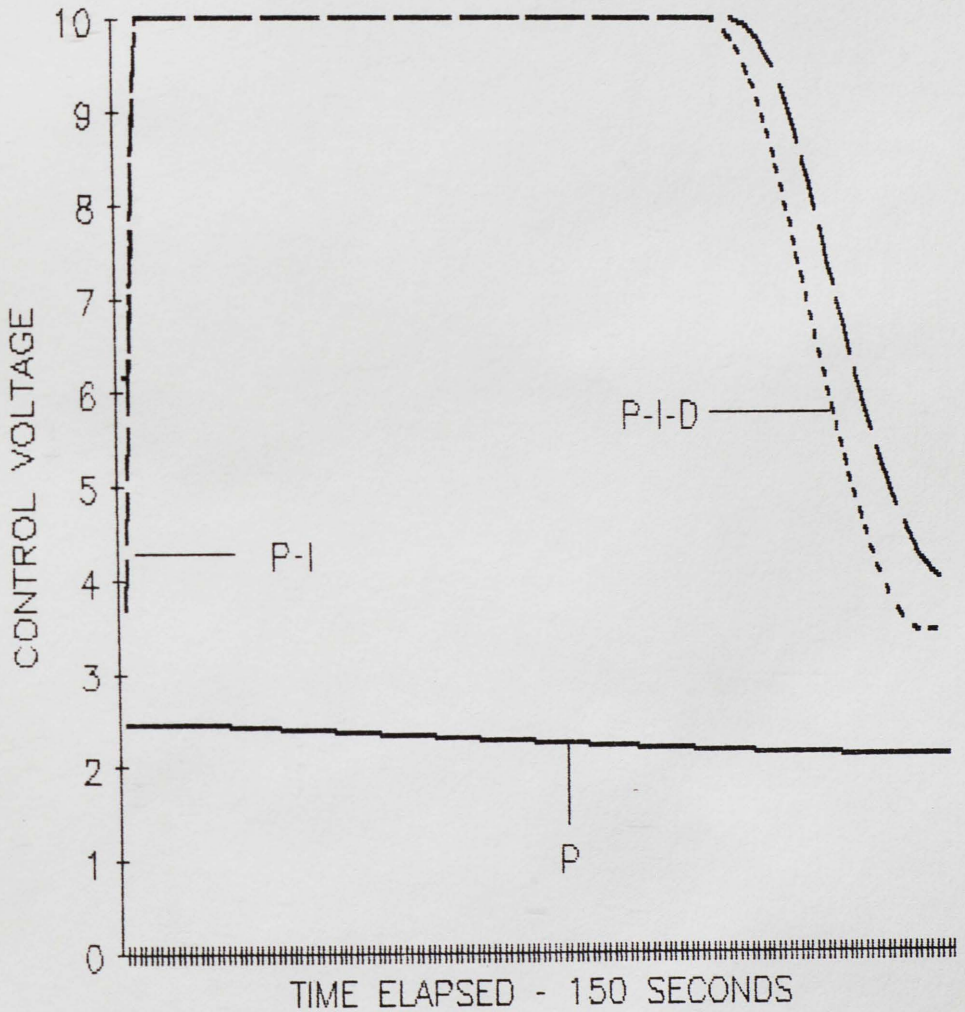


Figure 20. Comparison of Control Voltages to Attain 7.5 In.



COMPARISON OF 3 CONTROL ALGORITHMS  
 AT SETPOINT = 3 INCHES  
 "P" CONTROL NEVER DEVIATED  
 FROM LEVEL 0.0

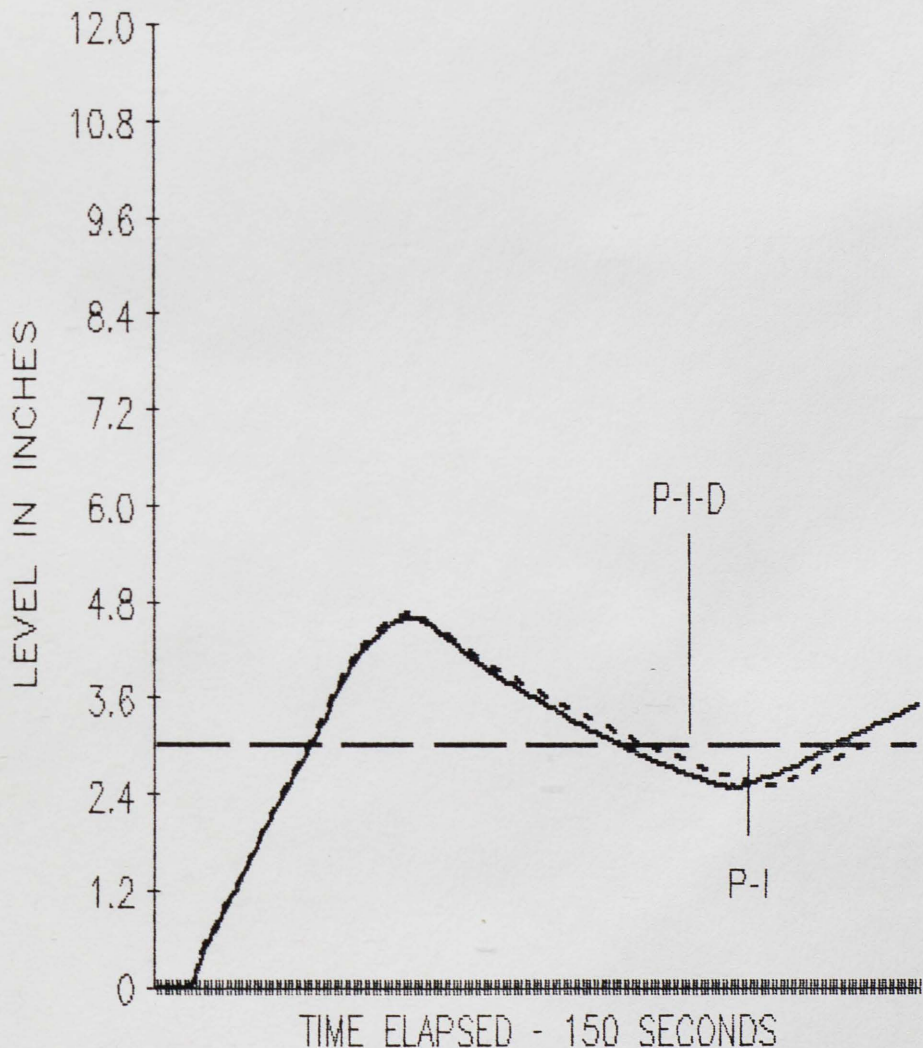


Figure 21. Levels Attained by C.A. at Setpoint = 3 in.

COMPARISON OF VOLTAGES USING  
3 CONTROL ALGORITHMS  
AT SETPOINT = 3 INCHES

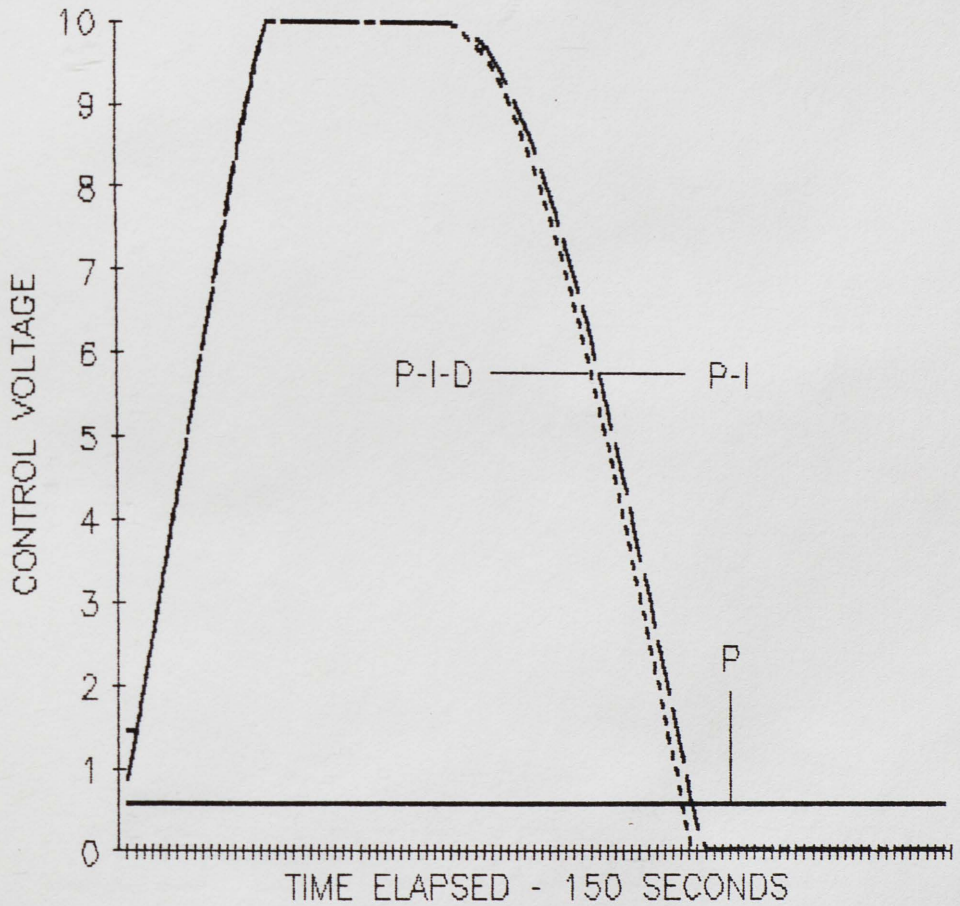


Figure 22. Comparison of Control Voltages to Attain 3.0 in.

# COMPARISON OF 3 CONTROL ALGORITHMS AT SETPOINT = 10 INCHES.

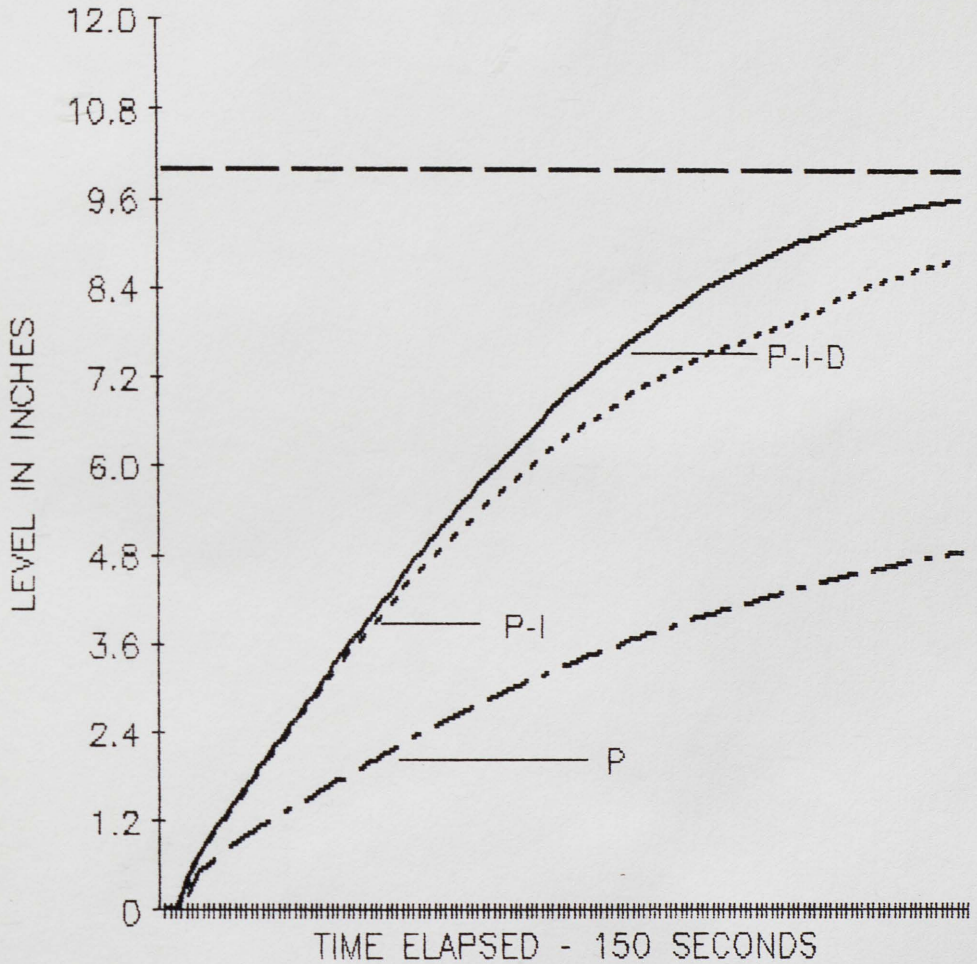


Figure 23. Comparison to C.A. to Attain 10 in.

COMPARISON OF VOLTAGES USING  
3 CONTROL ALGORITHMS  
AT SETPOINT = 10 INCHES.

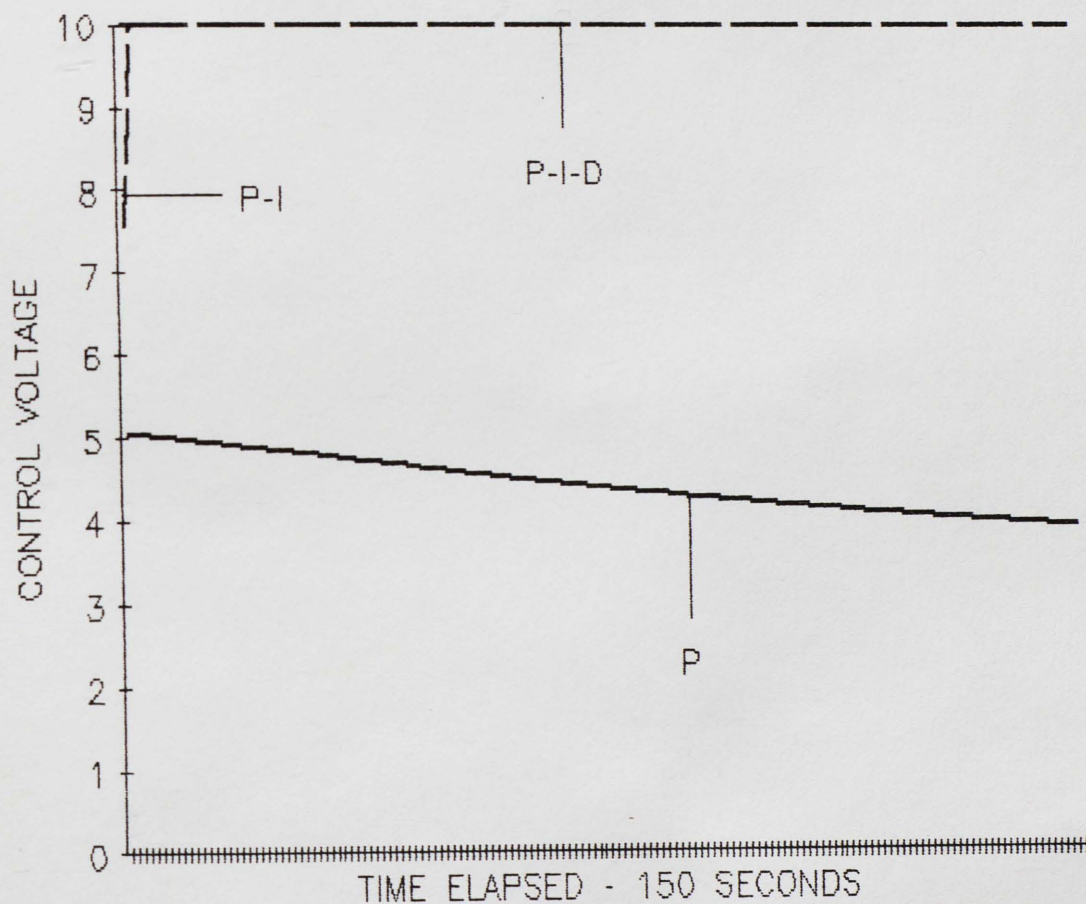


Figure 24. Control Voltages to Attain Setpoint = 10.0 in.

# COMPARISON OF 3 CONTROL ALGORITHMS AT SETPOINT = 12 INCHES.

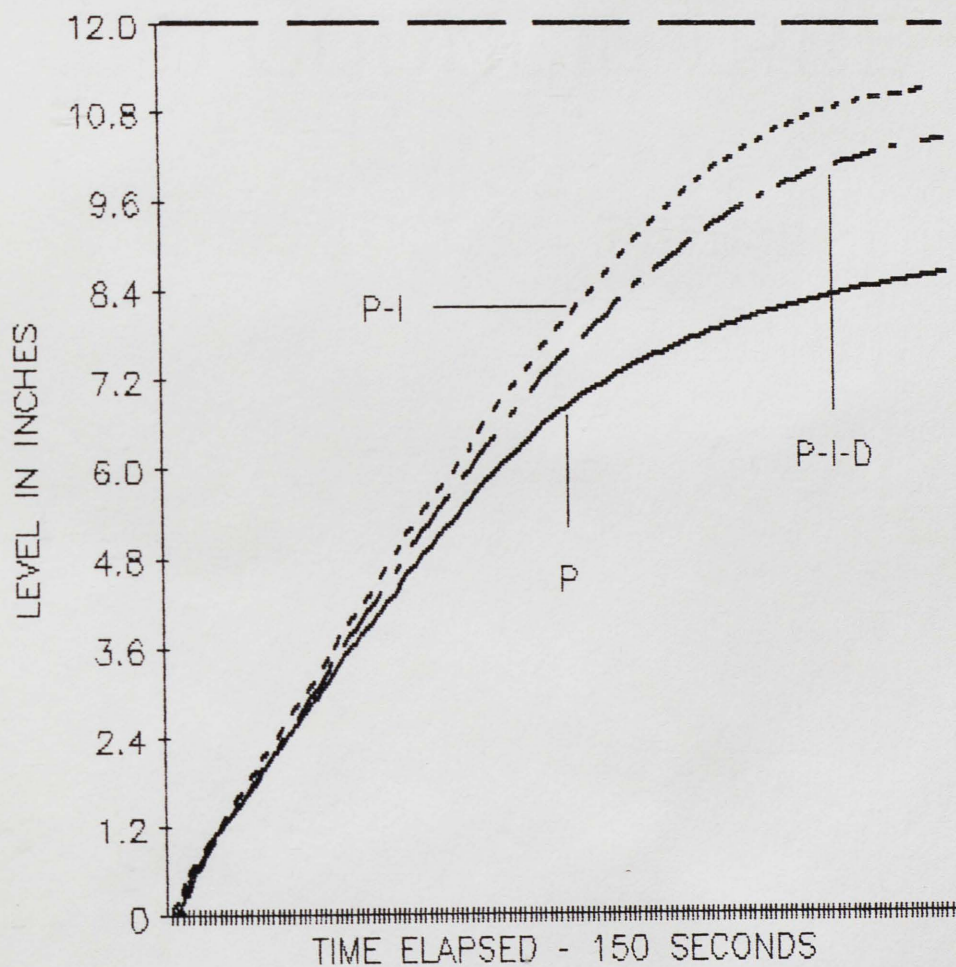


Figure 25. System At Maximum Level.

PROPORTIONAL CONTROL WITH  
DIVERSE  $K_P$  VALUES  
SETPOINT = 7.5 INCHES.

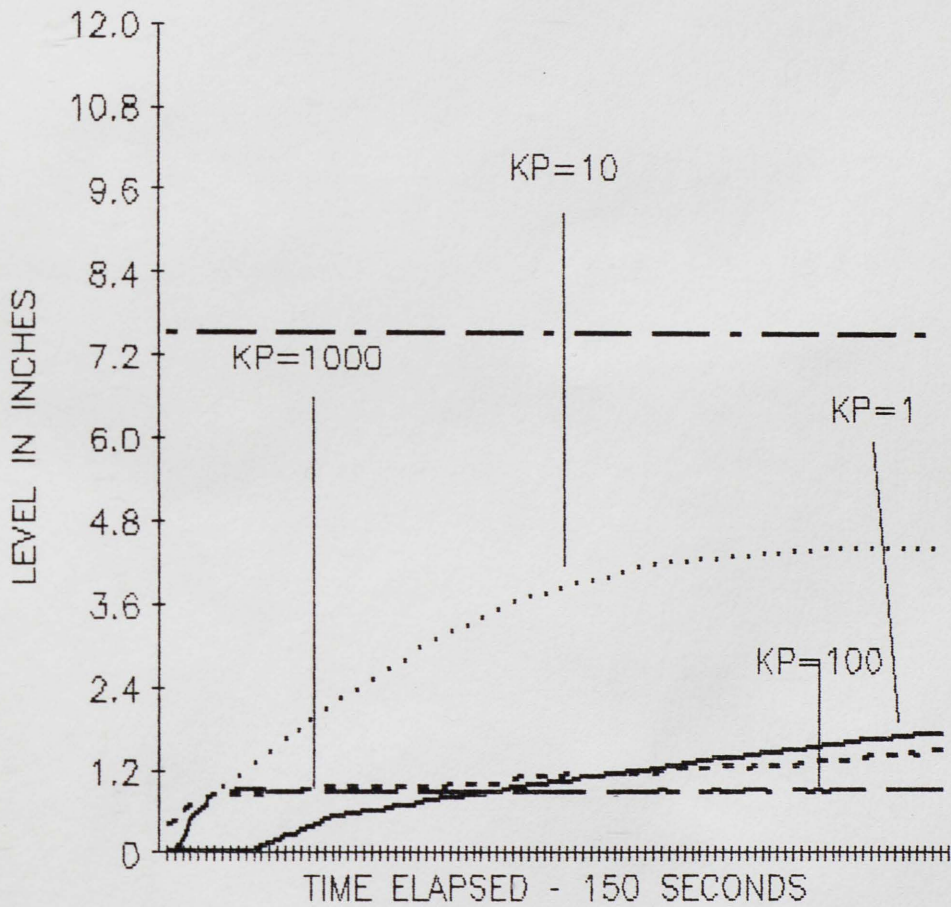


Figure 26. Levels with Diverse  $K_P$  Values.

PROPORTIONAL CONTROL WITH  
DIVERSE  $K_P$  VALUES  
SETPOINT = 7.5.

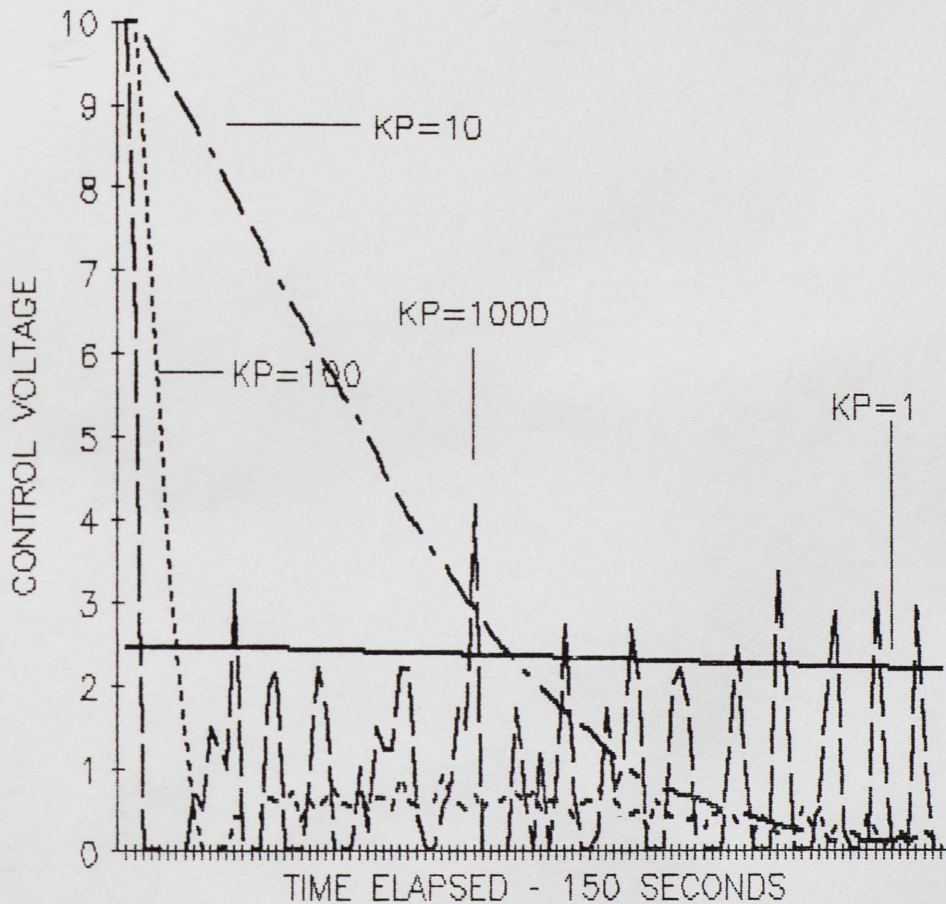


Figure 27. Control Voltages with Diverse  $K_P$  Values.

P-I CONTROL WITH VARIOUS  
VALUES FOR  $K_P$   
SHOWING PROCESSOR LEVEL  
 $K_I = 1.$

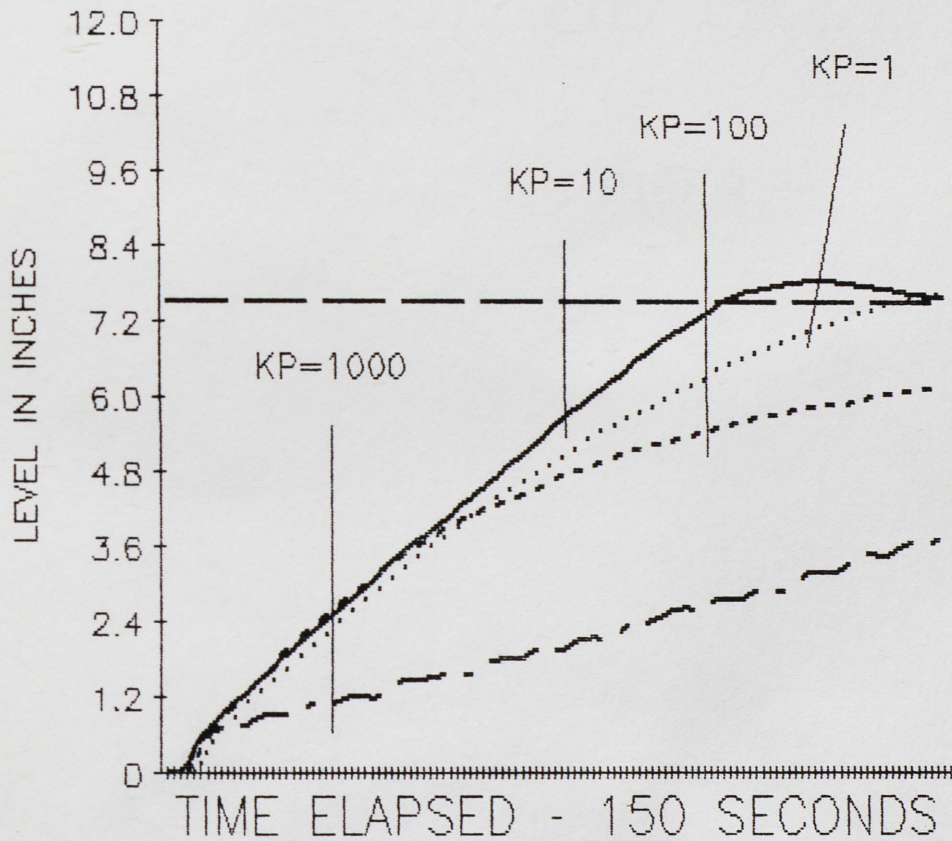


Figure 28. P-I Levels with Diverse  $K_P$  Values.



VOLTAGES OF P-I CONTROL  
WITH VARIOUS VALUES OF  $K_P$   
 $K_I = 1.$

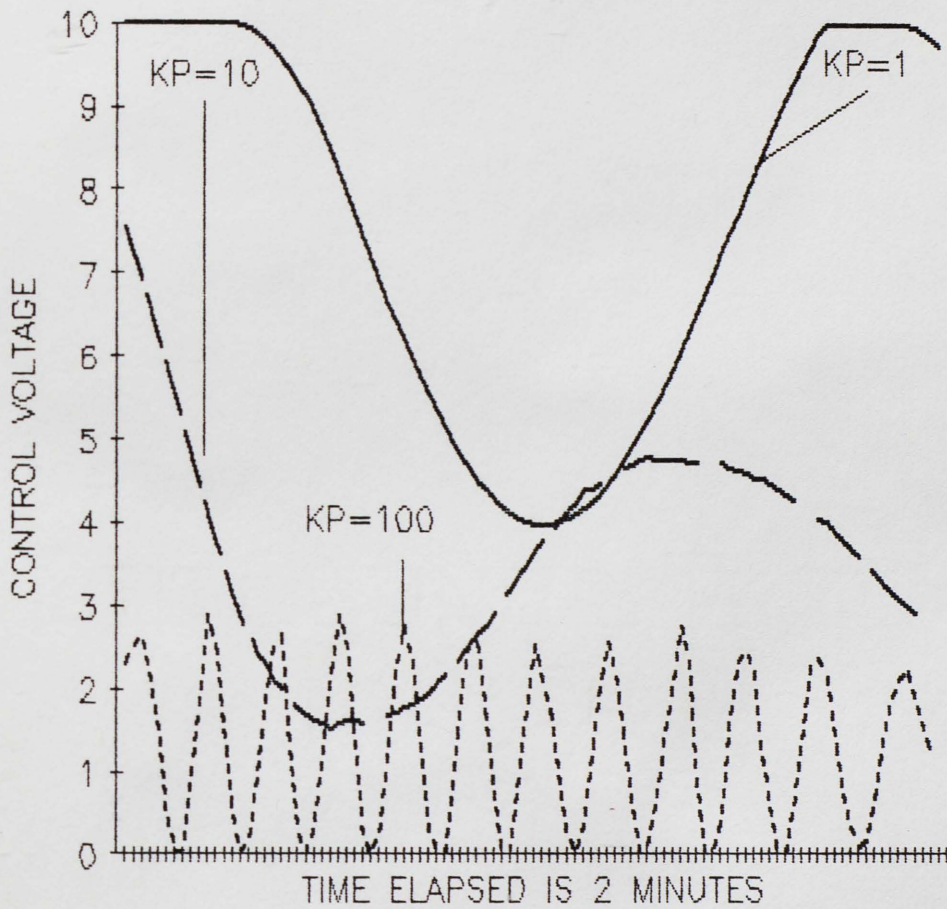


Figure 29. Control Voltages with Various  $K_P$  Values,  $K_I = 1.$

PROCESSOR LEVELS AFTER SETPOINT  
OF 7.5 WAS REACHED WITH VARIOUS  
VALUES OF  $K_I$ .  $K_P=1$ .

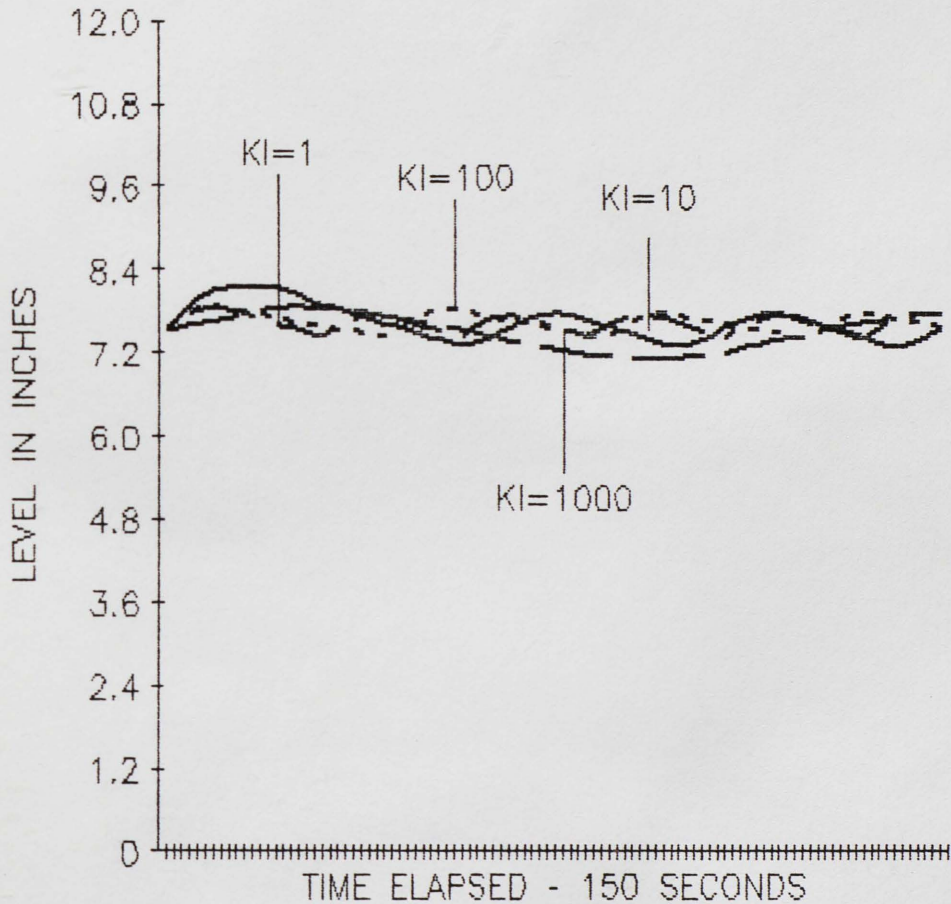


Figure 30. Level Fluctuations With Various  $K_I$  Values.

VOLTAGES TO MAINTAIN SETPOINT  
WITH VARIOUS VALUES OF  $K_I$ .  $K_P=1$ .

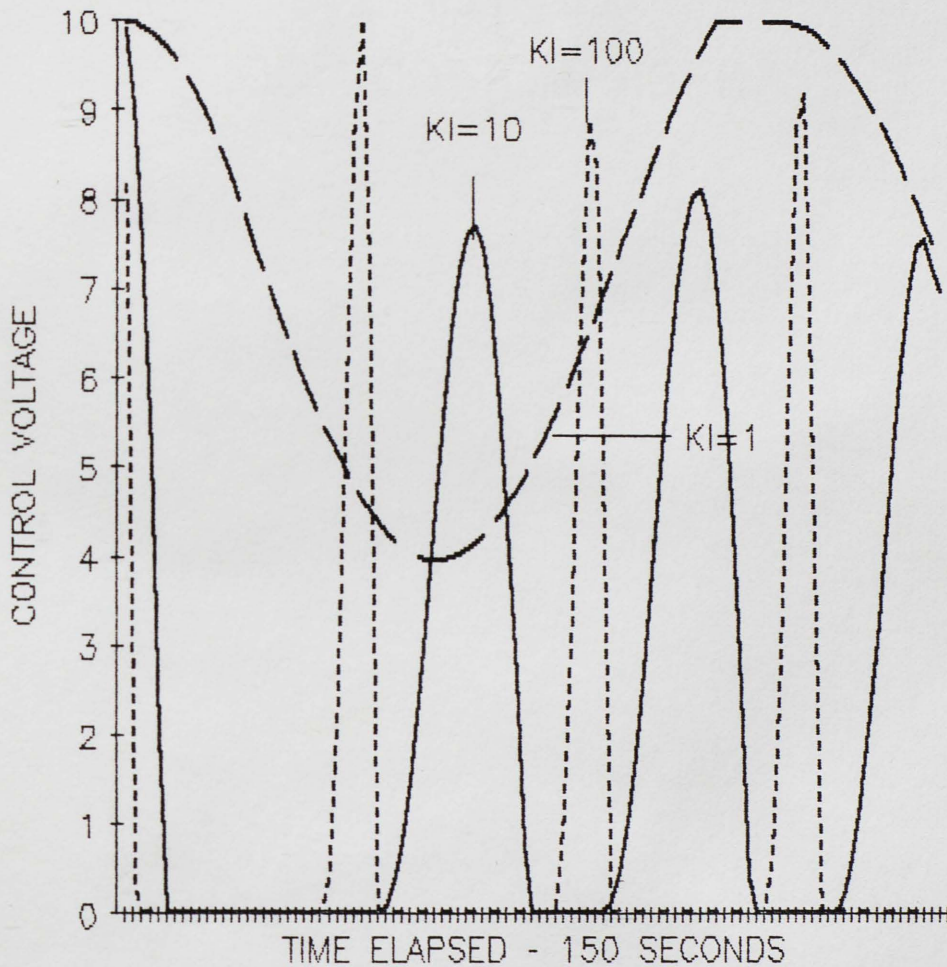


Figure 31. Control Voltages with Various  $K_I$  Values.

PROCESSOR LEVEL WITH VARIOUS VALUES  
OF KD AFTER SETPOINT IS ATTAINED.  
 $K_P = K_I = 1.$

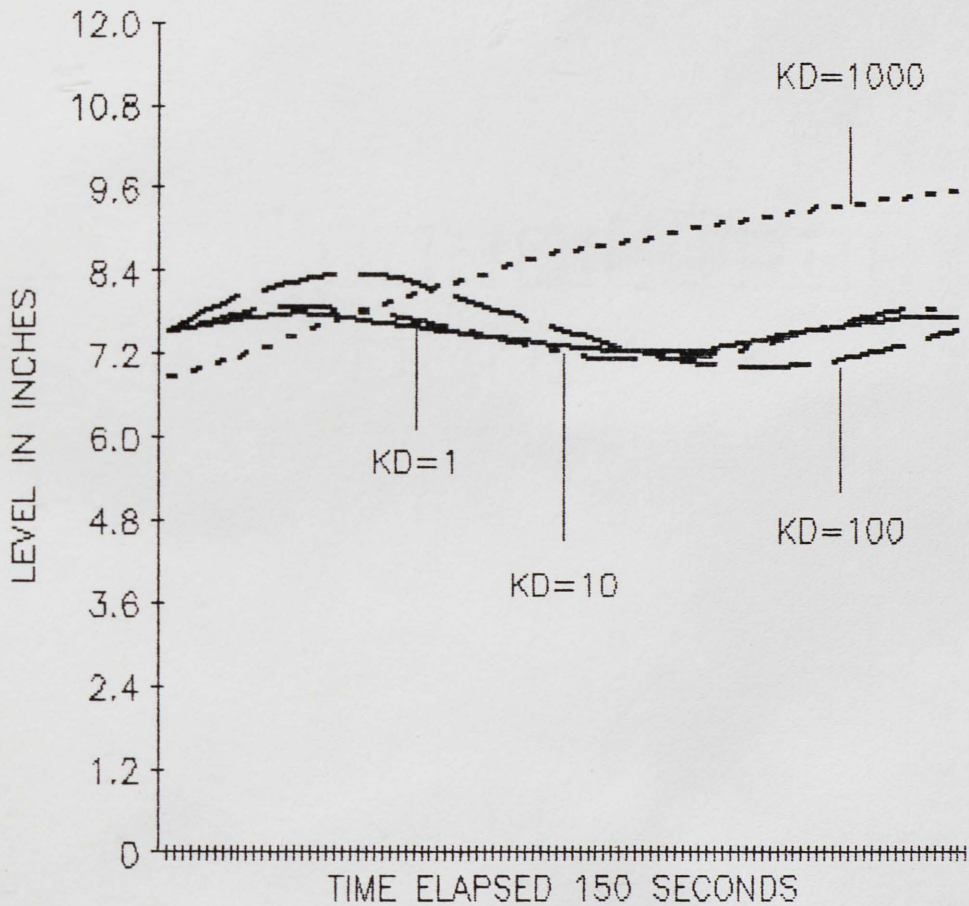


Figure 32. Processor Levels with Various  $K_D$  Values.

VOLTAGES AFTER ATTAINING SETPOINT  
USING PID CONTROL WITH VARIOUS  
VALUES OF  $K_D$ .  $K_P = K_I = 1$ .

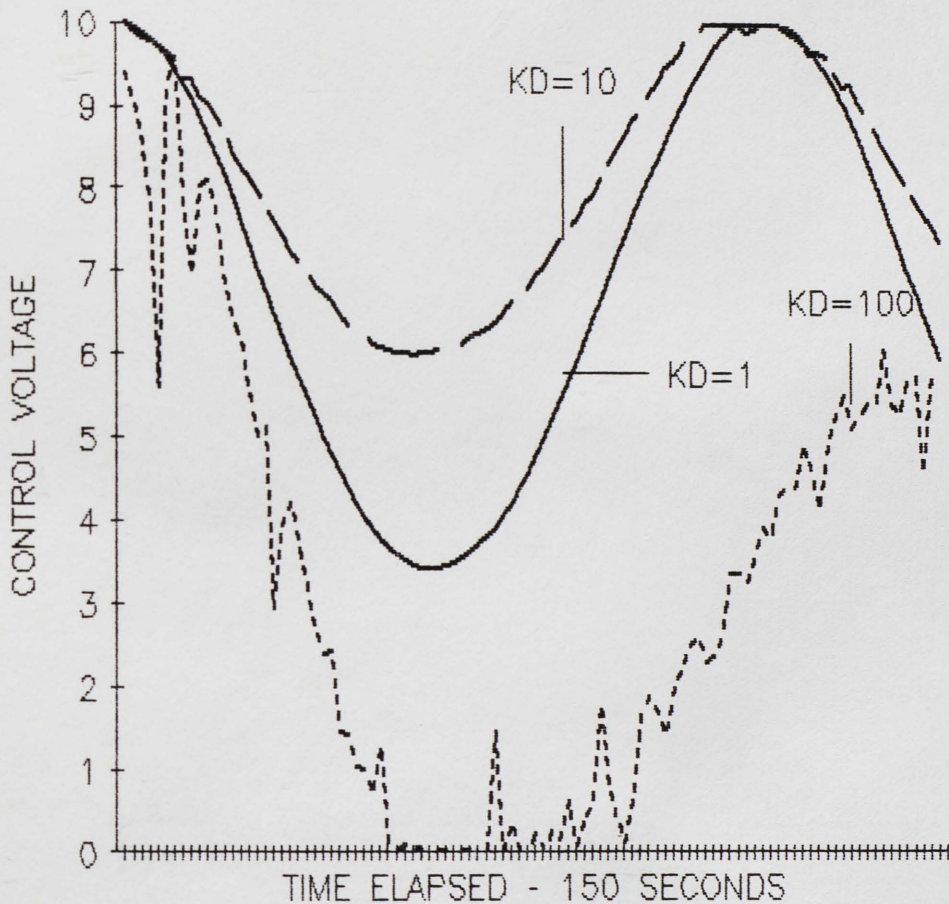


Figure 33. Voltages with Various  $K_D$  Values.

## CHAPTER V - CONCLUSIONS

The control program developed in this paper is capable of controlling the TECHNOVATE Model 9030 satisfactorily over a range of zero to twelve inches Processor level. The program itself is user friendly and can be used for a laboratory situation. The offset error can be attributed to proportional control mode. The addition of integral control eliminated the offset error. The further addition of derivative control did increase voltage damping and decrease overshoot and undershoot of the reference level. The sampling time restriction of a sampling time not less than one second did not limit the response of the system. The system is pneumatic and therefore has a slow response time. The position of the output flow needs to be lowered to remove a load on the system.

A further suggestion for work on the system is an extensive study of the control algorithm its different parameters be implemented. The optimum range for each parameter of the different types of control could be found. The various control algorithms available could be studied and compared using digital control. Further work is needed to digitally regulate the input/output flows for the system.

## APPENDICES

APPENDIX A  
SYSTEM CONFIGURATION



The settings for the switches in the ISAAC chassis and the interface are located in Table 5. A block diagram showing the setup of the different instruments is shown in Figure 34. The connections for the system are listed in Table 6. Table 7 defines the settings for the H-P power supply.

On the LINSEIS Flat-bed Recorder the colors corresponding to the variables are RED - setpoint, BLUE - Processor level and GREEN - volts to achieve or maintain setpoint. The analog channels on the ISAAC that correspond to the parameters are output channel 1 - GSETPT (setpoint in graph units), output channel 2 - GHEISE (HEISE voltage in graph units, actual level in processor) and channel 0 for U(1) (voltage to achieve /maintain setpoint). This is demonstrated in Figure 35. Range of the setpoint and processor level is zero to twelve inches. Voltage range is zero to ten volts.

TABLE 5

## SWITCH SETTINGS FOR ISAAC HARDWARE

SWITCH	POSITION
<u>CHASSIS SWITCHES</u>	
UNIPOLAR INPUT VOLTAGE RANGE	
S7/1	OFF
S7/2	ON
S7/3	OFF
NO ANALOG INPUT GAIN	
S7/4	OFF
SINGLE ENDED INPUT TYPE	
S8/1	OFF
S8/2	OFF
UNIPOLAR OUTPUT VOLTAGE RANGE	
S9/1	OFF
S9/2	ON
S9/3	OFF
+/- 5 VOLTS	
S10	OFF
SCHMITT TRIGGER OFF	
EXTERNAL POWER SUPPLY	
S11/1-8	OFF
EXTERNAL POWER SUPPLY	
S12	EXT
NO ANALOG INPUT DIVIDING	
S A/D 1-2	OFF
<u>INTERFACE CARD SWITCHES</u>	
LEVEL 7 INTERRUPT	
U6/1-5,1-3	OFF
U6/4-5	ON
NO ADDITIONAL INTERFACE CARDS	
U7/1	OFF
U7/2	OFF

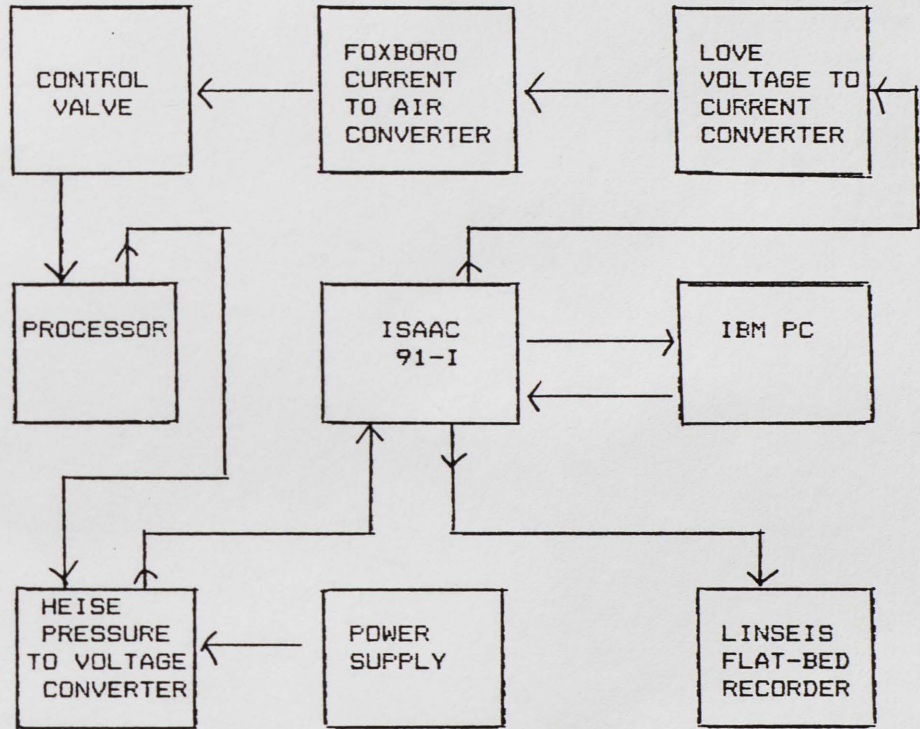


Figure 34. Block Diagram of Instruments in the System.

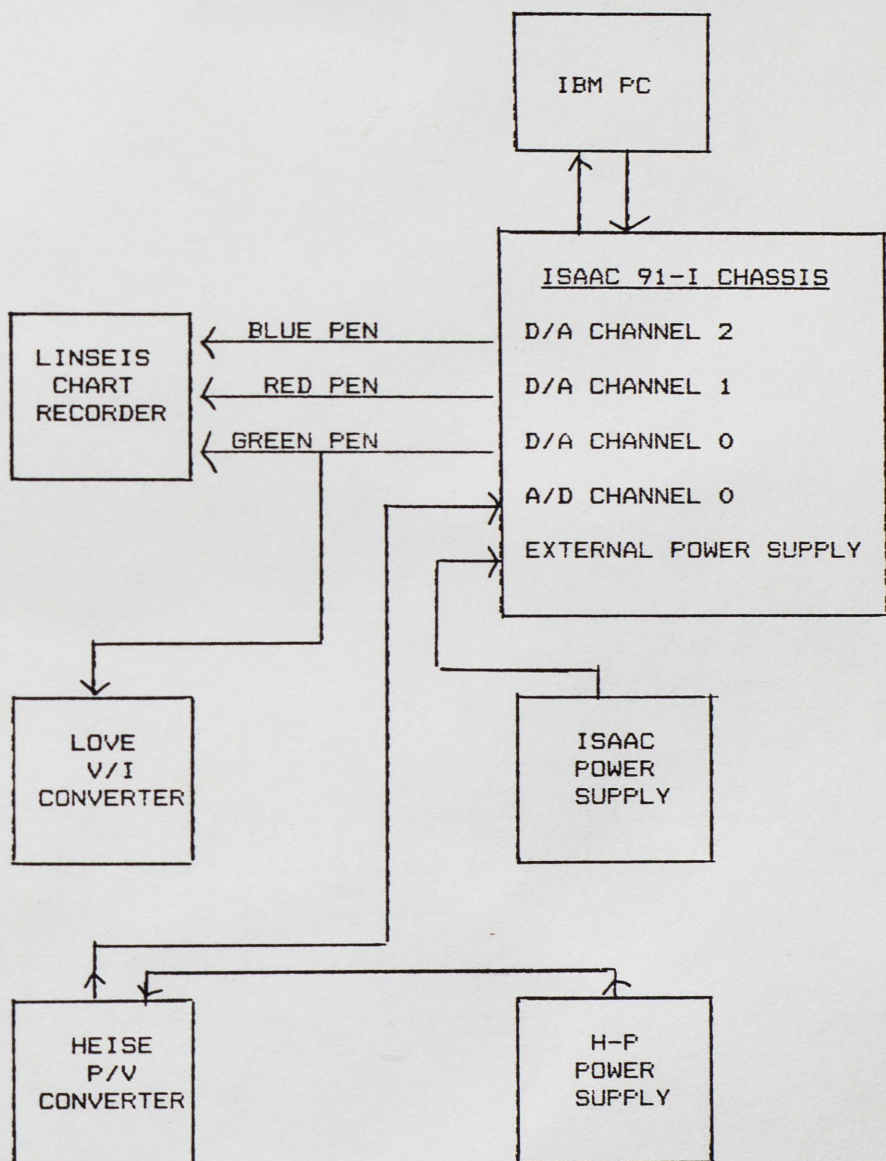


Figure 35. ISAAC 91-I Connections.

TABLE 6

## INSTRUMENT CONNECTIONS

## CONTROL VALVE

Pneumatic hose from Foxboro to QC2

## PROCESSOR

Pneumatic hose from QC3 to Heise

## HEISE

Green wire	ISAAC 91-I A/D CHANNEL 0
White wire	ISAAC 91-I A/D GROUND
Orange wire	H-P COM outlet
Black wire	H-P -20V outlet
Pneumatic hose	QC3

## LOVE

Pink wire	+ of Foxboro
Grey wire	- of Foxboro
White wire	ISAAC 91-I D/A GROUND
Green wire	ISAAC 91-I D/A CHANNEL 0
Plug	110V

## FOXBORO

Pneumatic hose QC2

## ISAAC 91-I

A/D CHANNEL 0	HEISE green wire
A/D GROUND	HEISE white wire
D/A CHANNEL 0	LOVE green wire

## LINSEIS

Blue pen	ISAAC 91-I D/A CHANNEL 2
Red pen	ISAAC 91-I D/A CHANNEL 1
Green pen	ISAAC 91-I D/A CHANNEL 0
3 - Black (grounds)	ISAAC 91-I A/D GROUND

TABLE 7

HEWLETT-PACKARD TRIPLE OUTPUT  
POWER SUPPLY CONNECTIONS AND SETTINGS

## KNOB SETTINGS

Meter	+20V
Voltage +6V	Not used
Tracking Ratio	Not used
Voltage +/-20V	Turned completely to right
Line	ON

## OUTLETS

+6V	Not used
COM	Orange wire of HEISE
+20V	Not used
-20V	Black wire of HEISE
GND	Not used

APPENDIX B  
PROGRAM LISTING

```
1000 'INTRODUCTORY SCREEN BLOCK
1010 CLS
1020 LOCATE 5,5: PRINT "WELCOME TO THE COMPUTER CONTROLLED
      TECHNOVATE SYSTEM."
1030 LOCATE 7,5: PRINT "THIS SYSTEM CONTROLS THE LEVEL OF
      THE PROCESSOR."
1040 LOCATE 9,5: PRINT "ON THE FIRST RUN OF THE SYSTEM THE
      FOLLOWING IS NECESSARY:"
1050 LOCATE 10,10: PRINT "1. INPUT THE SETPOINT."
1060 LOCATE 11,10: PRINT "2. SET THE CONTROL ALGORITHM."
1070 LOCATE 12,10: PRINT "3. SET THE CONTROL ALGORITHM
      PARAMETERS."
1080 LOCATE 14,5: PRINT "AFTER THESE HAVE BEEN SET THE
      SYSTEM IS READY TO RUN."
1090 LOCATE 16,5: PRINT "IF DATA IS SENT TO DATA DISK,
      PLEASE PLACE DATA DISK IN DRIVE B."
1100 LOCATE 18,5: PRINT "IF DATA IS SENT TO PRINTER THEN
      HAVE PRINTER ON LINE."
1110 LOCATE 21,5: INPUT "DO YOU WISH TO SAVE DATA ON DISK?
      Y/N?", B$
1120 ' DFLAG IS FLAG TO STORE DECISION ON SAVING DATA ON
      DISK.
1130 IF B$ = "Y" OR IF B$ = "y" THEN DFLAG = 1
1140 IF B$ = "N" OR B$ = "n" THEN DFLAG = 0
1150 IF DFLAG = 0 OR DFLAG = 1 GOTO 1170
```



```
1160 LOCATE 22,5: PRINT "INPUT DATA DECISION AGAIN."  
1170 GOTO 1110  
1180 'BLOCK TO PRINT MAIN MENU  
1190 'VARIABLES ARE CHOICE$,S$  
1200 'BLOCKS CALLED ARE SETPOINT MENU, SHUTDOWN,  
1210 'SET ALGORITHMS AND BEGIN  
1220 'EXECUTION BLOCKS  
1230 '  
1240 ' DIMENSION ARRAYS, INITIALIZE E U ARRAYS AND CONTROL$  
    VARIABLE.  
1250 DIM E(2),U(1), V(24), RAW%(99)  
1260 CONTROL$ = "" 'INITIALIZE CONTROL VARIABLE.  
1270 E(1) = 0!  
1280 E(0) = 0!  
1290 CONTROL$ = ""  
1300 U(0) = 0!  
1310 IF DFLAG = 0 GOTO 1440  
1320 'OPEN FILE FOR WRITING DATA.FILE NAME IS "VALSAMP.01"  
    AND IS IN "B" DRIVE.  
1330 ' DATA TO BE SAVED ARE LEVEL (IHEISE),  
1340 ' SETPOINT (SETPT), VOLTAGE TO ACHIEVE/MAINTAIN  
    SETPOINT (U(1))  
1350 ' AND SAMPLE NUMBER FOR THAT RUN (X).  
1360 OPEN "B:VALSAMP.01" FOR OUTPUT AS #1  
1370 '
```

```
1380 'OPEN FILE FOR WRITING DATA. FILE NAME IS "VARVAL" AND
      IS IN "B" DRIVE.
1390 ' DATA SAVED IS KP, KI, KD, CONTROL$ AND SAMPLING TIME.
1400 OPEN "B:VARVAL" FOR OUTPUT AS #2
1410 '
1420 'SET UP SCREEN FORMAT
1430 '
1440 KEY OFF
1450 WIDTH 80
1460 CLS
1470 '
1480 '
1490 LOCATE 3,20: PRINT "*****
      *****"
1500 LOCATE 4,20: PRINT "*"
      "*"
1510 LOCATE 5,20: PRINT "* WELCOME TO THE TECHNOVATE
      SYSTEM *"
1520 LOCATE 6,20: PRINT "*" MAIN MENU
      "*"
1530 LOCATE 7,20: PRINT "*"
      "*"
1540 LOCATE 8,20: PRINT "*" PLEASE SELECT ONE
      "*"
```

```

1550 LOCATE 9,20: PRINT "*" A. SET/CHANGE SETPOINT VALUE.
      "*"
1560 LOCATE 10,20: PRINT "*" B. SET/CHANGE CONTROL
      ALGORITHMS "*"
1570 LOCATE 11,20: PRINT "*" C. BEGIN EXECUTION OF SYSTEM
      "*"
1580 LOCATE 12,20: PRINT "*" D. SHUT DOWN SYSTEM
      "*"
1590 LOCATE 13,20: PRINT "*"
      "*"
1600 LOCATE 14,20: PRINT "*****"
      "*****"
1610 ' TAKE INPUT
1620 LOCATE 19,1: INPUT "ENTER MENU SELECTION. ", CHOICE#
1630 ' BRANCH TO CORRECT SUBROUTINE.
1640 '
1650 IF CHOICE# = "A" GOTO 1670
1660 IF CHOICE# <> "a" GOTO 1690
1670 GOSUB 2230 ' SETPOINT SELECTED.
1680 GOTO 1440 ' BRANCH BACK TO MAIN
      MENU.
1690 IF CHOICE# = "B" GOTO 1710
1700 IF CHOICE# <> "b" GOTO 1730
1710 GOSUB 2980 ' CONTROL ALGORITHMS
      SELECTED.

```

```

1720 GOTO 1460 'BRANCH BACK TO MAIN
      MENU.
1730 IF CHOICE$ = "C" GOTO 1750
1740 IF CHOICE$ <> "c" GOTO 1770
1750 GOSUB 5410 'BEGIN EXECUTION
      SELECTED.
1760 GOTO 1440 'BRANCH TO MAIN MENU.
1770 IF CHOICE$ = "D" GOTO 1790
1780 IF CHOICE$ <> "d" GOTO 1850
1790 CLS 'SHUT DOWN SELECTED.
1800 LOCATE 5,5: INPUT "DO YOU WANT TO EXIT? Y/N. " ,S$
1810 IF (S$ = "N") OR (S$ = "n") GOTO 1440 'RETURN
      TO MAIN MENU.
1820 IF (S$ = "Y") OR (S$ = "y") GOTO 1890
1830 '
1840 'INVALID SELCTION HANDLER
1850 CLS
1860 LOCATE 23,1: .PRINT "INVALID SELECTION. TRY AGAIN."
1870 GOTO 1490 'BRANCH TO MAIN MENU.
1880 '
1890 'SEGMENT TO SET UP VARIABLES TO SHUT DOWN SYSTEM AND
      DRAIN PROCESSOR.
1900 SOURCE% = 0!
1910 OUTCHAN% = 0
1920 OPT$ = ""

```

```
1930 CALL ADUS(OUTCHAN%,SOURCE%,OPT$)
1940 IF DFLAG = 0 GOTO 2220
1950 ' CLOSE DATA FILES
1960 CLOSE #1
1970 CLOSE #2
1980 '
1990 ' QUERY FOR HARD COPY OF DATA FILES.
2000 LOCATE 8,5: PRINT "DO YOU WISH DATA FILES BE SENT TO
      PRINTER?"
2010 LOCATE 10,5: INPUT "ENTER 'Y' FOR YES AND 'N' FOR NO.
      ";S$
2020 IF S$ = "Y" OR S$ = "y" GOTO 2080
2030 IF S$ = "N" OR S$ = "n" GOTO 2220
2040 CLS
2050 LOCATE 16,5: PRINT "INVALID ENTRY. INPUT AGAIN."
2060 GOTO 2000
2070 'SEND DATA FILE "VALSAMP" TO PRINTER FOR HARD COPY.
2080 OPEN "B:VALSAMP.01" FOR INPUT AS #1
2090 INPUT #1, IHEISE,U(1),SETPT,X
2100 LPRINT "LEVEL = ";IHEISE, "SETPOINT = ";SETPT,
      "OUTPUT VOLTAGE = ";U(1), "SAMPLE NUMBER FOR
      PARAMETER SET = ";X
2110 IF EOF(1) THEN 2130
2120 GOTO 2090
2130 CLOSE #1
```

```
2140 '
2150 'SEND DATA FILE "VARVAL" TO PRINTER FOR HARD COPY.
2160 OPEN "B:VARVAL" FOR INPUT AS #2
2170 INPUT #2, KP,KI,KD,CONTROL$,TIME
2180 LPRINT "KP = ";KP, "KI = ";KI, "KD = ";KD, "CONTROL$
      = ";CONTROL$, "SAMPLING TIME = ";TIME
2190 IF EOF(2) THEN 2210
2200 GOTO 2170
2210 CLOSE #2
2220 END
2230 'SUBROUTINE FOR INPUTTING SETPOINT
2240 'THIS SUBROUTINE IS FOR INPUTTING THE SETPOINT. A MENU
      IS USED FOR
2250 'FOR CHOOSING EITHER A DEFAULT SETPOINT (7.5 INCHES) OR
      YOUR OWN.
2260 'THIS SUBROUTINE IS CALLED BY THE MAIN PROGRAM. IT
      CALLS NO OTHER
2270 'SUBROUTINE. THE SUBROUTINE RETURNS TO THE MAIN
      PROGRAM.
2280 'THE VARIABLES USED ARE AS FOLLOWS: SETPT - VARIABLE
      TO HOLD SETPOINT.
2290 'SETPT$ -VARIABLE TO HOLD SETPOINT AS A STRING TO BE
      USED IN GRAPHING.
2300 'SETPT1 -VARIABLE TO HOLD INPUT OF SETPOINT UNTIL IT IS
      PROVEN VALID.
```

```

2310 'CHD$ -VARIABLE TO HOLD YOUR MENU SELECTION.
2320 '
2330 'SET SCREEN FORMAT
2340   CLS
2350 '
2360 'PRINT MENU
2370   LOCATE 3,20: PRINT "SETPOINT MENU"
2380   LOCATE 5,20: PRINT "PLEASE MAKE SELECTION."
2390   LOCATE 7,20: PRINT "A. DEFAULT SETPOINT (7.5 INCHES)"
2400   LOCATE 9,20: PRINT "B. ENTER SETPOINT IN INCHES."
2410   LOCATE 11,20: PRINT "C. RETURN TO MAIN MENU."
2420   LOCATE 13,20: PRINT "SETPOINT ="; SETPT
2430   LOCATE 19,1: INPUT "ENTER MENU SELECTION. ", CHD$
2440 '
2450 'CHECK SELECTION.
2460   IF (CHD$ ="A") GOTO 2510
2470   IF CHD$ <> "a" GOTO 2590
2480 '
2490 'DEFAULT SELECTED.
2500 'DEFAULT HANDLER FOLLOWS.
2510   SETPT = 7.5           'SETPOINT = 7.5 IN.
2520   SETPT$ = STR$(SETPT) 'SET STRING VARIABLE FOR
    GRAPH TITLE.
2530   GSETPT = SETPT * 341.25 'SETPOINT IN GRAPH UNITS.
2540   VSETPT = 2.456         'SET VOLTAGE FOR SETPOINT.

```

```

2550 GFLAG = 1!
2560 GOTO 2340 'BRANCH TO SETPOINT MENU.
2570 '
2580 'CHECK SELECTION
2590 IF CHD$ ="B" GOTO 2630
2600 IF CHD$ <> "b" GOTO 2860
2610 '
2620 'SETPOINT HANDLER.
2630 LOCATE 19,1: INPUT "ENTER SETPOINT. (0 - 12.0 INCHES)
      SETPT = ", SETPT1$
2640 SETPT1 = VAL(SETPT1$)
2650 IF SETPT1 > 0 GOTO 2690
2660 IF SETPT1$ = "0" OR SETPT1$ = "0." OR SETPT1$ =
      "0.0" GOTO 2690
2670 IF SETPT1 = 0 GOTO 2800
2680 ' CHECK FOR INVALID SETPOINT.
2690 IF SETPT1<0! GOTO 2800
2700 IF SETPT1> 12! GOTO 2800
2710 'VALID SETPOINT HANDLER.
2720 SETPT = SETPT1 'SETPOINT IS SET.
2730 SETPT$ = STR$(SETPT) 'VARIABLE FOR TITLE IN GRAPH
2740 GSETPT = SETPT * 341.25 'VARIABLE FOR GRAPHING
      SETPOINT
2750 GFLAG = 0!

```



```
2760  GOSUB 6600          'CONVERT SETPOINT TO VOLTAGE
      UNITS.
2770  GOTO 2340          'BRANCH TO SETPOINT MENU.
2780  '
2790  'INVALID SETPOINT HANDLER.
2800  CLS
2810  LOCATE 5,1: PRINT "SETPOINT IS OUT OF RANGE. "
2820  LOCATE 7,1: PRINT "VALUE ENTERED WAS ";SETPT1$
2830  GOTO 2630          'BRANCH TO INPUT SETPOINT.
2840  '
2850  'SELECTION IS CHECKED.
2860  IF CHO$ ="C" GOTO 2940
2870  IF CHO$ = "c" GOTO 2940
2880  '
2890  'INVALID SELECTION HANDLER.
2900  CLS
2910  LOCATE 23,1: PRINT "INVALID MENU SELECTION. TRY
      AGAIN."
2920  GOTO 2350
2930  '
2940  RETURN            'BRANCH TO MAIN MENU.
2950  '
2960  '
2970  '
2980  'CONTROL ALGORITHM SUBROUTINE
```

2990 'THIS SETS THE PARAMETERS NEEDED TO COMPUTE THE  
CONTROL ALGORITHMS.

3000 'VARIABLES ARE AS FOLLOWS: KP- GAIN. USED FOR  
COMPUTING THE VOLTAGE OUT.

3010 'KP1 -USED TO HOLD KP INPUT UNTIL VALIDITY IS CHECKED.

3020 'KI -INTREGRAL CONSTANT. USED FOR COMPUTING VOLTAGE  
OUT.

3030 'KI1 - USED TO HOLD KI INPUT UNTIL VALIDITY IS CHECKED.

3040 'KD USED FOR COMPUTING VOLTAGE OUT.

3050 'KD1 -USED TO HOLD KD INPUT UNTIL VALIDITY IS CHECKED.

3060 'CONTROL# -USED TO DESIGNATE WHICH CONTROL ALGORITHM  
WAS CHOSEN.

3070 'CHOICE# -HOLDS THE SELECTION FROM THE CONTROL  
ALGORITHM MAIN MENU.

3080 'CHO# -HOLDS THE SELECTION FROM THE CONTROL ALGORITHM  
SUBMENUS.

3090 '

3100 'SET MENU SCREEN

3110 KEY OFF

3120 WIDTH 80

3130 CLS

3140 LOCATE 5,20: PRINT "CONTROL ALGORITHM MENU"

3150 LOCATE 7,20: PRINT "PLEASE SELECT ONE."

3160 LOCATE 9,20: PRINT "A. DEFAULT -- PROPORTIONAL  
CONTROL, P"

```
3170 LOCATE 11,20: PRINT "B. PROPORTIONAL INTEGRAL
      CONTROL, P-I"
3180 LOCATE 13,20: PRINT "C. PROPORTIONAL INTEGRAL
      DERIVATIVE CONTROL, P-I-D"
3190 LOCATE 15,20: PRINT "D. RETURN TO MAIN MENU."
3200 LOCATE 17,20: PRINT "CURRENT CONTROL ALGORITHM IS
      ";CONTROL$
3210 LOCATE 19,20: PRINT "KP = ";KP,"KI = ";KI,"KD =
      ";KD
3220 LOCATE 21,1: INPUT "ENTER SELECTION. ", CHOICE$
3230 '
3240 'CHECK SELECTION
3250 IF (CHOICE$ = "a") OR (CHOICE$ = "A") GOTO 3370 ' "P"
      CONTROL CHOSEN.
3260 IF (CHOICE$ = "b") OR (CHOICE$ = "B") GOTO 3930
      ' "P-I" CONTROL CHOSEN.
3270 IF (CHOICE$ = "c") OR (CHOICE$ = "C") GOTO 4590
      ' "P-I-D" CONTROL CHOSEN
3280 IF (CHOICE$ = "d") OR (CHOICE$ = "D") GOTO 3350
      'RETURN TO MAIN MENU.
3290 '
3300 'INVALID SELECTION HANDLER.
3310 CLS
3320 LOCATE 23,1: PRINT "INVALID SELECTION. TRY AGAIN."
3330 GOTO 3140 'BRANCH TO CONTROL ALGORITHM MENU.
```

```
3340 'RETURN TO MAIN MENU SELECTED.
3350   RETURN                               'RETURN TO MAIN MENU.
3360 '
3370 'PROPORTIONAL CONTROL SELECTED.
3380 '
3390 'SET INPUT MENU FOR PROPORTIONAL PARAMETER
      ALGORITHM.
3400   CLS
3410   LOCATE 5,20: PRINT "PROPORTIONAL PARAMETER MENU"
3420   LOCATE 7,20: PRINT "PLEASE MAKE SELECTION."
3430   LOCATE 9,20: PRINT "A. ENTER GAIN, KP > 0."
3440   LOCATE 11,20: PRINT "B. RETURN TO CONTROL ALGORITHM
      MENU."
3450   LOCATE 13,20: PRINT "C. RETURN TO MAIN MENU."
3460   LOCATE 15,20: PRINT "CURRENTLY KP ="; KP
3470 'SET CONTROL PARAMETERS TO PROPORTIONAL CONTROL VALUES.
3480   KI = 0!
3490   KI1$ = "0.0"
3500   KD = 0!
3510   KD1$ = "0.0"
3520   CONTROL$ = "P"
3530   LOCATE 19,1: INPUT "ENTER MENU SELECTION.  ", CHO$
3540 '
3550 'CHECK SUBMENU SELECTION.
3560   IF CHO$ = "A" GOTO 3600
```

```
3570  IF CH0$ <> "a" GOTO 3760
3580  '
3590  'INPUT KP AND CHECK FOR VALIDITY.
3600  LOCATE 21,1: INPUT "ENTER VALUE > 0.0 FOR KP.  KP =
      ", KP1$
3610  KP1 = VAL(KP1$)      'CONVERT TO NUMERIC VALUE
3620  IF KP1 > 0! GOTO 3710      'CHECK FOR VALID INPUT.
3630  '
3640  'INVALID INPUT HANDLER.
3650  CLS
3660  LOCATE 7,1: PRINT "INVALID INPUT. KP MUST BE
      GREATER THAN 0."
3670  LOCATE 9,1: PRINT "YOUR INPUT WAS KP ="; KP1$
3680  GOTO 3600
3690  '
3700  'VALID INPUT HANDLER.
3710  KP = KP1      'VALID INPUT.
3720  LOCATE 15,20: PRINT "CURRENTLY KP = ";KP
3730  GOTO 3400      'BRANCH TO "P" MENU.
3740  '
3750  'CHECK SELECTION.
3760  IF CH0$ = "B" GOTO 3800
3770  IF CH0$ <> "b" GOTO 3830
3780  '
3790  'BRANCH TO CONTROL ALGORITHM.
```

```
3800   GOTO 3130           'RETURN TO CONTROL
      ALGORITHM MENU.

3810 '
3820 'CHECK SELECTION.

3830   IF CHO# = "C" GOTO 3920 'CHECK FOR VALID SELECTION.
3840   IF CHO# = "c" GOTO 3920
3850 '
3860 'INVALID SELECTION HANDLER
3870   CLS
3880   LOCATE 23,1: PRINT "INVALID MENU SELECTION.
      TRY AGAIN."
3890   GOTO 3410           'BRANCH TO "P" MENU.
3900 '
3910 'VALID INPUT. RETURN TO MAIN MENU.
3920   RETURN             'RETURN TO MAIN MENU.
3930 'SET INPUT MENU FOR PROPORTIONAL INTEGRAL CONTROL.
3940   CLS
3950   LOCATE 5,20: PRINT "PROPORTIONAL INTEGRAL
      PARAMETER MENU"
3960   LOCATE 7,20: PRINT "A. ENTER GAIN AND INTEGRAL
      CONSTANT."
3970   LOCATE 9,20: PRINT "B. RETURN TO CONTROL ALGORITHM
      MENU."
3980   LOCATE 11,20: PRINT "C. RETURN TO MAIN MENU."
3990   LOCATE 13,20: PRINT "CURRENTLY KP ="; KP, "KI =";KI
```

```
4000 'SET CONTROL PARAMETERS TO P-I CONTROL VALUES.
4010   KD = 0!
4020   KD1$ = "0.0"
4030   CONTROL$ = "P-I"
4040   LOCATE 19,1: INPUT "ENTER MENU SELECTION. ",CHO$
4050 '
4060 'CHECK SUBMENU SELECTION.
4070   IF (CHO$ = "A") GOTO 4110
4080   IF CHO$ <> "a" GOTO 4410
4090 '
4100 'KI AND KP INPUT HANDLER.
4110   LOCATE 15,1: INPUT "ENTER GAIN (KP > 0).  KP = ",
      KP1$
4120   KP1 = VAL(KP1$)
4130   IF KP1 > 0! GOTO 4220           'CHECK FOR VALID INPUT.
4140 '
4150 'INVALID KP HANDLER.
4160   CLS
4170   LOCATE 5,1: PRINT "INVALID INPUT. KP MUST BE
      GREATER THAN 0. "
4180   LOCATE 7,1: PRINT "YOUR INPUT FOR KP = ";KP1$
4190   GOTO 4110           'BRANCH TO INPUT KP.
4200 '
4210 'VALID KP INPUT HANDLER.
4220   KP = KP1
```

```
4230 '
4240 'RESET SCREEN TO SHOW VALID KP INPUT AND INPUT KI.
4250 LOCATE 13,20: PRINT "CURRENTLY KP = "; KP,
      "KI = "; KI
4260 LOCATE 17,1: INPUT "ENTER INTEGRAL CONSTANT
      (KI > 0.0). KI = ",KI1$
4270 KI1 = VAL(KI1$)
4280 IF KI1 > 0! GOTO 4370 'CHECK FOR VALID INPUT.
4290 '
4300 'INVALID KI HANDLER.
4310 CLS
4320 LOCATE 5,1: PRINT "INVALID INPUT. KI MUST BE
      GREATER THAN 0."
4330 LOCATE 7,1: PRINT "YOUR INPUT FOR KI = ";KI1$
4340 GOTO 4260 'BRANCH TO INPUT KI
4350 '
4360 'VALID KI INPUT HANDLER.
4370 KI = KI1
4380 GOTO 3940 'BRANCH TO "P-I" MENU.
4390 '
4400 'CHECK SUBMENU SELECTION.
4410 IF CHO$ = "B" GOTO 4430
4420 IF CHO$ <> "b" GOTO 4460
4430 GOTO 3130 'BRANCH TO CONTROL ALGORITHM MENU.
4440 '
```



```
4450 'CHECK SUBMENU SELECTION.
4460   IF CHO$ = "C" GOTO 4550
4470   IF CHO$ = "c" GOTO 4550
4480 '
4490 'INVALID SUBMENU SELECTION HANDLER.
4500   CLS
4510   LOCATE 23,1: PRINT "INVALID SELECTION. TRY AGAIN."
4520   GOTO 3950           'BRANCH TO CONTROL
                          ALGORITHM MENU.
4530 '
4540 'RETURN TO MAIN MENU SELECTED.
4550   RETURN
4560 '
4570 '
4580 '
4590 'SET MENU FOR PROPORTIONAL INTEGRAL DERIVATIVE CONTROL.
4600   CLS
4610   LOCATE 5,20: PRINT "PROPORTIONAL INTEGRAL
                          DERIVATIVE CONTROL MENU"
4620   LOCATE 7,20: PRINT "A. ENTER PARAMETERS KP, KI, AND
                          KD. "
4630   LOCATE 9,20: PRINT "B. RETURN TO CONTROL ALGORITHM
                          MENU."
4640   LOCATE 11,20: PRINT "C. RETURN TO MAIN MENU."
```

```
4650 LOCATE 13,20: PRINT "CURRENTLY KP = "; KP, "KI = ";
      KI, "KD = "; KD
4660 CONTROL$ = "P-I-D"
4670 '
4680 'INPUT SUBMENU SELECTION.
4690 LOCATE 21,1: INPUT "ENTER MENU SELECTION. ", CHO$
4700 '
4710 'CHECK SUBMENU SELECTION.
4720 IF CHO$ = "A" GOTO 4760
4730 IF CHO$ <> "a" GOTO 5240
4740 '
4750 'KP INPUT HANDLER.
4760 LOCATE 15,1: INPUT "ENTER GAIN. (KP > 0) KP = ",
      KP1$
4770 KP1 = VAL(KP1$)
4780 IF KP1 > 0! GOTO 4880 'CHECK FOR VALID INPUT.
4790 '
4800 'INVALID KP HANDLER.
4810 CLS
4820 LOCATE 5,20: PRINT "INVALID INPUT. KP MUST BE
      GREATER THAN 0."
4830 LOCATE 7,20: PRINT "INPUT WAS KP = "; KP1$
4840 LOCATE 13,20: PRINT "CURRENTLY KP = "; KP, "KI = ";
      KI, "KD = "; KD
4850 GOTO 4760 'BRANCH TO INPUT KP.
```

```
4860 '
4870 'VALID KP INPUT HANDLER.
4880   KP = KP1
4890 'RESET VALUES ON SCREEN.
4900   LOCATE 13,20: PRINT "CURRENTLY KP = ";KP, "KI = "; KI ,
      "KD = "; KD
4910 '
4920 'KI INPUT HANDLER.
4930   LOCATE 17,1: INPUT "ENTER VALUE FOR KI > 0.
      KI = ",KI1$
4940   KI1 = VAL(KI1$)
4950   IF KI1 > 0! GOTO 5040      'CHECK FOR VALID INPUT.
4960 '
4970 'INVALID KI INPUT HANDLER.
4980   CLS
4990   LOCATE 5,20: PRINT "INVALID INPUT. KI MUST BE
      POSITIVE."
5000   LOCATE 7,20: PRINT "INPUT WAS KI = "; KI1$
5010   GOTO 4900   'BRANCH TO INPUT KI.
5020 '
5030 'VALID KI INPUT HANDLER.
5040   KI = KI1
5050 'RESET SCREEN.
5060   LOCATE 13,20: PRINT "CURRENTLY KP = ";KP, "KI = ";
      KI, "KD = ";KD
```

```
5070 LOCATE 19,1: INPUT "ENTER VALUE FOR KD > 0.  
      KD = ",KD1$  
5080 KD1 = VAL(KD1$)  
5090 '  
5100 'KD INPUT HANDLER.  
5110 IF KD1 > 0! GOTO 5200 'CHECK FOR VALID INPUT.  
5120 '  
5130 'INVALID KD INPUT HANDLER.  
5140 CLS  
5150 LOCATE 5,20: PRINT "INVALID INPUT. KD MUST BE  
      POSITIVE."  
5160 LOCATE 7,20: PRINT "INPUT WAS KD = "; KD1$  
5170 GOTO 5060 'BRANCH TO INPUT KD.  
5180 '  
5190 ' VALID INPUT HANDLER  
5200 KD = KD1  
5210 GOTO 4600 'BRANCH TO "P-I-D" MENU.  
5220 '  
5230 CHECK SUBMENU SELECTION.  
5240 IF CHD$ = "B" GOTO 5280  
5250 IF CHD$ <> "b" GOTO 5310  
5260 '  
5270 'BRANCH TO MAIN CONTROL ALGORITHM MENU SELECTED.  
5280 GOTO 3130 'BRANCH TO CONTROL  
      ALGORITHM MENU.
```

```
5290 '
5300 'CHECK SUBMENU SELECTION.
5310   IF CHO# = "C" GOTO 5400
5320   IF CHO# = "c" GOTO 5400
5330 '
5340 'INVALID SUBMENU SELECTION HANLDLER.
5350   CLS
5360   LOCATE 23,1: PRINT "INVALID SELECTION. TRY AGAIN."
5370   GOTO 4610                               'BRANCH TO "P-I-D" MENU.
5380 '
5390 'RETURN TO MAIN MENU SELECTED.
5400   RETURN                               'RETURN TO MAIN MENU.
5410 'EXECUTION SUBROUTINE
5420 'EXECUTION SUBROUTINE IS CALLED BY THE MAIN MENU.
5430 'THIS SUBROUTINE CALLS INIT.GRAF, NEXT POINT.GRAF,
      CALCULATE VOLTAGE,
5440 'OUTPUT VOLTAGE AND SETSTAT.  IT RETURNS TO THE MAIN
      MENU.
5450 'VARIABLES ARE AS FOLLOWS: KI-GAIN, KP-INTEGRAL
      CONSTANT, KD
5460 'SETPT - CURRENT SETPOINT, SETPT# - USED TO PLACE
      SETPOINT IN TITLE OF
5470 'OF GRAPH, TIME -SAMPLING RATE OF CHECKING THE SYSTEM,
      CONTROL# - TYPE
```

5480 'OF CONTROL ALGORITHM IN USE, CHO# - HOLDS "Y" OR "N"  
TO RETURN TO MAIN

5490 'MENU, E(2) - ARRAY TO HOLD PRESENT AND PREVIOUS VALUES  
OF THE DIFFERENCE

5500 'BETWEEN THE SETPOINT AND THE CURRENT LEVEL, U(1) -  
ARRAY TO HOLD PRESENT

5510 'AND PREVIOUS VALUE OF VOLTAGE TO VALVE OF PROCESSOR,  
GFLAG - FLAG TO

5520 'ASCERTAIN WHETHER SETPOINT OR CURRENT LEVEL NEEDS TO  
BE CHANGED TO

5530 'GRAPH UNITS OR VOLTAGE (GFLAG = 0 --SETPT,VSETPT AND  
GSETPT ARE SET,

5540 'GFLAG = 1 ---GHEISE IS SET),

5550 'STAT - HOLDS ERROR NUMBER IF ONE OCCURS.

5560 'A,B AND C ARE USED TO CALCULATE THE VOLTAGE SENT TO  
THE VALVE OF THE

5570 'PROCESSOR AND ARE CALCULATED AHEAD OF HERE TO SAVE  
TIME IN THE OTHER

5580 'SUBROUTINE.

5590 '

5600 'THE FIRST BLOCK IS A SCREEN TO SHOW THE

5610 'CURRENT VALUES FOR VARIABLES KP,KI,KD,

5620 'SETPOINT, TIME AND CONTROL.

5630 '

5640 CLS

```
5650  
5660 LOCATE 9,5: PRINT "SAMPLING TIME FOR THE SYSTEM MUST  
    BE GREATER THAN OR EQUAL TO ONE SECOND"  
5670 LOCATE 11,5: INPUT"ENTER SAMPLING TIME.  
    TIME = ",TM$  
5680 TIME = VAL(TM$)  
5690 TIME = INT(TIME)  
5700 IF TIME < 1! GOTO 5720  
5710 IF TIME <= 86400! GOTO 5750  
5720 CLS  
5730 LOCATE 7,5: PRINT "SAMPLING RATE MUST BE AN INTEGER  
    GREATER THAN OR EQUAL TO ONE SECOND."  
5740 GOTO 5660  
5750 CLS  
5760 LOCATE 5,5: PRINT "CURRENT VALUES ARE"  
5770 LOCATE 7,5: PRINT "KP = "; KP; "KI = "; KI; "KD = ";  
    KD  
5780 LOCATE 9,5: PRINT "SETPOINT = "; SETPT  
5790 LOCATE 11,5: PRINT "SAMPLING TIME = "; TIME  
5800 LOCATE 13,5: PRINT "CONTROL ALGORITHM = "; CONTROL$  
5810  
5820 'GIVE CHOICE TO CONTINUE OR RETURN TO MAIN MENU.  
  
5830 LOCATE 17,5: PRINT "PRESS 'Y' AND ENTER KEY TO RETURN  
    TO MAIN MENU."
```

```
5840 LOCATE 19,5: PRINT "PRESS 'N' AND ENTER KEY TO
      BEGIN/CONTINUE EXECUTION OF THE SYSTEM."
5850 LOCATE 21,5: INPUT "ENTER CHOICE. (Y/N) ",CHO$
5860 IF CHO$ = "Y" OR CHO$ = "y" THEN RETURN
5870 CLS
5880 'CHECK DATA FLAG
5890 IF DFLAG = 0 GOTO 5950
5900 '
5910 'WRITE THE VALUES OF KP,KI,KD,CONTROL$ AND TIME TO
      DISK FILE NAMED
5920 ' "VARVAL" IN DRIVE "B".
5930 WRITE #2, KP,KI,KD,CONTROL$,TIME
5940 '
5950 'INITIALIZE X -NUMBER OF SAMPLES TAKEN FOR THIS
      PARAMETER SET.
5960 X = 1
5970 '
5980 'SET UP VARIABLES FOR GRAF SUB
5990 PALETTE%=1
6000 BACKGROUND% = 1
6010 TYPE% = 3
6020 LOVAL = 0
6030 HIVAL = 4095
6040 FORMAT%(0) =1 'ACTUAL LEVEL IS CYAN
6050 FORMAT%(1) =2 'SETPOINT IS MAGENTA
```



```
6060  FORMAT%(2) =3  'VOLT TO ATTAIN SETPOINT IS WHITE
6070  FORMAT%(3) =-1 'NO MORE PLOTS
6080  TITLE$= "SETPT=" +SETPT$+ " CRL ALG " +CONTROL$
6090  YLABEL$="12      0 12      0"
6100  XLABEL$="T= " +TM$+ " KP="+KP1$+" KI="+KI1$+"
      KD="+KD1$
6110  COMMENT$ = "LEVEL-BLUE,SETPT-RED,VOLTAGEOUT-WHITE
      PRESS ANY KEY TO RETURN TO MAIN MENU"
6120  GOSUB 51000
6130  '
6140  'SET VARIABLES FOR MULTIPLE ANALOG INPUT
6150  CHAN% = 0      'CHANNEL 0 TAKES IN VOLTS.
6160  NUMSAMP = 10   'TEN SAMPLES TO BE TAKEN ON EACH
      SAMPLE
6170  RT = 8000     '8000 PER SECOND SAMPLING RATE
6180  OPT$ = ""
6190  '
6200  'CHECK FOR SYSTEM ERRORS.
6210  STAT%=0       'SET UP SETSTAT PARAMETER
6220  CALL SETSTAT(STAT%) 'SET UP STATUS
6230  IF STAT% <> 0 THEN PRINT "STATUS = ";STAT%:END
6240  '
6250  'SET UP SINGLE ANALOG OUTPUT FOR VOLTAGE TO
      ISOVERTER.
```

```
6260  OUTCHAN% = 0      'SET UP VARIABLE FOR SINGLE ANALOG
      OUTPUT.
6270  '
6280  'SET UP SINGLE ANALOG OUTPUT FOR VOLTAGE CORRESPONDING
      TO SETPOINT.
6290  SETCHAN% = 1
6300  '
6310  'SET UP SINGLE ANALOG OUTPUT FOR VOLTAGE CORRESPONDING
      TO LEVEL.
6320  LEVCHAN% = 2
6330  '
6340  'CALCULATE VARIABLES USED IN DETERMINING VOLTAGE OUT.
6350  A = KP + KI *(TIME/2) + KD/TIME      ' SET VARIABLE FOR
      CONTROL ALGORITHM
6360  B = KI * (TIME/2) - KP - 2 * (KD/TIME) 'SET VARIABLE
      FOR CONTROL ALG.
6370  C = KD/TIME      'SET VARIABLE FOR CONTROL ALG.
6380  '
6390  'CALL SUBROUTINE TO CONVERT SETPOINT INTO VOLTS AND SET
      GSETPT.
6400  IF GFLAG <> 0! GOTO 6430
6410  GOSUB 6600      'GO TO TABLE TO CONVERT
      SETPOINT INTO VOLTS.
6420  'INITIATE GRAPH SUBROUTINE.
6430  '

```

```
6440 'INITIALIZE A# FOR WHILE LOOP.
6450 'WHILE LOOP HAS A TRAP CALL EVERY "TIME" SECONDS.
6460 'ON EACH TRAP THE SYSTEM IS CHECKED AND VALVE TO
        PROCESSOR IS ADJUSTED.
6470 'THE REMARK LOOP PASSES THE UNUSED TIME FOR THE TRAP
        CALL.
6480  A# = ""
6490  R = TIME
6500  D = 1      'COUNTER FOR TIME
6510  WHILE A# = ""
6520      ON TIMER(1) GOSUB 7360
6530      TIMER ON
6540      G = 1!
6550      WHILE G <> 0!
6560          REM LOOP TO WAIT FOR SAMPLING TIME TO ELAPSE
6570          WEND
6580  WEND
6590  RETURN
6600 'CONVERSION SUBROUTINE
6610 'CALLED BY EXECUTION SETPOINT AND TRAP SUBROUTINES, AND
        RETURNS THERE.
6620 'VARIABLES ARE AS FOLLOWS: V ARRAY TO HOLD THE
        CORRESPONDING VOLTAGES
6630 'TO THE LEVEL IN THE PROCESSOR, SETPT - THE LEVEL TO BE
        ATTAINED,
```

6640 'GSETPT - THE SETPT IN GRAPH UNITS, VSETPT - THE SETPT  
IN VOLTS,  
6650 'VHEISE - VOLTAGE COMING FROM THE PROCESSOR, GHEISE -  
VHEISE  
6660 'CONVERTED TO GRAPH UNITS.  
6670 'GFLAG - FLAG TO DECIDE IF SETPOINT VARIABLES HAVE BEEN  
SET  
6680 'OR IF GRAPH UNITS FOR THE ACTUAL LEVEL NEED TO BE SET.  
6690 'GFLAG = 0 --SETPOINT VARIABLES NEED TO BE SET.  
GFLAG = 1 -- GHEISE  
6700 'NEEDS TO BE SET.  
6710 '  
6720 'READ IN VOLTAGE VALUES IN ARRAY V.  
6730 '0 TO 2 INCHES  
6740 V(0) = 0!  
6750 V(1) = .034  
6760 V(2) = .117  
6770 V(3) = .22  
6780 V(4) = .33  
6790 '2 1/2 TO 4 INCHES  
6800 V(5) = .447  
6810 V(6) = .574  
6820 V(7) = .705  
6830 V(8) = .857  
6840 '4 1/2 TO 6 INCHES

6850 V(9) = 1.018  
6860 V(10) = 1.194  
6870 V(11) = 1.389  
6880 V(12) = 1.6214  
6890 '6 1/2 TO 8 INCHES  
6900 V(13) = 1.858  
6910 V(14) = 2.129  
6920 V(15) = 2.456  
6930 V(16) = 2.83  
6940 '8 1/2 TO 10 INCHES  
6950 V(17) = 3.25  
6960 V(18) = 3.735  
6970 V(19) = 4.297  
6980 V(20) = 5.032  
6990 '10 1/2 TO 12 INCHES  
7000 V(21) = 5.903  
7010 V(22) = 7.009  
7020 V(23) = 8.559  
7030 V(24) = 9.902001  
7040 '  
7050 'FIND SUBSCRIPT OF LOWER INTERVAL VALUE, I  
7060 'SETPT IS THE SETPOINT.  
7070 'DIVIDE SETPT BY 0.5 TO FIND THE LOWER  
7080 'SUBSCRIPT OF THE INTERVAL. THIS WAY  
7090 'A LOOP IS NOT REQUIRED TO FIND WHICH INTERVAL

```
7100 'THE SETPOINT FALLS IN. THE RESULT IS
7110 'THE SUBSCRIPT NEEDED.
7120 '
7130 'CHECK GFLAG.
7140 IF GFLAG <> 0! GOTO 7280 'CHECK FOR WHICH VARIABLE
    IS BEING CALCULATED.
7150 '
7160 'CALCULATE THE LOWER INTERVAL SUBSCRIPT.
7170 I = INT(SETPT/.5) 'FIND LOWER SUBSCRIPT OF
    INTERVAL
7180 '
7190 'INTERPOLATION CHECK
7200 Q = SETPT - (I*.5) 'CHECK IF INTERPOLATION IS
    NECESSARY.
7210 IF Q = 0! THEN VSETPT= V(I) ELSE
    VSETPT = (((V(I+1)-V(I))*Q)/.5)+V(I)
7220 '
7230 'SET GFLAG AND RETURN.
7240 GFLAG = 1 'SET FLAG TO INDICATE VSETPT IS
    SET.
7250 RETURN
7260 '
7270 'CONVERT VHEISE TO GRAPH UNITS.
7280 FOR J = 0 TO 24 'LOOP TO FIND INTERVAL VHEISE
    IS CONTAINED.
```

```
7290     IF V(J) > VHEISE GOTO 7320
7300     NEXT J
7310     'CONVERT HEISE VOLTAGE TO CORRESPONDING VALUE IN INCHES
        FOR DATA STORAGE.
7320     IHEISE = ((VHEISE - V(J-1))/(V(J)-V(J-1))) * .5 +
        (J-1)/2!
7330     'CONVERT HEISE VOLTAGE TO GRAPH UNITS.

7340     GHEISE = (((VHEISE - V(J-1))/(V(J)-V(J-1))) * .5 +
        (J-1)/2) * 341.3
7350     RETURN
7360     'TRAP SUBROUTINE TO INPUT VOLTS FROM HIESE, CALCULATE
        OUTPUT
7370     'VOLTS NEEDED TO BRING THE SYSTEM UP TO SETPOINT LEVEL.
7380     'THIS SUBROUTINE IS CALLED FROM THE EXECUTION
        SUBROUTINE AND RETURNS
7390     'THERE. IT IS ENTERED VIA A TIMER CALL. THE TIME
        INTERVAL IS SET
7400     'BY THE PROGRAM USER.
7410     'THIS SUBROUTINE CALLS FOR MULTIPLE INPUTS FROM CHANNEL
        0 AND SINGLE
7420     'OUTPUTS FROM CHANNEL 0. GRAPH SUBROUTINES AND THE
        SUBROUTINE TO
7430     'CALCULATE THE VOLTAGE INTO GRAPH UNITS ARE CALLED.
```

7440 'VARIABLES ARE AS FOLLOWS: ST% -HOLD SYSTEM ERROR  
NUMBER, G - SET

7450 'TO 0. TO EXIT REMARK LOOP IN CALLING SUBROUTINE, SUM  
-USED TO

7460 'AVERAGE HEISE INPUTS TO INSURE AGAINST FALSE READINGS,  
RAW% - INPUTS

7470 'FROM THE HEISE, AVG - AVERAGE OF HEISE INPUTS, VHEISE  
- VOLTAGE FROM

7480 'THE HEISE, E ARRAY HOLDS PRESENT AND PREVIOUS  
DIFFERENCES OF THE

7490 'VOLTAGE CORRESPONDING TO THE SETPOINT AND THE VOLTAGE  
CORRESPONDING TO

7500 'THE ACTUAL LEVEL, U ARRAY - HOLDS PAST VOLTAGE OUTPUT  
AND CURRENT

7510 'VOLTAGE OUTPUT, GHEISE -GRAPH UNIT OF VHEISE, GSETPT -  
GRAPH UNIT OF

7520 'SETPT, VALUE - PASSING VARIABLE TO THE GRAPH  
SUBROUTINE, A# -USED

7530 'TO CHECK IF RETURN TO MAIN MENU IS REQUESTED.

7540 '

7550 'SET G TO EXIT REMARK LOOP

7560 G = 0!

7570 '

7580 'CALL FOR MULTIPLE INPUTS FROM A CHANNEL



```

7590 CALL AINM (CHAN%,NUMSAMP,RT,RAW%(0),OPT%)
7600 '
7610 'CHECK FOR SYSTEM ERROR
7620 IF ST% <> 0 THEN PRINT "EXECUTION ERROR # ";ST%:END
7630 SUM = 0!
7640 '
7650 'SUM INPUTS FROM CHANNEL 0
7660 FOR I = 0 TO 9
7670     SUM = SUM + RAW%(I)
7680 NEXT I
7690 '
7700 'AVERAGE TEN INPUTS FROM CHANNEL 0.
7710 AVG = SUM/10     'AVERAGE INPUTS
7720 VHEISE = AVG/409.6 'CONVERT ANALOG VALUE TO VOLTS
7730 GOSUB 6600     'CONVERT TO INCH AND GRAPH UNITS.
7740 IF D <> R GOTO 7900
7750 '
7760 'CALCULATE E(2) AND U(1).
7770 E(2) = VSETPT - VHEISE     'CALCULATE E(2)
7780 U(1) = U(0) + A*E(2) + B*E(1) + C*E(0)     'CALCULATE
    THE VOLTAGE OUT.
7790 IF U(1) < 0! THEN U(1)=0!     'CHECK IF NEGATIVE
7800 IF U(1) > 10! THEN U(1) = 9.998     'CHECK IF GREATER
    THAN 10 V.
7810 '

```

```
7820 'CONVERT VOLTAGE TO RANGE OF 0 TO 4095 TO OUTPUT IT.
7830 SOURCE% = U(1) * 409.6      'CONVERT TO RANGE
      0 - 4095
7840 '
7850 'UPDATE ARRAYS FOR NEXT SUBROUTINE CALL.
7860 U(0) = U(1)      'SET VARIABLES FOR NEXT LOOP
      THROUGH.
7870 E(0) = E(1)
7880 E(1) = E(2)
7890 IF DFLAG = 0 GOTO 7920
7900 'SAVE DATA ON DISK IN FILE NAME "VALSAMP.01" IN "B"
      DRIVE.
7910 PRINT #1,USING"#####.###"; IHEISE, U(1),SETPT,X
7920 '
7930 'INCREASE COUNTER OF SAMPLES TAKEN FOR THIS PARAMETER
      SET.
7940 X = X + 1
7950 '
7960 IF D <> R GOTO 8000
7970 'OUTPUT VOLTAGE ON CHANNEL 0.
7980 CALL ADUS(OUTCHAN%,SOURCE%,OPT#)      'OUTPUT VOLTAGE
7990 D = 0      'RESET COUNTER
8000 D = D+1    'INCREMENT COUNTER
8010 'GRAPH THE THREE VARIABLES.
8020 VALUE = GHEISE      'GRAPH ACTUAL LEVEL
```

```
8030  GOSUB 54000
8040  VALUE = GSETPT      'GRAPH SETPOINT
8050  GOSUB 54000
8060  VALUE = U(1) * 409.6  'GRAPH VOLTAGE OUT TO FOXBORD.
8070  GOSUB 54000
8080  '
8090  'OUTPUT VOLTS OF SETPOINT AND LEVEL TO PLOTTER.
8100  CALL ADUS (SETCHAN%,GSETPT,OPT#)
8110  CALL ADUS (LEVCHAN%,GHEISE,OPT#)
8120  '
8130  'CHECK FOR KEYBOARD INPUT TO RETURN TO MAIN MENU.
8140  A# = INKEY#
8150  IF A# = "" GOTO 8180
8160  TIMER OFF
8170  TIMER STOP
8180  RETURN
```

## LIST OF REFERENCES

- Anand, D.K. 1974. Introduction to Control Systems.  
New York: Pergamon Press, Inc.
- Clark, Andrew. 1977. Automatic and Process Control  
Technology Experiments. Pompano Beach, Florida:  
Technovate, Inc.
- Claudio, Pedro. 1986. "Direct Digital Control of a Liquid  
Level/Temperature Control Unit." Master's Thesis,  
University of Central Florida, 1986.
- Hale, Francis. 1988. Introduction to Control System  
Analysis and Design. Englewood Cliffs, N.J.: Prentice  
Hall.
- Jacquot, R. 1981. Modern Digital Control Systems. New  
York: Marcel Dekker.
- Klee, Harold. August, 1985. "Mathematical Modeling of  
Continuous Dynamic Systems." Paper presented at the  
International Conference on Mathematical Modeling,  
Berkeley, Calif.
- Klee, Harold. 1988. "Design of a Digital Control System by  
Simulation." Unpublished manuscript.
- Wong, Wing Tsan. Oct.-Dec., 1987. "Computer Control of the  
Liquid Level in a Tank by a Microprocessor." Computers  
in Education, Division of ASEE. Vol. VII, No. 4:  
pp. 40-44.