To the Graduate Council:

I am submitting herewith a thesis written by Grady R. Wetherington, Jr., entitled "Development of a Digitally Based Filter for Noise Reduction in Purged Dip-Tube Liquid Density Measurements." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

James C. Hung

James C. Hung, Major Professor

We have read this thesis and recommend its acceptance:

J M Googe

Ernest L Hall

Accepted for the Council:

C W Minkel

The Graduate School

DEVELOPMENT OF A DIGITALLY BASED FILTER FOR NOISE REDUCTION IN PURGED DIP-TUBE LIQUID DENSITY MEASUREMENTS

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Grady R. Wetherington, Jr.

August 1983

ACKNOWLEDGEMENTS

The author wishes to thank the members of his Graduate Committee: Drs. J. M. Googe, E. L. Hall, and J. C. Hung. Special thanks go to Dr. Hung for his assistance and guidance in the preparation of this thesis.

The author wishes to express his appreciation to the Instrumentation and Controls Division of the Oak Ridge National Laboratory, operated by the Nuclear Division of Union Carbide Corporation for the United States Department of Energy, for supplying the support necessary for the completion of this effort.

Appreciation is also expressed to Mr. W. A. Walker for his part in fabricating the density probe.

Special thanks go to Mrs. B. W. Brummitt, L. D. Canning, G. F. Grubbs, P. B. Hrushanyk, L. E. Klobe, and W. M. Murrill for their assistance in preparing this document.

Final thanks must go to the author's wife, Sarah, for her patience and support.

ABSTRACT

A digitally based noise reduction filter was developed to reduce the noise associated with a purged dip-tube density measurement. An experimental density probe and measurement system was constructed to observe the measurement process and develop the corresponding filter design. The resulting filter provides adequate noise attenuation while passing the desired density information. Filter implementation utilizes existing measurement system capacity without the addition of extra signal-conditioning hardware.

TABLE OF CONTENTS

CHAPTER Page

I. INTRODUCTION 1

II. DESCRIPTION OF THE MEASUREMENT SYSTEM 5

System Description 5
Measurement Mechanism 12
Analysis of Pressure Line Characteristics 17
Transmission line analogy 20
Lumped-parameter method 23

III. DIGITAL DATA PROCESSING 26

Method Selection 26
Filter Design 27

IV. FILTER IMPLEMENTATION AND SYSTEM PERFORMANCE 36

Algorithm Implementation 36
Performance Testing 38

V. SUMMARY 44

LIST OF REFERENCES 46

APPENDIXES 48

APPENDIX A 49
APPENDIX B 65
APPENDIX C 67

VITA 72

LIST OF TABLES

TABLE PAGE

1. Per-unit electrical equivalent values 21

2. Pneumatic line per-unit-length characteristic values 21

3. Lumped-parameter pneumatic line characteristic values 25

4. First-order step invariant filter parameters and coefficients 29

5. First-order bilinear filter parameters and coefficients 29

6. Second-order step invariant filter parameters and coefficients 30

7. Second-order bilinear filter parameters and coefficients 30

8. Real-time filter implementation 43

LIST OF FIGURES

FIGURE PAGE

1. Conceptual block diagram of a digital filter in a liquid density measurement device 3

2. Dual dip-tube liquid density probe 6

3. Pneumatic system block diagram 8

4. Cross-section of the Validyne DP-15 differential pressure sensor 9

5. Block diagram of the data acquisition system for liquid density measurement 11

6. Conceptual diagram of the dip-tube liquid level measurement technique 13

7. Conceptual diagram of the dual dip-tube liquid density measurement technique 16

8. Typical unfiltered density data 18

9. Diagram of a lumped-parameter equivalent model 24

10. First-order step invariant filter step response 31

11. First-order bilinear filter step response 32

12. Second-order step invariant filter step response 33

13. Second-order bilinear filter step response 34

14. Signal flow graph of noise reduction filter 37

15. Filtered liquid density measurement data 39

16. Unfiltered liquid density measurement data 40

17. Unfiltered density data with idealized step 41

18. Filter response to unfiltered density data with idealized step 42

A-1. Second-order filter coefficient generation program block diagram 50

FIGURE PAGE

A-2. Listing of second-order filter coefficient generation program 51

A-3. Block diagram of first-order filter coefficient generation program 57

A-4. Listing of first-order filter coefficient generation program 58

A-5. Listings of plotting subroutines 64

B-1. Listing of noise reduction filter subroutine 66

C-1. Block diagram of real-time filter implementation program 68

C-2. Listing of real-time filter implementation program 69

C-3. Real-time filter implementation program output 71

CHAPTER I

INTRODUCTION

This paper describes the development of a digitally based noise reduction filter for a purged dip-tube liquid density measurement. The project originated as a by-product of work associated with chemical process development at the Oak Ridge National Laboratory, which is operated by Union Carbide Corporation for the United States Department of Energy.

The purged dip-tube density measurement technique, which will be described in Chapter II, is frequently used in industrial and developmental chemical processes where automated on-line measurement of liquid density is required. It allows the determination of liquid density without exposing the sensing element, a differential pressure sensor, to the process. A disadvantage of this technique, however, is that it introduces observable noise with the density information. Older monitoring methods, such as analog chart recorders, were usually not sensitive to this error. This was due to the slow response characteristics associated with these devices. Newer, digitally based sampling methods are capable of data sample rates that accurately resolve this noise component. The problem is especially severe, however, when the data acquisition system samples the density signal at very high rates. When these systems also perform process control functions, a data processing rate mismatch can exist between the sampling rate of the process data and the execution rate of the control algorithm. Specifically, the sampling rate of the incoming data exceeds the execution, or processing, rate of the control algorithm. A data-aliasing phenomenon results, and digital filtering can be applied to alleviate the problem.

This effort to develop a digital filtering solution was initiated to reduce measurement-induced noise associated with the purged dip-tube density measurement technique. A pictorial representation of this concept is shown in Figure 1. In this example, the process density signal consists of a combination of liquid density information and a noise component. This signal is sampled by a microcomputer-based data acquisition system. The resulting binary representation of the process signal is subsequently digitally filtered within the microcomputer. The filter output is then made available to other measurement system tasks.

To develop the digital filter, a laboratory measurement system and density probe were constructed. A model of the purged dip-tube technique was developed to predict its operational charcteristics. After the density probe model was developed, a filtering technique was selected. Filter coefficients were then derived using a coefficient calculation program. The resulting filter was implemented in the microcomputer, in subroutine form, and the entire measurement system was operated to verify overall performance with the filter algorithm present.

The organization of the following chapters generally follows the development sequence of the digital filter. Chapter II presents a detailed description of the experimental density measurement system,

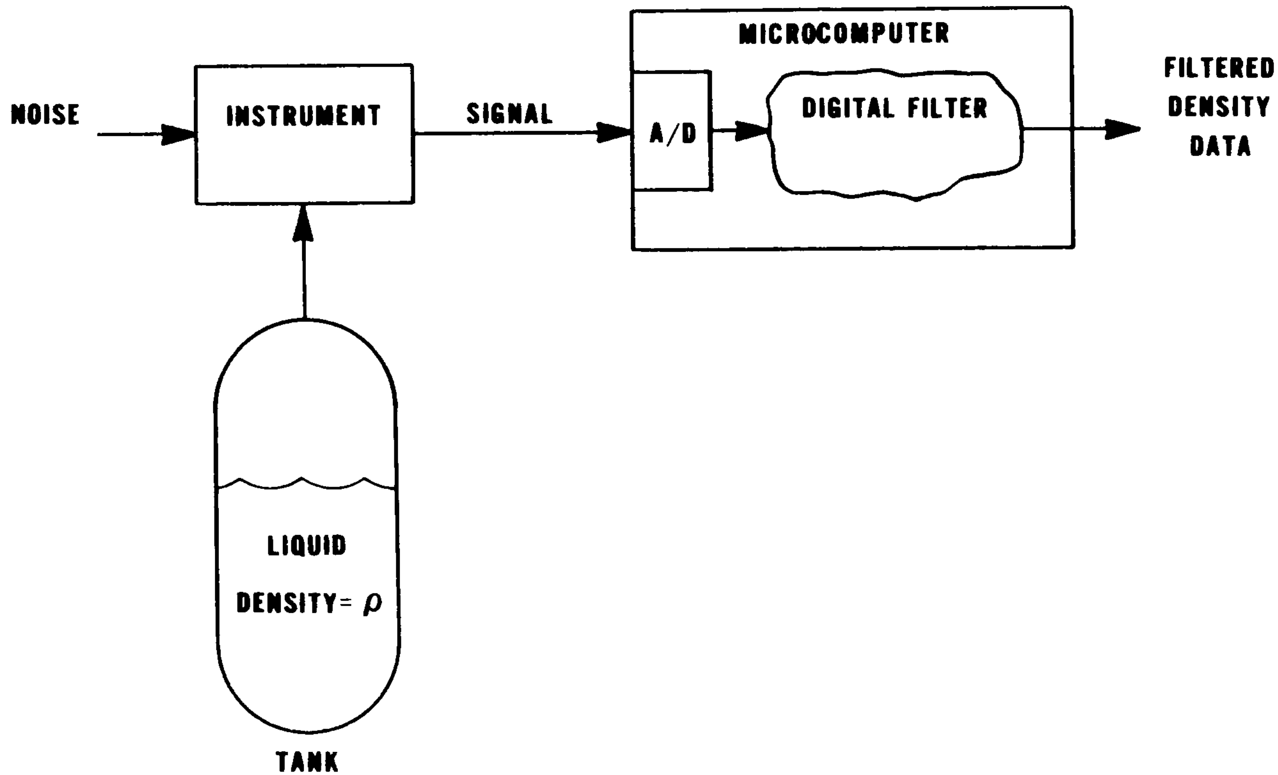


Figure 1. Conceptual block diagram of a digital filter in a liquid density measurement device.

the underlying theory behind the purged dip-tube measurement mechanism, and the development of a mathematical model. Chapter III discusses the selection of the filtering method and the subsequent filter design, and Chapter IV describes the implementation of the resulting filter and its performance in the overall measurement system. This paper concludes with a short summary in Chapter V, followed by supporting appendices.

CHAPTER II

DESCRIPTION OF THE MEASUREMENT SYSTEM

The experimental liquid density measurement system consists of a dual dip-tube liquid density probe, a pneumatic source and coupling system, a differential pressure sensor, and a microcomputer-based data acquisition system. This chapter will give a full description of each of the components of the measurement system, followed by an overview of the measurement mechanism and an analysis of pressure line characteristics.

System Description

The dual dip-tube density probe is shown in Figure 2. The unit is constructed of 3/8-in.-OD stainless steel tubing welded through a 3/16-in.-thick piece of steel plate stock. The tubes are mounted such that their axes are parallel and separated by approximately 2 in. A small reinforcement weldment is located near the end of the shorter tube to maintain the relative axial separation and minimize any stray vibrations on either tube. A triple-point, spring-loaded pivot mount connects the main supporting plate and the support block. Each pivot point features an Allen adjustment screw for leveling fine adjustments, and a small bubble level is mounted on the main supporting plate for level reference.

The immersible tube ends are precision milled to be parallel to the top of the main support plate and therefore to the level reference plane. The tube end separation was caliper measured to be 0.9904 in.

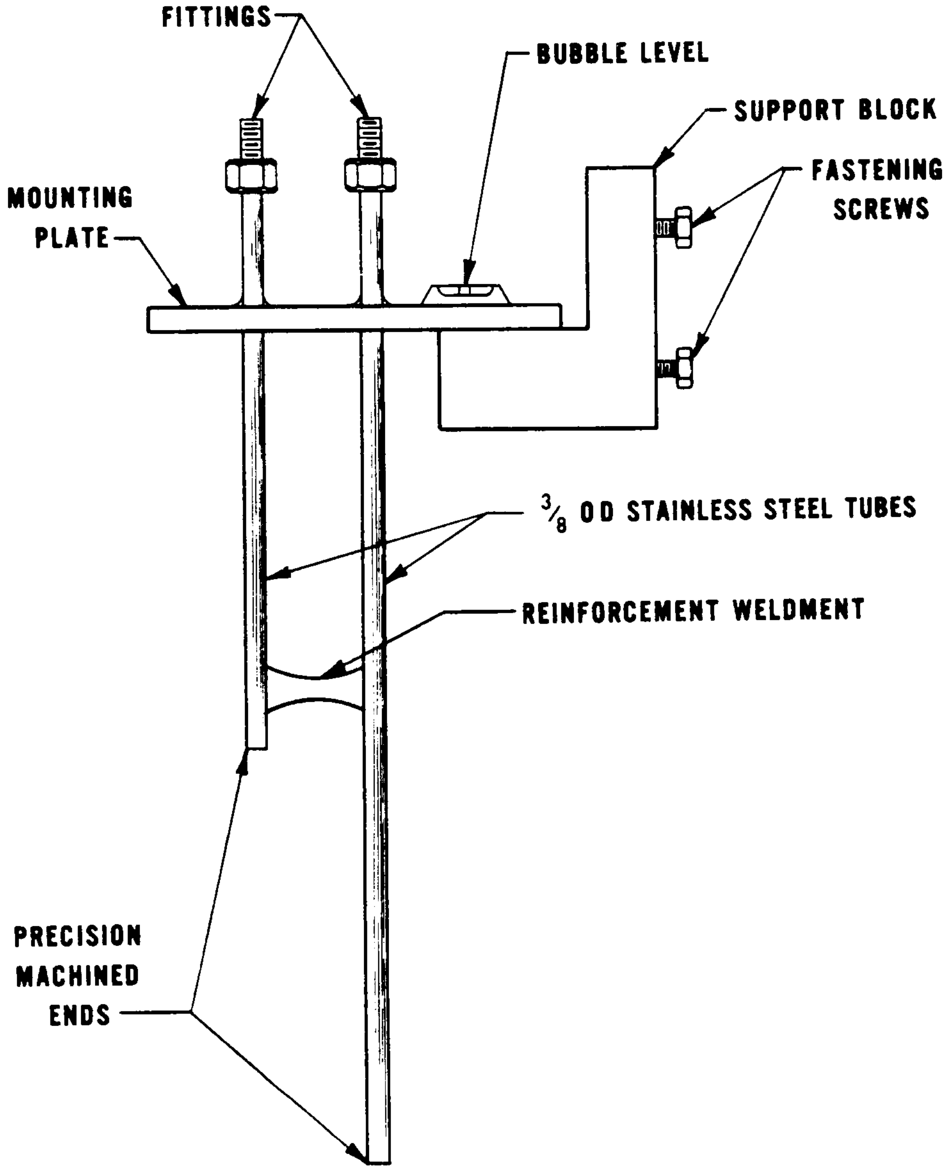


Figure 2. Dual dip-tube liquid density probe.

The upper ends of the tubes are fitted with Swagelok fittings for hook-up convenience.

During operation the dip-tube density probe produces a pressure differential between the two tubes that is related to liquid density. This pressure signal is coupled from the dip-tube probe to the differential pressure sensor by two 9-ft lengths of 1/4-in.-OD plastic pneumatic tubing. The coupling tubes also serve as a purge flow medium between the density probe and the pneumatic source. The purge flow is used to maintain the displacement of the liquid column in each tube and prevent liquid vapor from back-flowing and condensing in the differential pressure sensor. Demineralized water at a density of 0.997 kg/L was used as the process variable throughout this effort.

The pneumatic source is shown in Figure 3. The pressure regulator valve reduces 90-psig plant air to a nominal 25-psig level. The 25-psig air is then fed through a needle valve/rotameter assembly for precise manual setting of the density purge flow rate. During operation the rate was typically set to 0.1 standard cubic feet per hour (SCFH). An isolation/block valve assembly is used to connect the differential pressure sensor to the pneumatic source system.

A Validyne DP-15 differential pressure sensor (see Figure 4) is used to measure the density probe pressure signal. The DP-15 is a diaphragm type pressure transducer based on the magnetic-reluctance principle. In the pressure transducer, a diaphram of magnetically permeable stainless steel is rigidly mounted between two assembly half-blocks in such a manner that any pressure increase on one side of the diaphragm causes it to deflect. Embedded in each assembly half-block

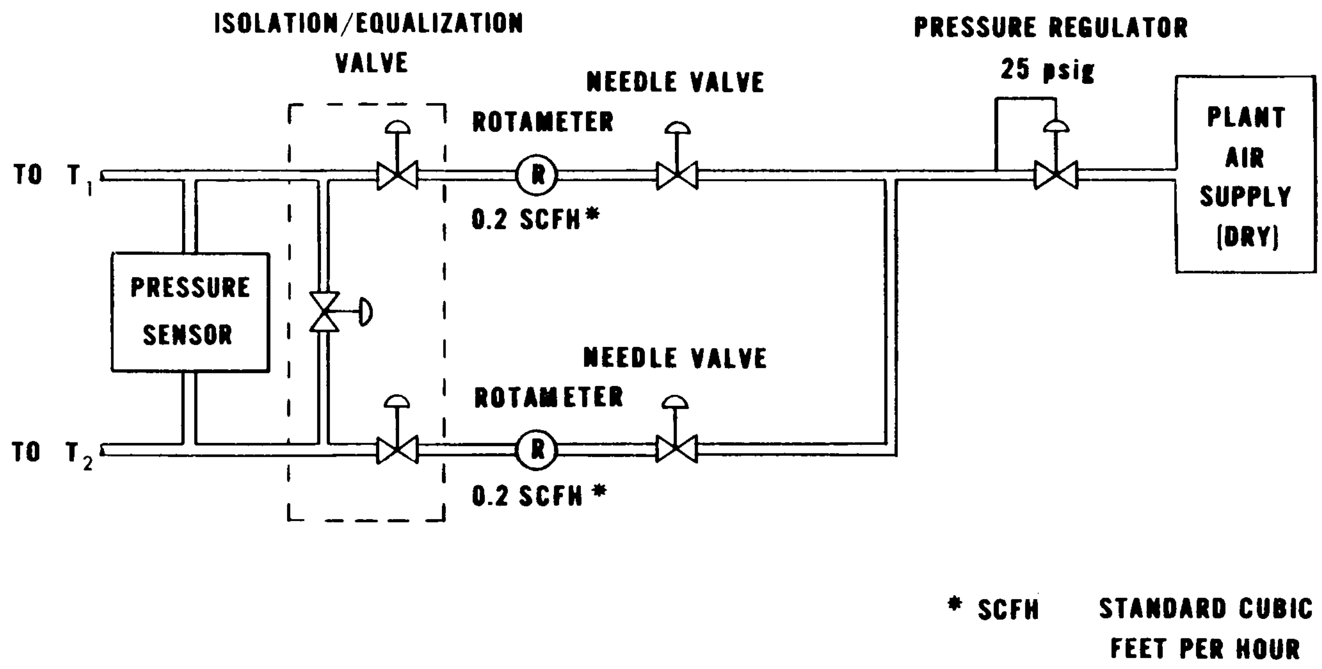


Figure 3. Pneumatic system block diagram.

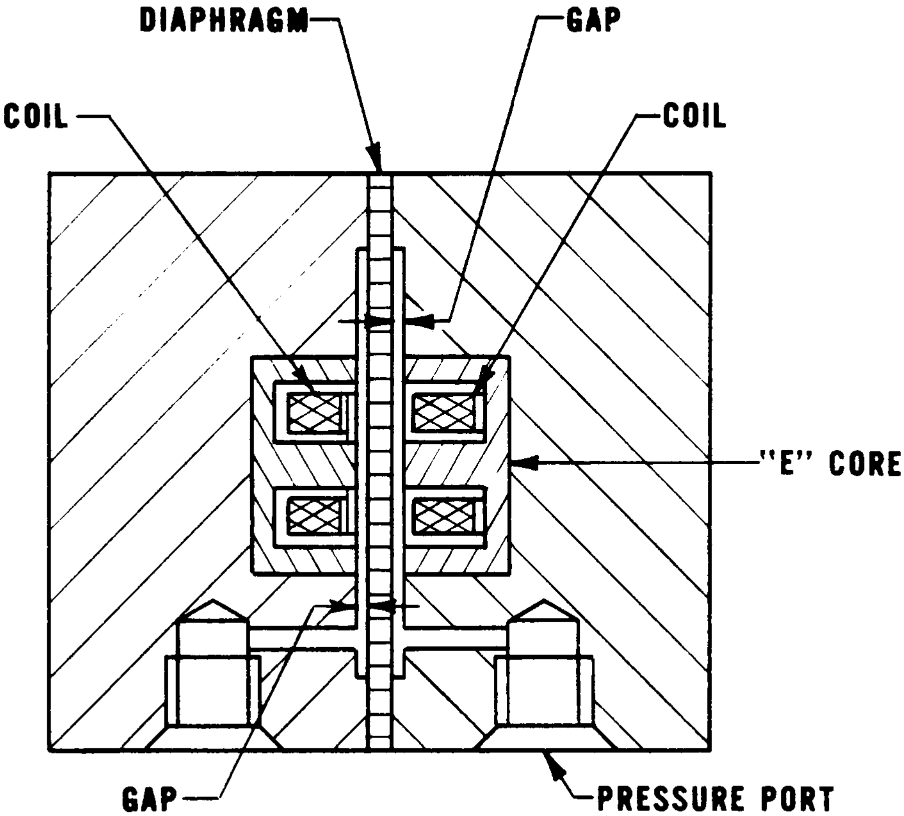


Figure 4. Cross-section of the Validyne DP-15 differential pressure sensor.

is an electromagnet coil/core assembly. When assembled the stainless diaphragm is normally flat and centered between the half-block assemblies, which results in a uniformly equal gap between each coil/core assembly and the diaphragm. The magnetic flux impedance is the same for each circuit. When one side of the diaphragm is pressurized with respect to the other side, the diaphragm deflects toward the side with the lower pressure, causing the gap separating the diaphragm from the coil/core assembly to increase on one side and decrease on the other. This change in the magnetic flux path causes a corresponding change in the magnetic impedance of each circuit.

A Validyne Model CD-15 signal conditioning unit was used to provide pressure sensor calibration adjustment and a compatible output signal. In the CD-15 signal conditioning unit, the coils are connected in a bridge circuit and are excited by an ac carrier signal. The voltage across each coil varies proportionally to the differential pressure applied across the diaphragm. A carrier demodulator circuit is used to produce a dc signal proportional to the pressure applied across the transducer.

Span and zero adjustments on the signal conditioning unit were adjusted to provide a ±10-V dc output signal corresponding to ±14.0 in. of H_2O pressure across the DP-15 transducer.

The data acquisition system is shown in Figure 5. It was built around a Digital Equipment Corp. (DEC) LSI-11/2 microcomputer. It features 32K words of random access memory, a DLV-11J multi-port RS-232-C interface, a Data Translation DT2769 real-time clock, and a Data System Design (DSD) 747 double-sided, double-density dual 8-in.

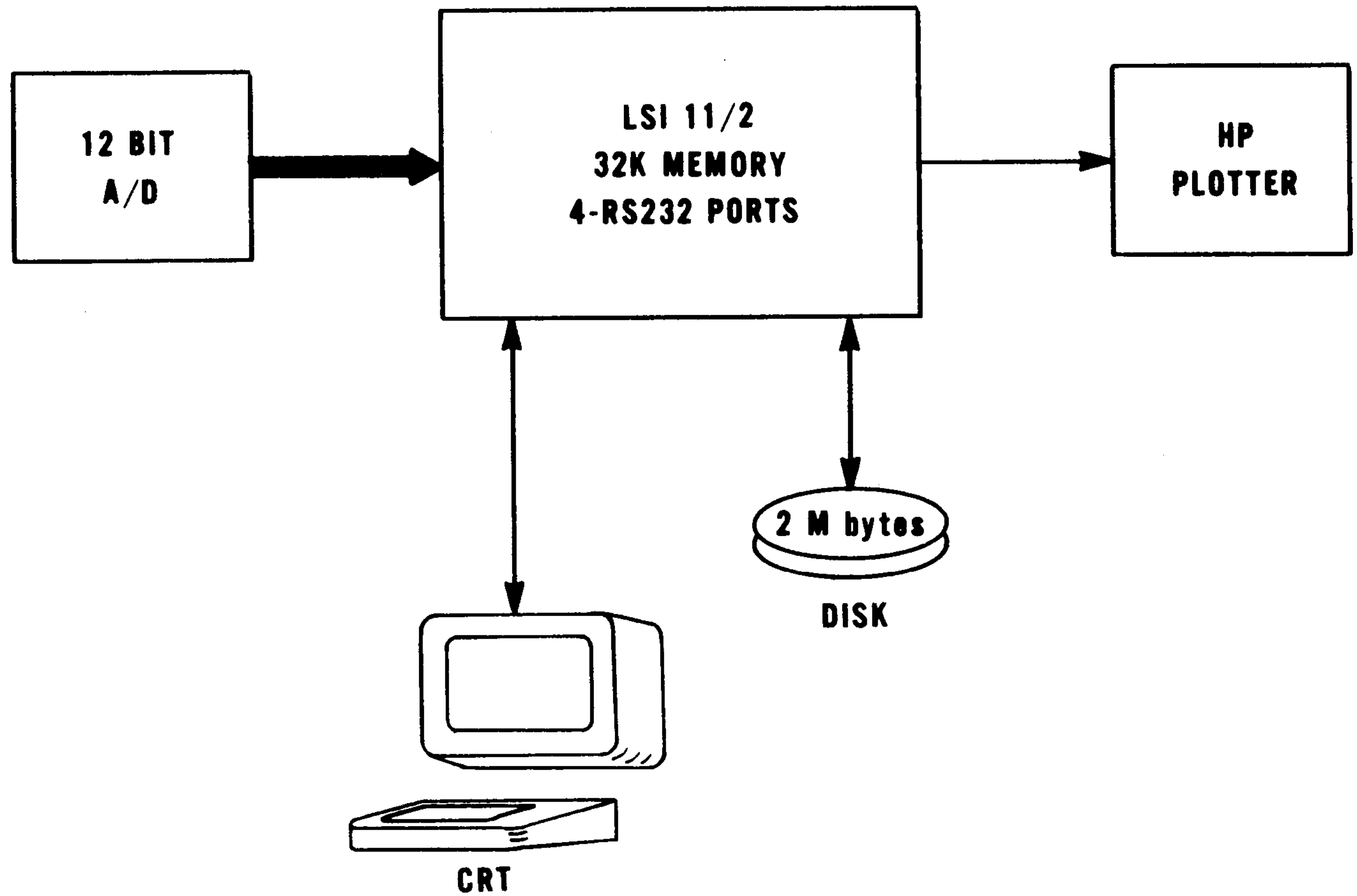


Figure 5. Block diagram of the data acquisition system for liquid density measurement.

floppy disk system. Analog-to-digital (A/D) conversion is provided by a Data Translation D2782 A/D card.

A Visual Model 100 CRT terminal and a DEC LA-120 printer are used for operator interface and software development. A Hewlett-Packard 7225A single-pen plotter is used to plot data graphically. The DEC RT-11 version 4.0 real-time operating system was utilized for this work.

The microcomputer-based data acquisition system sampled the input density signal at a rate of 500 Hz. A double buffering technique was utilized to allow simultaneous processing and storage of past data values while the real-time density sampling continued. The buffer size was designed to provide a 10-block (or 5120-byte) quantity of data for processing and disk storage each time the double buffer swapped out.

The laboratory environment did not permit a permanent installation of the density probe for the duration of this work. To mitigate the effect of this restriction, a master data file was created to provide a reference data standard to work with. Data values were stored on disk in an unformatted integer form to minimize disk requirements.

Measurement Mechanism

The purged dip-tube density measuring technique is based on the principal that liquid level can be determined by measuring the static pressure required to displace the liquid column from a vertically positioned tube immersed in the liquid. That principal is illustrated in Figure 6. In the example shown, a gas is slowly injected into the supply end of the tube by controlling the flow control valve (FCV).

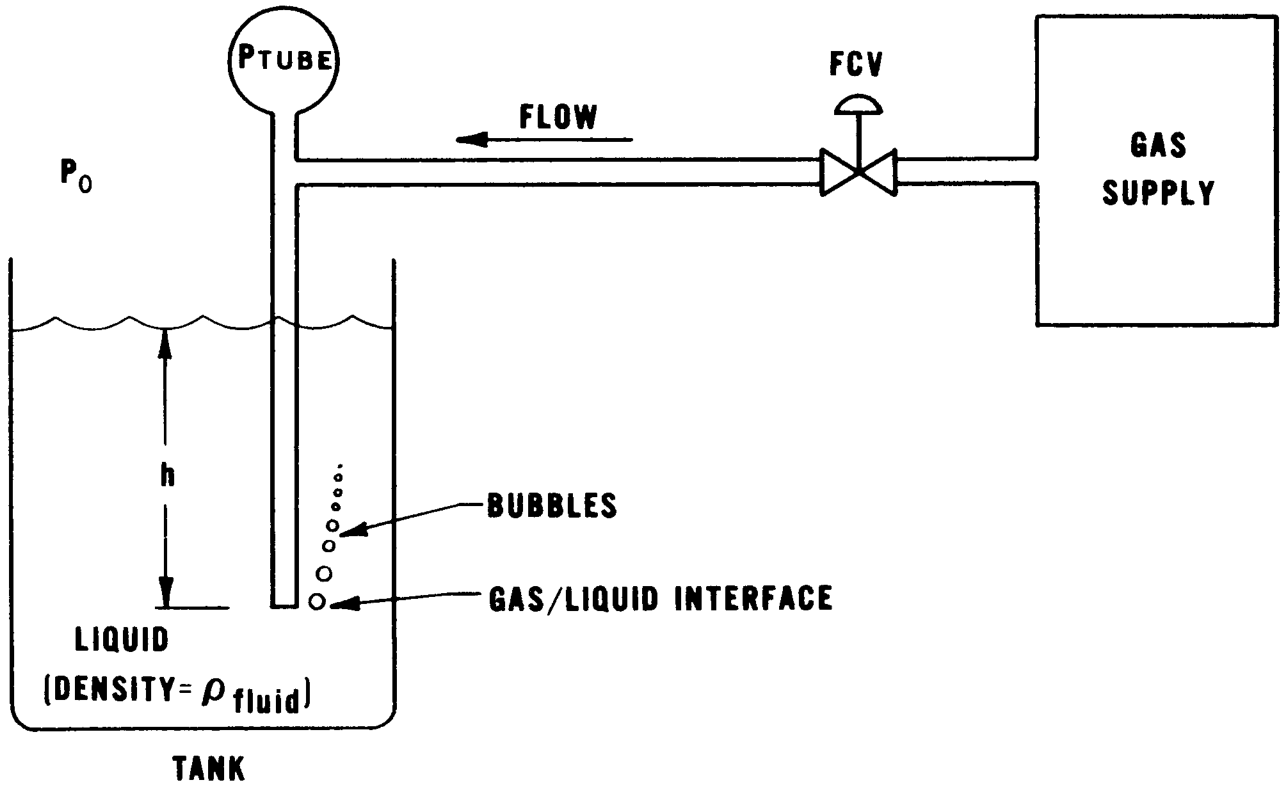


Figure 6. Conceptual diagram of the dip-tube liquid level measurement technique.

This flow is set such that a bubble is released from the immersed end of the tube every few seconds.

If this gas purge flow is very low and the tube diameter is sufficiently large, the resulting pressure drop of the gas along the tube can be neglected. With this assumption taken, the total pressure drop through the entire tube is described by

$$P_{tube} + \int_{o}^{h} g\rho_{gas}(Z)dZ = \int_{o}^{h} g\rho_{fluid}(Z)dZ + P_o \ , \qquad (1)$$

where h is the tube displacement depth referenced in the Z (vertical) axis, g is the gravitational constant, ρ_{gas} is the density of the purge gas, ρ_{fluid} is the density of the liquid, P_o is atmospheric pressure, and P_{tube} is the tube gas pressure referenced to atmopheric pressure P_o.

The subject of this paper involves liquids with uniform densities; therefore Equation (1) can be reduced to

$$P_{tube} + \int_{o}^{h} g\rho_{gas}(Z)dZ = gh\rho_{fluid} + P_o \ . \qquad (2)$$

If a well-behaved gas such as dry air is used for purging, the effects of gas density variations in the tube can be neglected and Equation (2) reduces to

$$P_{tube} + gh\rho_{gas} = gh\rho_{fluid} + P_o \ . \qquad (3)$$

Rearranging Equation (3) gives

$$P_{tube} = gh(\rho_{fluid} - \rho_{gas}) + P_o \ . \qquad (4)$$

Solving Equation (4) for h, the liquid level, gives

$$h = \frac{P_{tube} - P_o}{g(\rho_{fluid} - \rho_{gas})} \quad . \tag{5}$$

By adding a second tube (t_2) immersed to a different depth (h_2) in the liquid with respect to tube one (t_1) at depth h_1 as shown in Figure 7, any of the previous equations can be applied to each tube system. In particular, taking Equation (4) and rewriting to represent each tube system, the resulting equation pair is

$$P_{t1} = gh_1(\rho_{fluid} - \rho_{gas}) + P_o \tag{6}$$

and

$$P_{t2} = gh_2(\rho_{fluid} - \rho_{gas}) + P_o \quad . \tag{7}$$

Subtracting Equation (7) from Equation (6) yields

$$P_{t1} - P_{t2} = g(\rho_{fluid} - \rho_{gas})(h_1 - h_2) \quad . \tag{8}$$

If the separation of the two tubes, Δh, is known, then

$$\Delta h = h_1 - h_2 \quad . \tag{9}$$

Substituting Equation (9) into Equation (8) gives

$$P_{t1} - P_{t2} = g(\rho_{fluid} - \rho_{gas})\Delta h \quad . \tag{10}$$

Since dry air is being used as the purge gas, the effects of the gas weight over the length Δh on the value $P_{t1} - P_{t2}$ is approximately three orders of magnitude smaller than the effects of the liquid. Therefore the effects of ρ_{gas} can be neglected, yielding

$$\Delta P = g\rho\Delta h \quad , \tag{11}$$

where

$$\rho = \rho_{fluid} \tag{12}$$

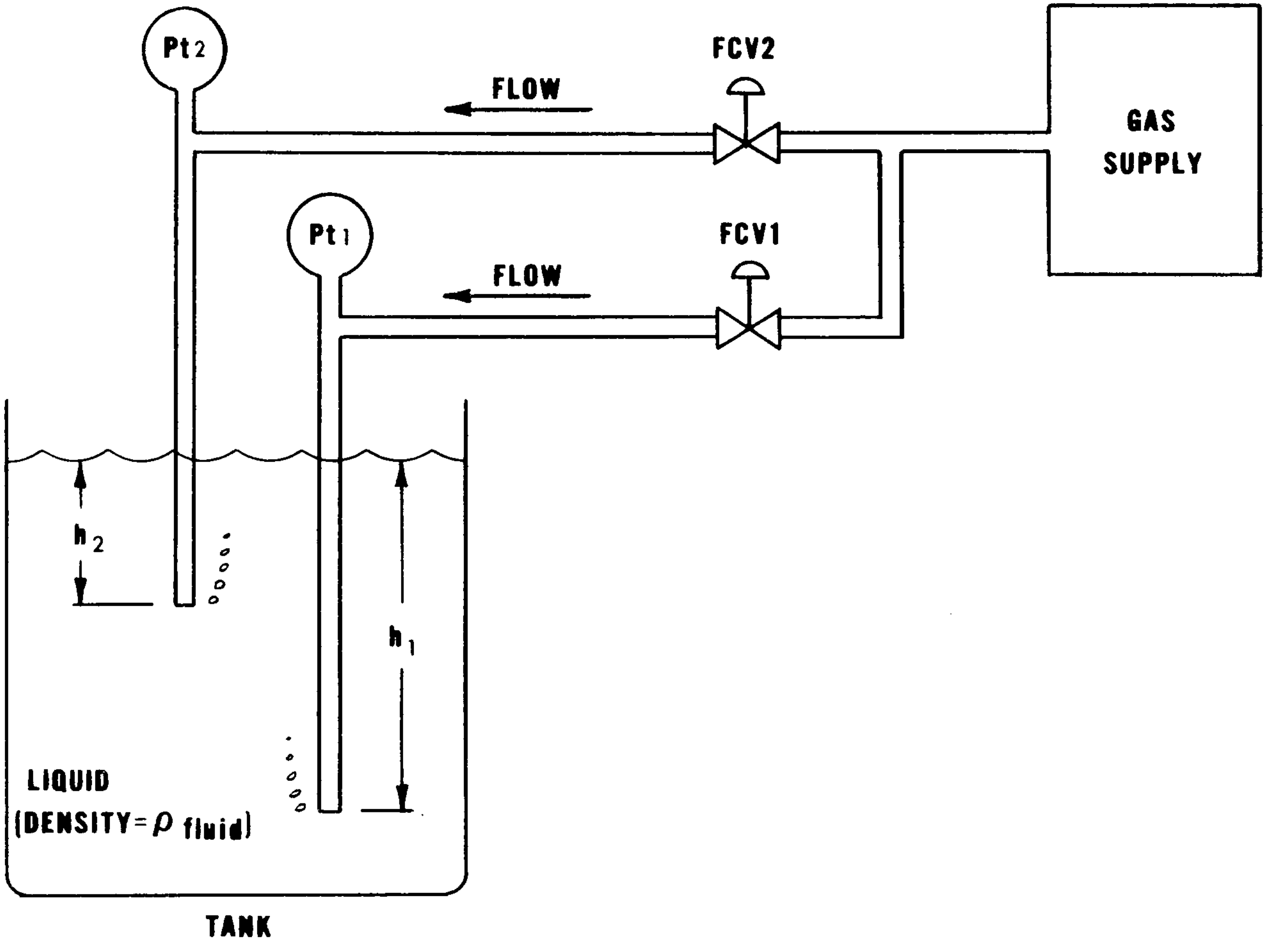


Figure 7. Conceptual diagram of the dual dip-tube liquid density measurement technique.

and

$$\Delta P = P_{t1} - P_{t2} \ . \tag{13}$$

It is now evident that the liquid density can be determined by rearranging Equation (11) to yield

$$\rho = \frac{\Delta P}{g \Delta h} \ . \tag{14}$$

Equation (14) is the theoretical basis for the operation of the dip-tube density instrument and is used to characterize the operation of the laboratory setup used in this study.

In actual operation, the bubbles released from the dip-tube ends cause detectable pressure variations in the pneumatic coupling lines. This pressure noise is typically 0.3 in. of water in magnitude. The noise frequency range is generally high in comparison to the density signal, but it is detectable by most common pressure transducers. Figure 8 shows a typical 2-s plot of this phenomenon as recorded by the experimental setup.

Analysis of Pressure Line Characteristics

Two methods, distributed and lumped parameter, were utilized in modeling the density probe. Both of these approaches are described below.

The characteristics of gas-filled pneumatic lines can be approximated by electrical equivalents (Buckley, 1964). Specifically, flow and pressure are analogous to current and voltage. In addition, inductance, capacitance, and resistance equivalent terms can be derived for pneumatic lines. Therefore, a completely passive electrical component model can be derived for a pneumatic line to predict flow or pressure behavior over time.

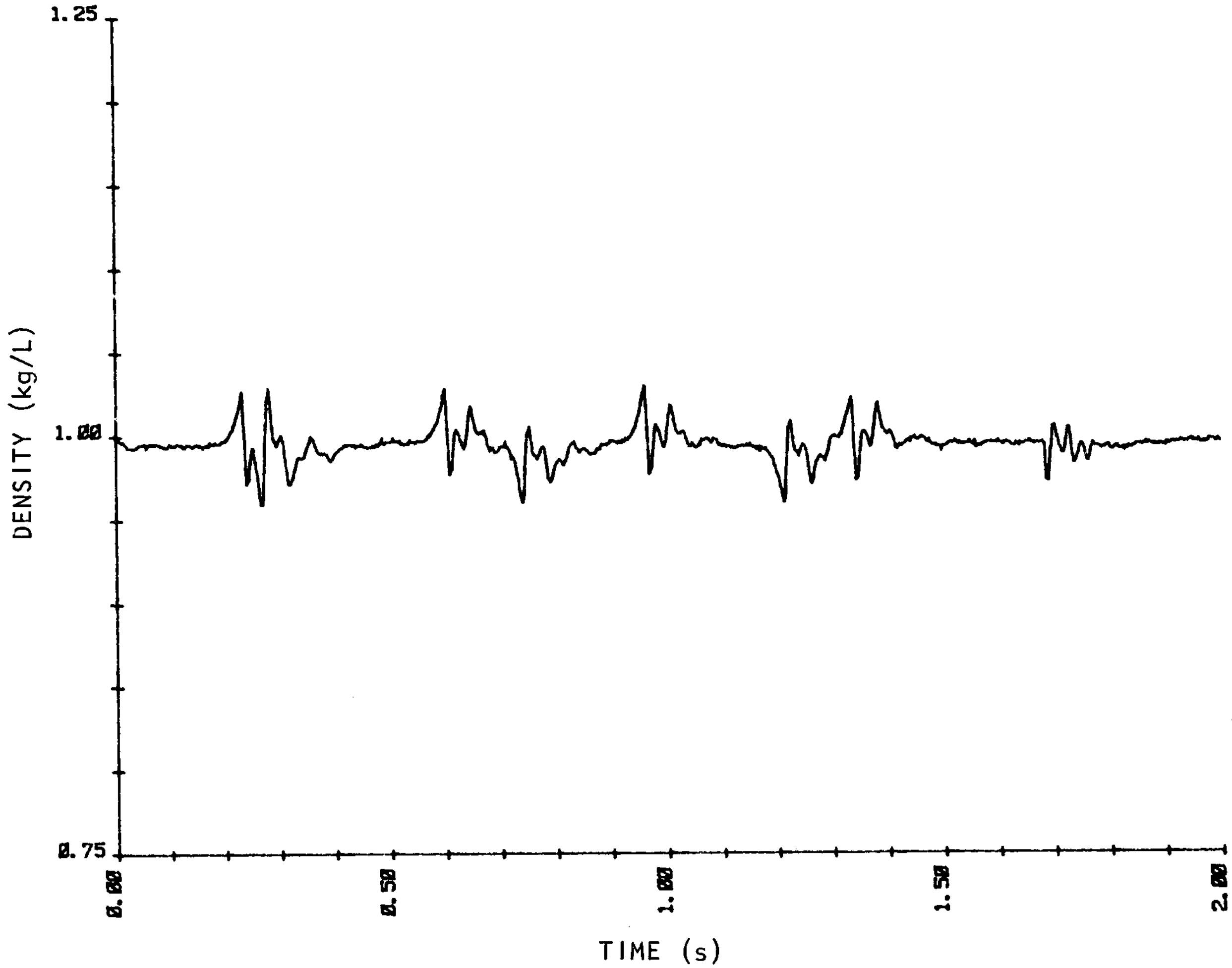


Figure 8. Typical unfiltered density data.

To assist in simplifying the modeling problem, the following conditions were specified:

1. The conductance term (losses) in the electrical equivalence model is assumed to be zero since there are no leakage points in either of the pneumatic lines.
2. The effects contributed by any tube couplings is assumed to be negligable due to the very small gas flow in the tubes.
3. All upstream pneumatic system components are neglected.
4. Rigid tube walls are assumed due to the small system pressures used.
5. The total effective coupling tube length for each line is assumed to be approximately ten feet, with an inside diameter of 0.16 in.

The pressure sensor can be regarded as a capacitance equivalent in the transmission line model. Pressure values are low enough that the diaphragm can be assumed to be rigid. Therefore the capacitance term determined by the cavity volume on either side of the diaphragm can be found by

$$C_{sensor} = \frac{\overline{V}}{\overline{P}} + \frac{A_B^{\,2}}{K_B} , \qquad (15)$$

where $\overline{V}$ is the nominal volume of the cavity, $\overline{P}$ is the nominal cavity pressure, A_B is the diaphragm cross-sectional area, and K_B is the diaphragm spring constant 4.0×10^3 lb/ft. From the manufacturer's literature it is found that the nominal cavity volume of the pressure sensor is 4×10^{-3} in^3.

Substituting into Equation (15) gives

$$C_{sensor} = 3.4 \times 10^{-9} \frac{ft^5}{lb} . \qquad (16)$$

Because the pressure sensor diaphram is assumed to be rigid, pressure coupling from one side of the sensor to the other side can be assumed to be zero. This allows each pneumatic line to be modeled separately. Since both pneumatic lines are approximately the same length, system characteristics predicted by a model of one line will represent characteristics of the whole dip-tube system.

The electrical equivalent relationships for pneumatic lines are shown in Table 1. Substituting appropriate values into the relationships in Table 1 gives the results from which the values in Table 2 are obtained.

<u>Transmission line analogy</u>. Recalling the traveling wave method for finite transmission lines provides the relationships

$$V(\chi,s) = \frac{V(\ell,s) + Z_o I(\ell,s) e^{\gamma(\ell-\chi)}}{2\,Z_o} + \frac{V(\ell,s) - Z_o I(\ell,s) e^{-\gamma(\ell-\chi)}}{2\,Z_o} \qquad (17)$$

and

$$I(\chi,s) = \frac{V(\ell,s) + Z_o I(\ell,s) e^{\gamma(\ell-\chi)}}{2\,Z_o} - \frac{V(\ell,s) - Z_o I(\ell,s) e^{-\gamma(\ell-\chi)}}{2\,Z_o} \qquad (18)$$

where $V(\chi,s)$ is the voltage at point χ,

$I(\chi,s)$ is the current at point χ,

$V(\ell,s)$ is the voltage at the receiving end,

$I(\ell,s)$ is the current at the receiving end,

Table 1. Per-unit electrical equivalent values

Capacitance (C´)	$\frac{\pi D^2 \ell}{4P}$	$\frac{ft^5/lb}{ft}$
Inductance (L´)	$\frac{\rho \ell}{32.2\pi r^2}$	$\frac{lb \cdot s^2}{ft^5}/ft$
Resistance (R´)	$\frac{0.78\mu}{A^2}$	$\frac{lb \cdot s}{ft^5}/ft$

where D = Tube diameter (0.016 in.)

ℓ = Tube length (10 ft)

P = Gas pressure (2174 lb/ft^2)

MW = Gas molecular weight (28.964)

ρ = Gas density (0.074 lb/ft^3)

μ = Gas viscosity (1.2226×10^{-5})

T = Gas temperature (26°C)

A = Cross sectional area (0.16 $in.^2$)

Table 2. Pneumatic line per-unit-length characteristic values

$C' = 1.004 \times 10^{-7}$	$\frac{ft^5/lb}{ft}$
$L' = 10.53$	$\frac{lb \cdot s^2}{ft^5}/ft$
$R' = 200.3$	$\frac{lb \cdot s}{ft^5}/ft$

Z_o is the line characteristic impedance, and

γ is the line propagation constant.

Rearranging Equations (17) and (18) gives

$$V(\chi,s) = V(\ell,s)\left[\frac{1+\frac{Z_o}{Z_r}\ e^{\gamma(\ell-\chi)}}{2} + \frac{1-\frac{Z_o}{Z_r}\ e^{-\gamma(\ell-\chi)}}{2}\right] \tag{19}$$

and

$$I(\chi,s) = \frac{V(\ell,s)}{Zo}\left[\frac{1+\frac{Z_o}{Z_r}\ e^{\gamma(\ell-\chi)}}{2} - \frac{1-\frac{Z_o}{Z_r}\ e^{-\gamma(\ell-\chi)}}{2}\right]. \tag{20}$$

The first term inside each bracket represents the incident wave traveling rightward and the second term represents the reflected wave traveling leftward. Defining

$$\Gamma_r = \frac{\text{Reflected wave at receiving end}}{\text{Incident wave at receiving end}}\text{, gives} \tag{21}$$

$$\Gamma_r = \frac{Z_r - Z_o}{Z_r + Z_o} \tag{22}$$

where

Z_r is the receiving end impedance, and

Z_o is the characteristic impedance.

Substituting Equation (22) into Equations (19) and (20) yields

$$V(\chi,s) = V(0,S)e^{-\gamma x}\ \frac{1+\Gamma_r e^{-2\gamma(\ell-\chi)}}{1+\Gamma_r e^{-2\gamma\ell}} \tag{23}$$

and

$$I(\chi,s) = I(0,2)e^{-\gamma\chi}\ \frac{1-\Gamma_r e^{-2\gamma(\ell-\chi)}}{1-\Gamma_r e^{-2\gamma\ell}}\ . \tag{24}$$

Since G = 0,

$$\gamma = \sqrt{(R+Ls)Cs} \ , \tag{25}$$

and

$$Zo = \sqrt{\frac{R + Ls}{Cs}} \ , \tag{26}$$

Equation (23) now becomes

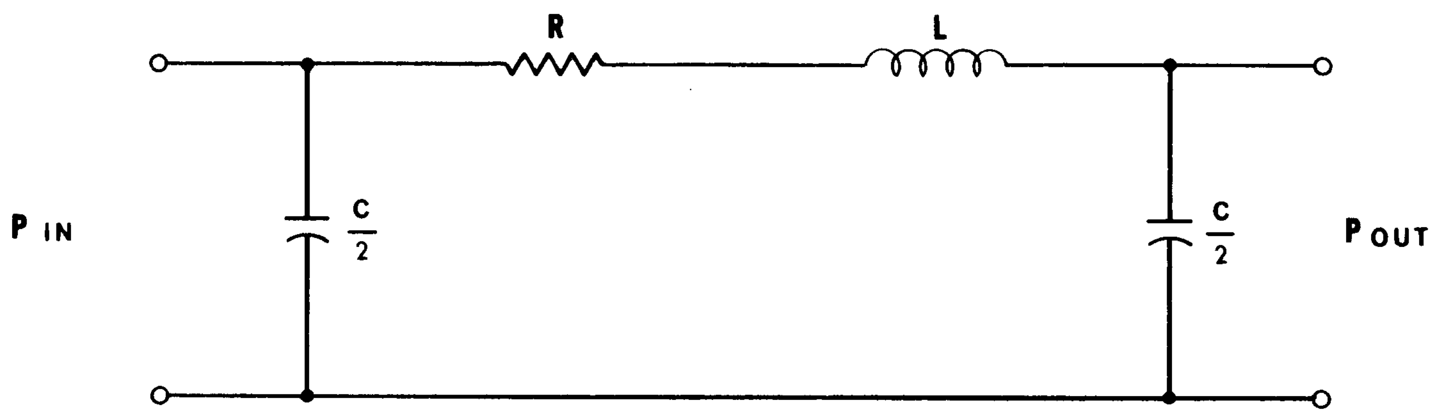
$$V(\chi,s) = V(0,s)\ e^{-\chi\sqrt{(R + Ls)Cs}}\ \frac{1 + \Gamma_r\ e^{-2(\ell-\chi)\sqrt{(R + Ls)Cs}}}{1 + \Gamma_r\ e^{-2\ell\sqrt{(R + Ls)Cs}}} \ . \tag{27}$$

If an infinite receiving end impedance is assumed at a point of $\chi = \ell$, Equation (27) reduces to

$$V(\ell,s) = V(0,s)e^{-\ell\sqrt{(R + Ls)Cs}}\ \frac{2}{1 + e^{-2\ell\sqrt{(R + Ls)Cs}}} \ . \tag{28}$$

The solution to Equation (28), while accurately conveying the more complex nature of the pneumatic line signal corruption, is non-trivial.

Lumped parameter method. Since the ultimate goal of this work is to eliminate noise corruption from the density signal, a simpler approach can be realized by utilizing a lumped-parameter model. The general form of a lumped-parameter model equivalent to a pneumatic system is shown in Figure 9. Component values R, L, and C are determined by multiplying the per-unit-length values as shown in Table 2 by the appropriate tube length. If a tube length of 10 ft is assumed, the corresponding lumped parameters are shown in Table 3. The impedance contribution of the pressure sensor is ignored since it is small in comparison to the lumped-capacitance value in Table 3. This assumption



$$T(s) = \frac{P_{OUT}(s)}{P_{IN}(s)}$$

Figure 9. Diagram of a lumped-parameter equivalent model.

Table 3. Lumped-parameter pneumatic line characteristic values

$C = C'\ell = 1.004 \times 10^{-6}$	ft^5/lb
$L = L'\ell = 105.3$	$\frac{lb \cdot s^2}{ft^5}$
$R = R'\ell = 2003$	$\frac{lb \cdot s}{ft^5}$

NOTE: $\ell = 10$ ft

yields an effective pneumatic line terminating impedance of infinity. If these values are substituted into the equivalent network shown in Figure 9, the resulting transfer function relating input pressure $P_{in}(s)$ and output pressure $P_{out}(s)$ is

$$T(s) = \frac{P_{out}(s)}{P_{in}(s)} = \frac{37853.3}{s^2 + 19.0\ S + 37853.3}\ , \qquad (29)$$

which exhibits the form of the classical second-order transfer function

$$T(s) = \frac{W_n^{\ 2}}{S^2 + 2\beta W_n S + W_n^{\ 2}} \qquad (30)$$

where

$$W_n = 194.56$$

and

$$\beta = 0.04882.$$

This predicts a primary pneumatic resonance of 30.89 Hz, which closely agrees with the observed dominant frequency of Figure 8, page 18.

CHAPTER III

DIGITAL DATA PROCESSING

Two kinds of infinite impulse response filters, step invariant and bilinear, were considered for reducing the density-probe-induced measurement noise observed in Figure 8, page 18. The following section describes the reasoning behind this preference, the selection of the filtering method, the design, and the resulting filter characteristics.

Method Selection

Digital filters generally fall into two classifications, differentiated by their response to an ideal impulse input. One class, the infinite impulse response (IIR) type, theoretically exhibits detectable output for all time following the occurrance of a single impulse on the input. The other class, the finite impulse response (FIR) type, eventually exhibits a zero output for all time following the occurrance of a single impulse on the input.

A finite impulse response filter was considered initially for implementation in the measurement system because its response to an impulse input, or disturbance, was desirable. However, this type of filter usually requires ten or more stages (delays) for an accurate approximation of its analog equivalent, and this number of filter stages presented an unacceptable processing load for the microcomputer-based data acquisition system. For this reason an infinite impulse response filter was pursued.

Two sub-classes of IIR filters, step invariant and bilinear, were investigated. Other kinds were considered, including impulse invariant and statistically based (Kalman) types, but they were rejected either for accuracy reasons or because the process statistics were not known. Both the step invariant and bilinear types of digital filters exhibit close approximations to the step response of their analog low-pass filter equivalents.

As described in Chapter II, the density probe and pneumatic coupling system approximate model predicted a system resonance at 30.97 Hz. Real-world chemical process systems generally exhibit variable time constants over the range of one second to many minutes. Liquid density usually belongs in the slower catagory due to the large tank capacities associated with the measurement. The noise reduction filter had to provide a high degree of attentuation over the noise band and higher, yet pass the real process data with a bandwidth sufficient to accurately present true process characteristics. In addition, the capacity of the host digital system to perform the filtering task in terms of its overall processing load had to be considered. For this reason a first- or second-order filter of the step invariant or bilinear type was identified as the design goal.

Filter Design

Two computer programs were developed for calculating the coefficients of first- and second-order step invariant and bilinear digital filters. Simple block diagrams and listings of these programs are shown in Appendix A. Subroutines were used to compute the coefficients for each type of filter. For second-order filters, a step invariant subroutine computed coefficients for an analog equivalent filter with a selectable damping coefficient between 0 and +1. In all cases the user was allowed to select the filter sample rate and cut-off frequency. A simulation subroutine was included for step response testing of results.

The digital filter cut-off frequency selection was based on the filter order used. A 1.0-Hz cut-off frequency was selected for second-order filters yielding a theoretical pneumatic resonant frequency attentuation of approximately 60 dB, and a 0.1-Hz cut-off frequency was selected for first-order filters yielding a theoretical pneumatic resonant frequency attentuation of approximately 51 dB.

The Z-transform transfer function of the digital filter is:

$$H(Z) = \frac{OUT(Z)}{IN(Z)} = \frac{A_0 + A_1 Z^{-1} + A_2 Z^{-2}}{1 + B_1 Z^{-1} + B_2 Z^{-2}} \, . \qquad (31)$$

Tables 4 through 7 show the filter design parameters and resulting coefficients. Figures 10 through 13 show each filter's response to an idealized step input at T = 1 s over a total time window of 10 s.

As seen in Figures 10 through 13, acceptable step response results were obtained for all of the indicated filter specifications. Both filter types, step invariant and bilinear, exhibited slight overshoot in the second-order cases.

With these results taken into account, it can be seen that a step invariant digital filter would provide acceptable approximation to an analog equivalent with one less implementation step ($A_0 = 0$). However,

Table 4. First-order step invariant filter parameters and coefficients

Sample frequency (Hz)	Cut-off frequency (Hz)	Coefficients
500	0.1	A_0 = 0.0
		A_1 = 0.9987441
		A_2 = 0.0
		B_1 = 0.9987441
		B_2 = 0.0

Table 5. First-order bilinear filter parameters and coefficients

Sample frequency (Hz)	Cut-off frequency (Hz)	Coefficients
500	0.1	A_0 = 0.627924E-3
		A_1 = 0.627924E-3
		A_2 = 0.0
		B_1 = -0.9987441
		B_2 = 0.0

Table 6. Second-order step invariant filter parameters and coefficients

Sample frequency (Hz)	Cut-off frequency (Hz)	Coefficients	Remarks
500	1.0	A_0 = 0.0	β =0.707106781
		A_1 = 0.7834099E-4	Butterworth
		A_2 = 0.7778406E-4	
		B_1 = -1.982229	
		B_2 = 0.9823853	

Table 7. Second-order bilinear filter parameters and coefficients

Sample frequency (Hz)	Cut-off frequency (Hz)	Coefficients
500	1.0	A_0 = 0.3913021E-4
		A_1 = 0.7826042E-4
		A_2 = 0.3913021E-4
		B_1 = -1.982229
		B_2 = 0.9823855

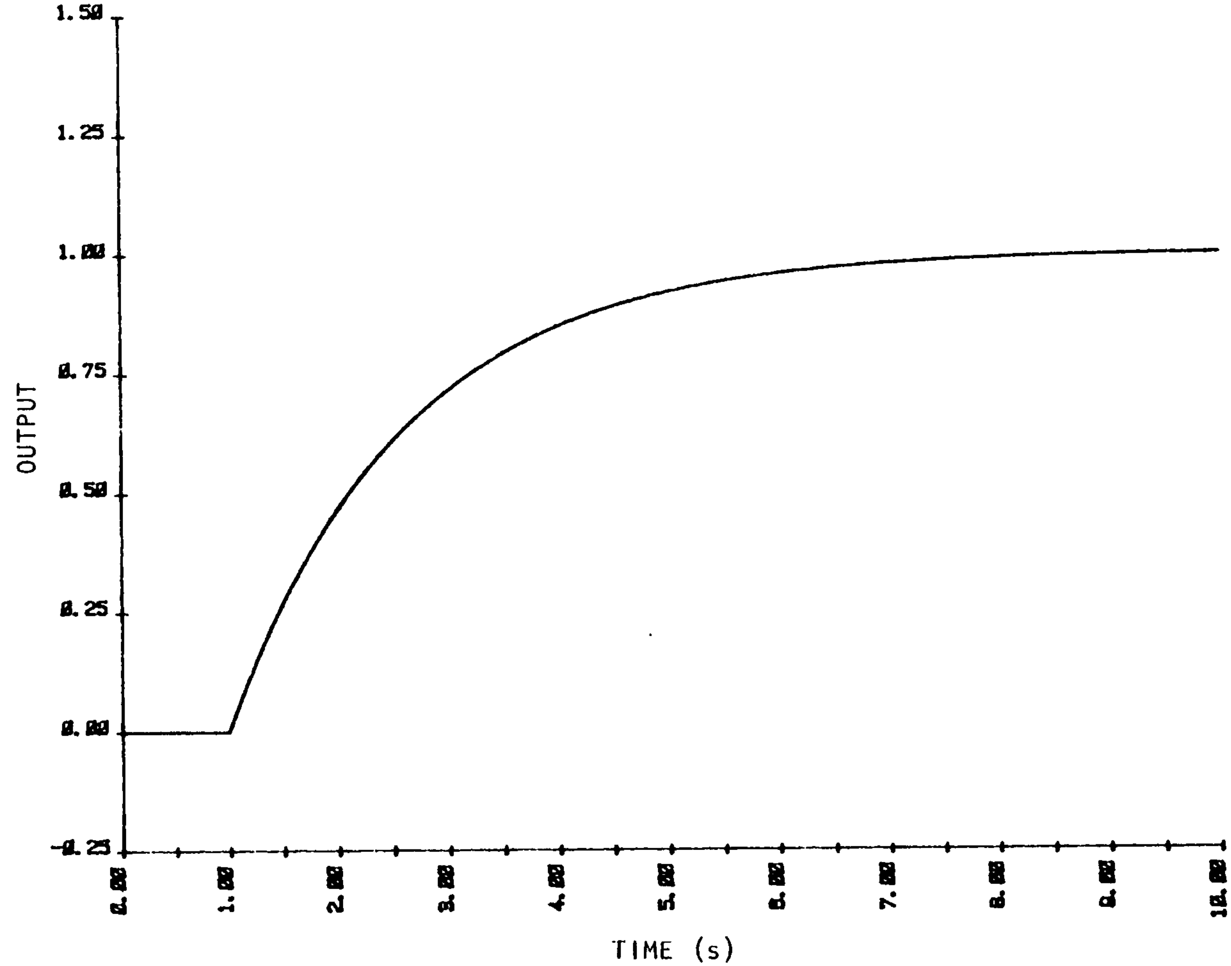


Figure 10. First-order step invariant filter step response.

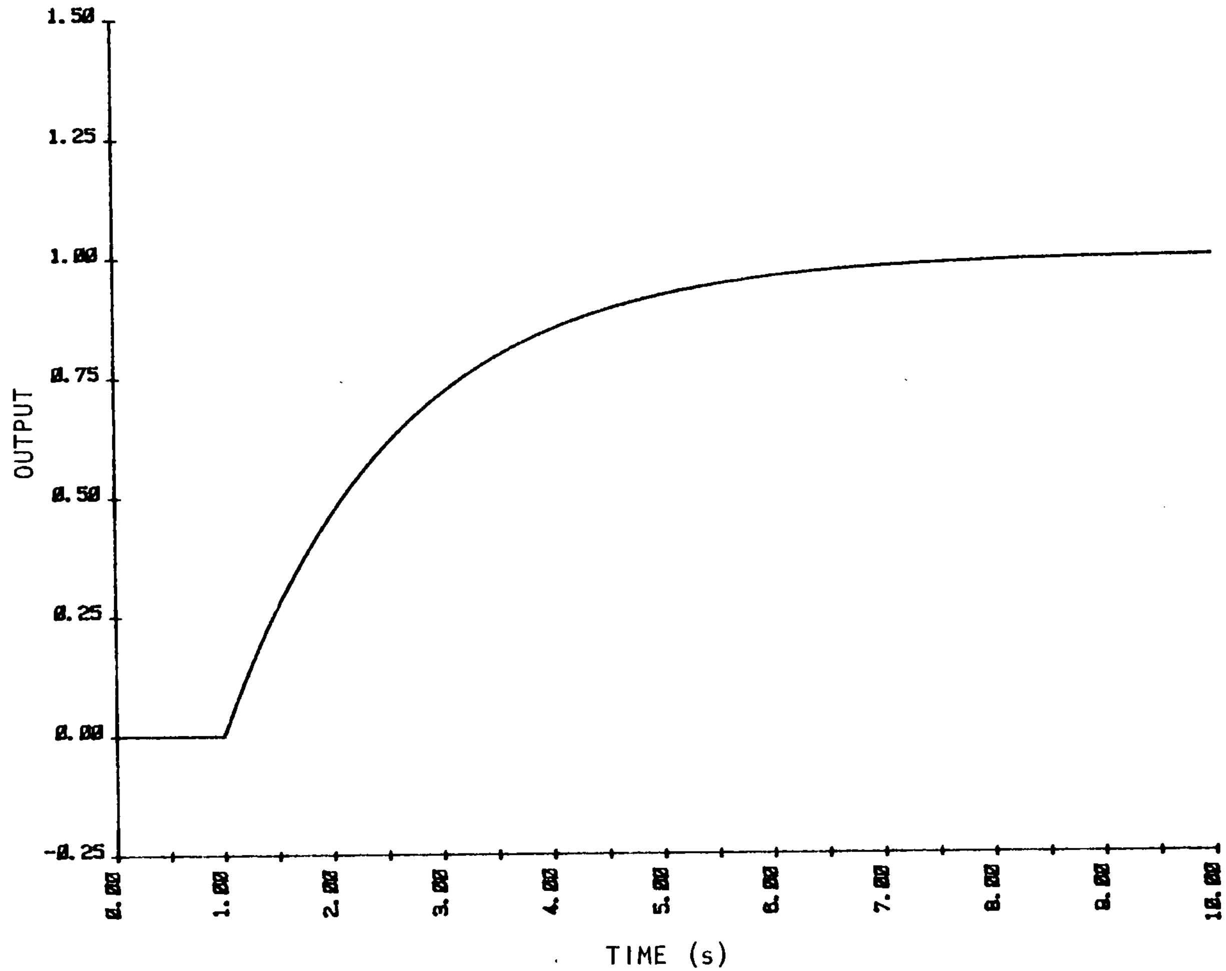


Figure 11. First-order bilinear filter step response.

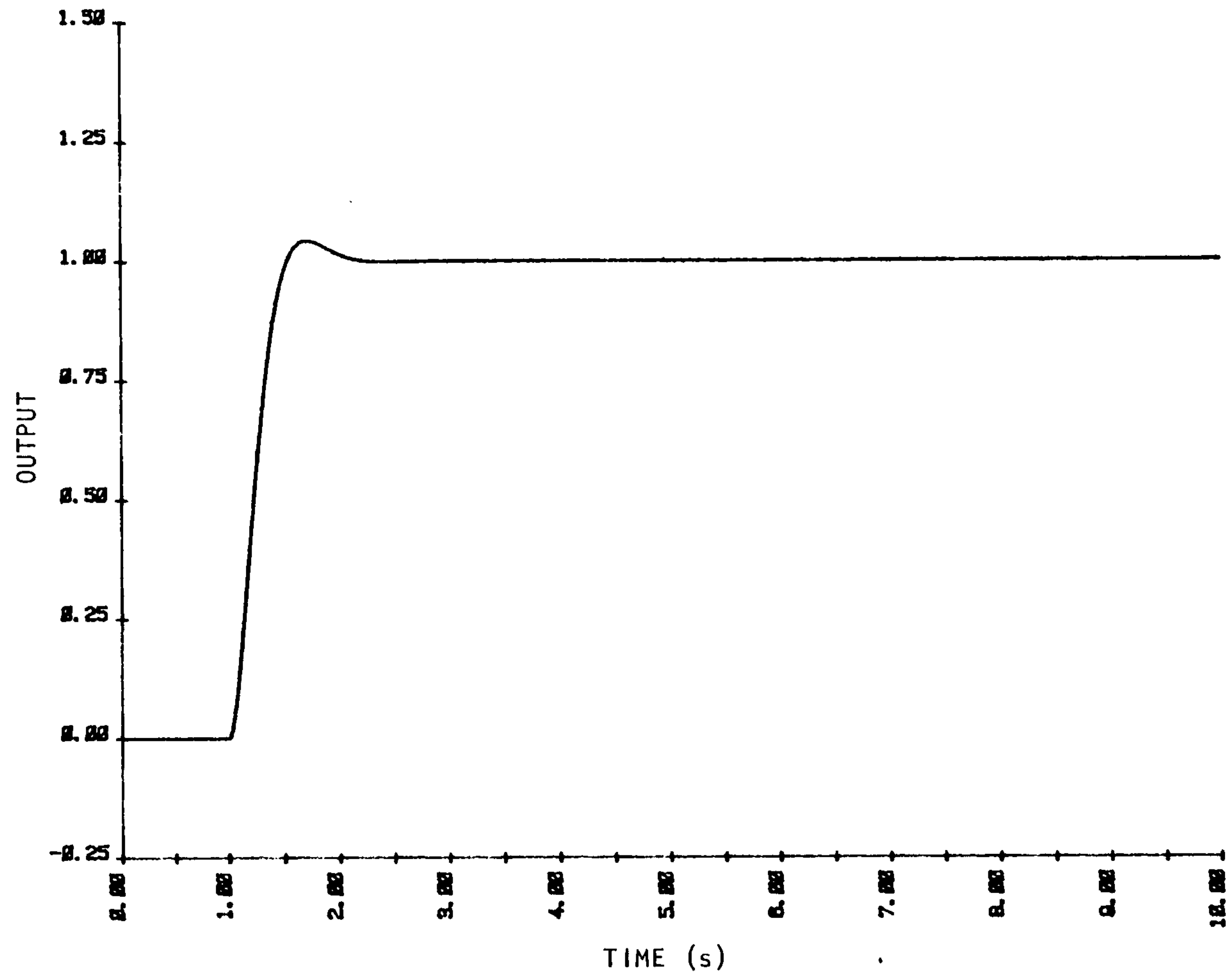


Figure 12. Second-order step invariant filter step response.

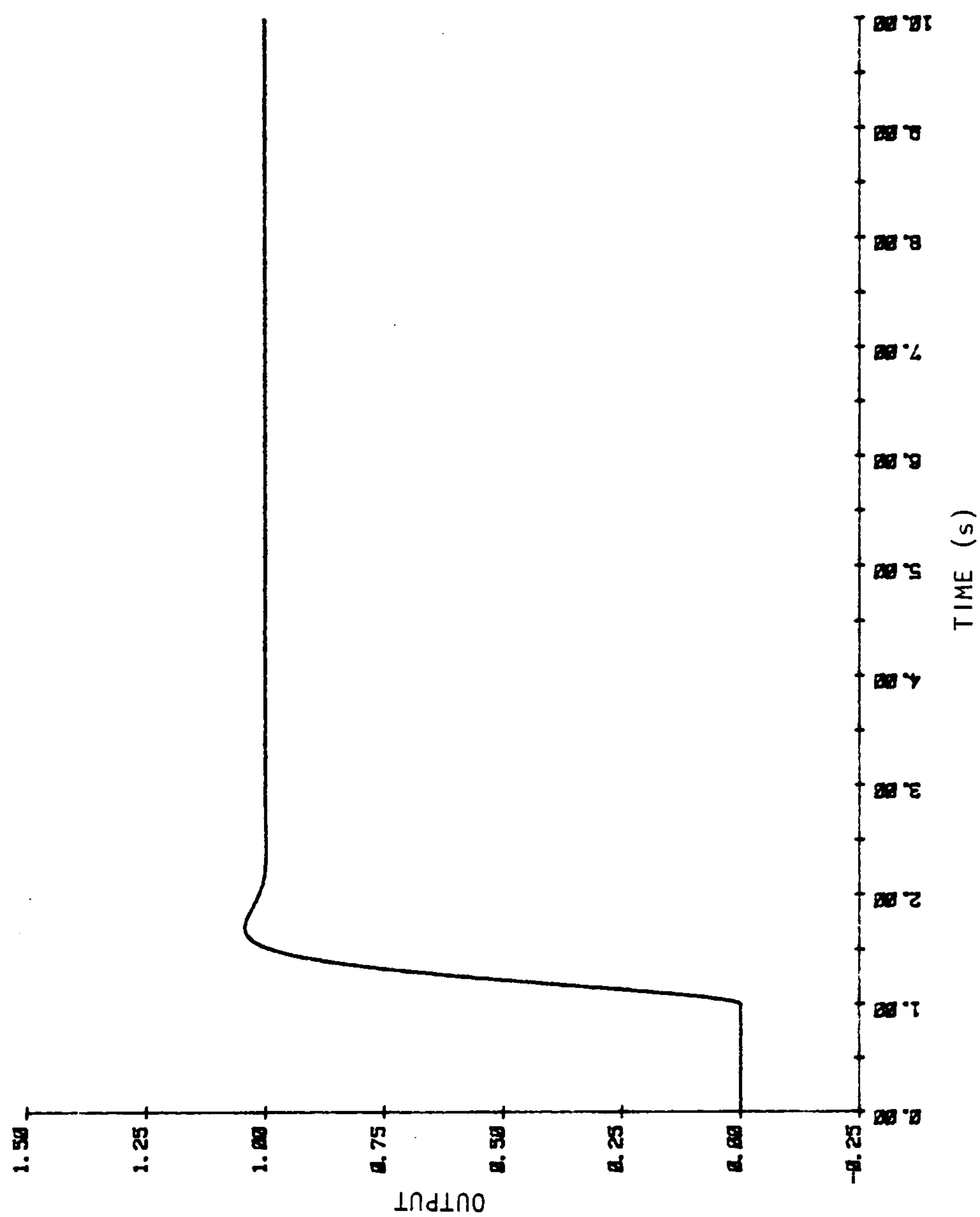


Figure 13. Second-order bilinear filter step response.

the bilinear type offers a better roll-off characteristic (Stanley, 1975). With this in mind, a first-order bilinear low-pass filter with a 3-dB cut-off frequency of 0.1 Hz was selected.

CHAPTER IV

FILTER IMPLEMENTATION AND SYSTEM PERFORMANCE

The implementation of the digital filter was developed on the microcomputer-based data acquisition system. This chapter describes the digital filter implementation and resulting measurement system performance.

Algorithm Implementation

A subroutine approach was selected for implementing the digital filter. This allowed functional implementation of the filter and provided a reusable system function for other measurements, if required. The input to the filter subroutine consisted of a single integer value that represented the raw sampled density data. The filter output was likewise an integer value. All internal calculations were performed in single precision floating-point format. Engineering unit conversion was performed on a demand basis external to the filter subroutine.

The first step in implementing the filter was to convert the discrete form of the algorithm to canonical form. Decomposition was used and, when applied to the coefficients from Table 5, page 29, in Equation (31), yields

$$OUT(Z) = .627924 \times 10^{-3}\, X(Z) + .627924 \times 10^{-3}\, Z^{-1}\, X(Z) \qquad (32)$$

and

$$X(Z) = IN(Z) + .9987441\, Z^{-1}\, X(Z)\ . \qquad (33)$$

Applying Equations (32) and (33) to graphical presentation gives the signal flow graph shown in Figure 14. The actual FORTRAN subroutine source code is shown in Figure B-1 of Appendix B.

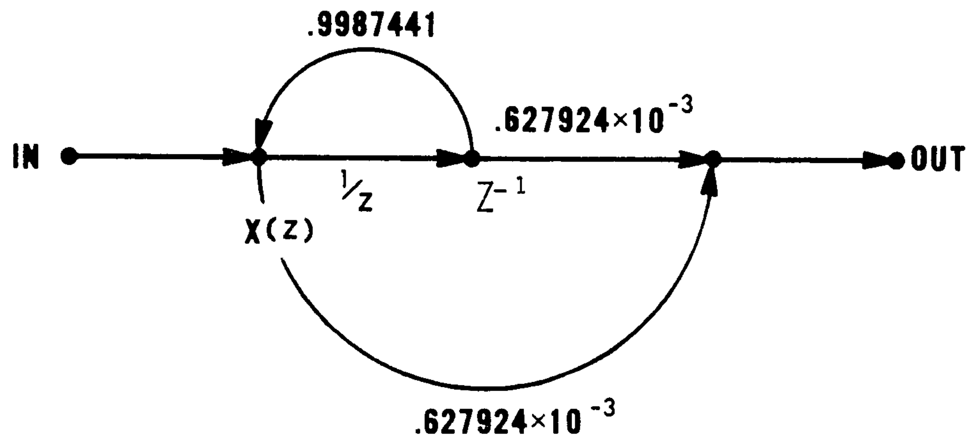


Figure 14. Signal flow graph of noise reduction filter.

Performance Testing

Initial testing of the digital filter subroutine involved filtering the master data file information and graphically plotting the results. Figure 15 shows these results.

An additional test was conducted to determine the system response to a large process variable step function, such as observed in an operating point change. This condition was ideally simulated by causing a fixed bias of approximately 0.25 kg/L in the master data file. Figure 16 shows a plot of the master data file, Figure 17 shows the same data with an idealized step at T = 2 s, and Figure 18 shows the filter response.

Final filter testing involved real-time evaluation of the overall measurement system performance. Since the filter processed information in integer format, an engineering units conversion had to be made for meaningful interpretation of the data. This conversion was peformed and displayed every 2556 samples. This test was executed from the LA-120 printer for permanent hard copy listing of the results as shown in Table 8. The computer program block diagram, listing, and test results are shown in Appendix C.

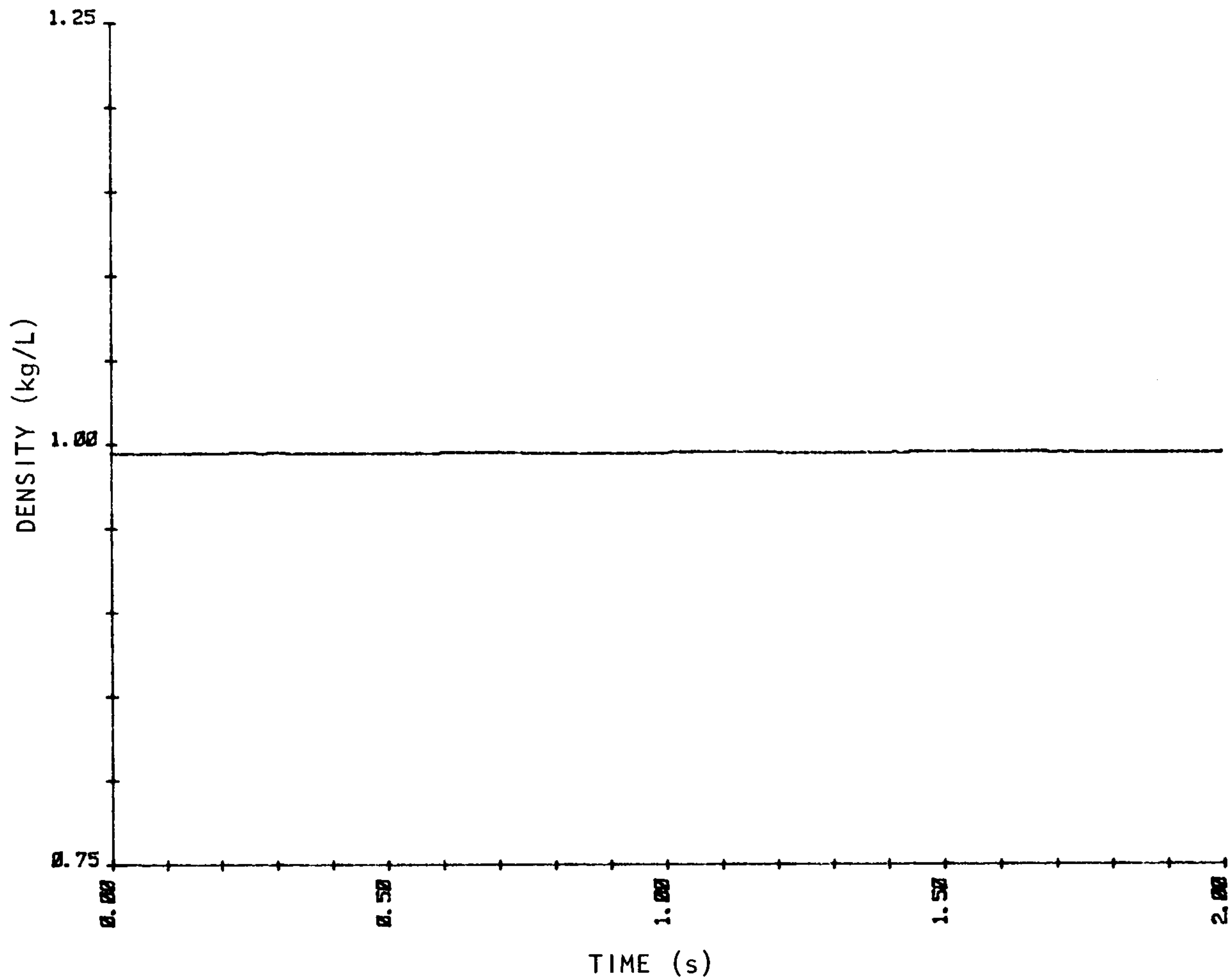


Figure 15. Filtered liquid density measurement data.

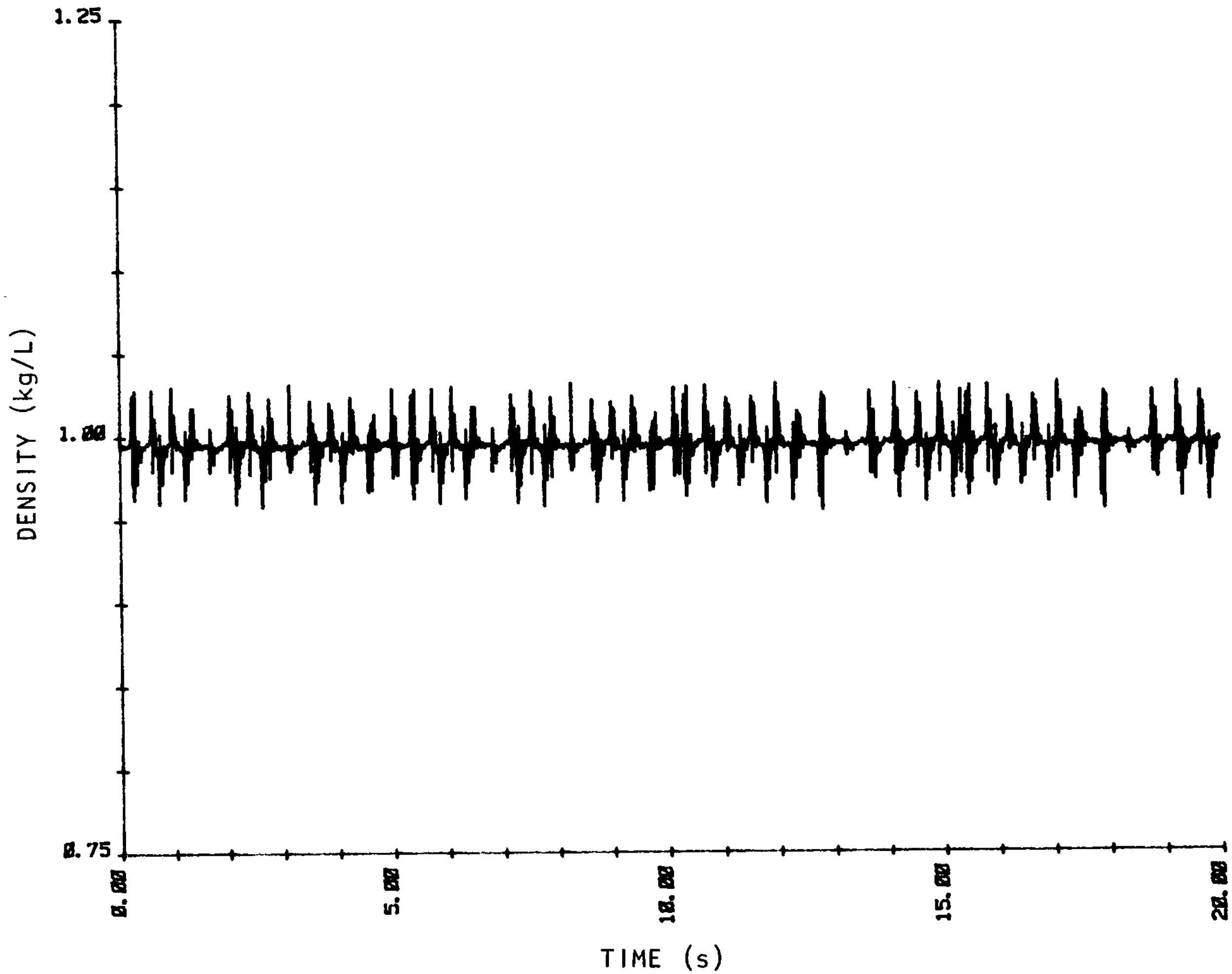


Figure 16. Unfiltered liquid density measurement data.

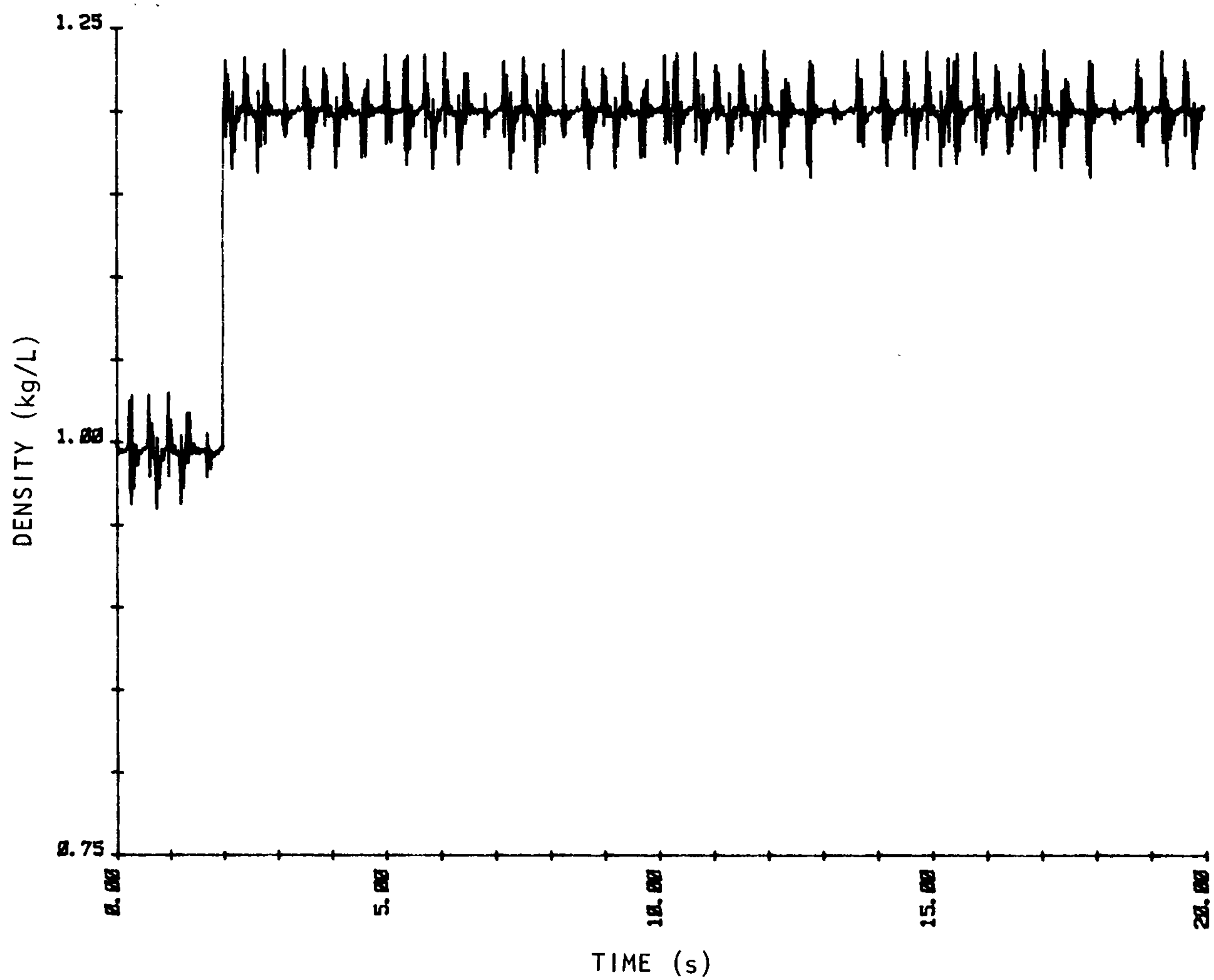


Figure 17. Unfiltered density data with idealized step.

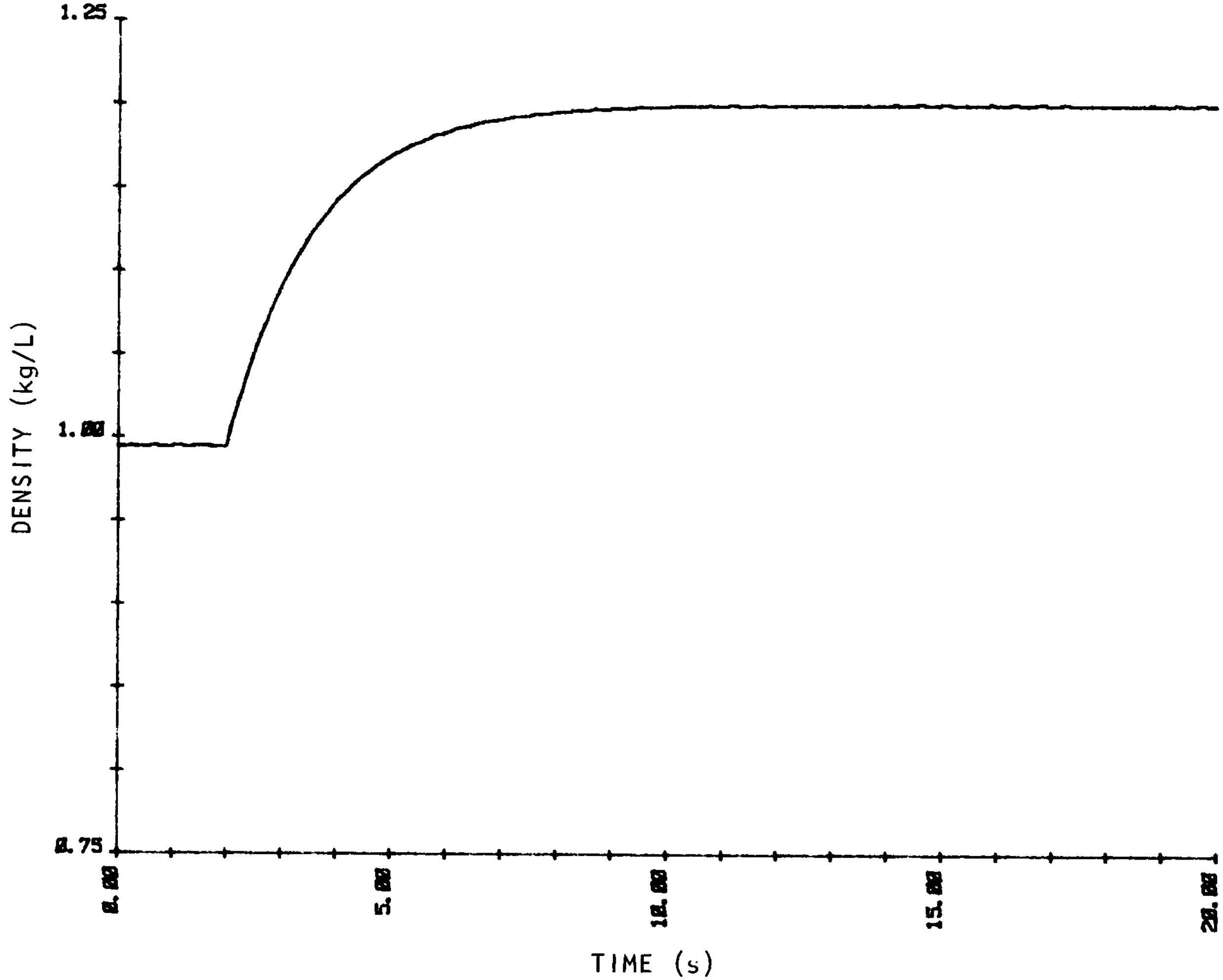


Figure 18. Filter response to unfiltered density data with idealized step.

Table 8. Real-time filter implementation

Raw data:	Filtered data:
0.996953	0.465975
1.008585	0.714358
1.001743	0.847102
1.005848	0.917580
0.998322	0.955214
0.997637	0.975741
0.994900	0.986005
0.973689	0.992163
0.987374	0.995585
1.001059	0.996953
0.995585	0.997637
0.999006	0.998322

CHAPTER V

SUMMARY

This project developed an analog low-pass equivalent digital filter to reduce noise associated with a purged dip-tube density measurement. An experimental measurement system was constructed to aid in filter design and implementation, and the resulting filter design was based on characteristics predicted by a dip-tube mechanism model that was also developed. The filter provided approximately 51 dB of attenuation at the predicted natural resonant frequency of the dip-tube mechanism.

The digital filtering method used here offers a cost-effective approach to reducing noise in digitally based measurement systems. In this case, existing measurement system capacity was utilized and no additional external hardware, such as analog signal conditioners, was required. The noise reduction filter processed raw binary sampled data, which reduced the numeric loading of the microcomputer. Engineering units conversion was performed only when it was required by other tasks. The filter was implemented in subroutine form; as a result, other similar density measurements could utilize the same filter coding.

The design procedures used in this study can be applied in cases where a low-pass equivalent filter function is sufficient for measurement noise reduction or aliasing associated with data processing rate mismatch. The programs developed to calculate the digital filter coefficients are general purpose and can be used in other applications.

These programs offer independent selection of sampling rate and cut-off frequency for both first- and second-order step invariant and bilinear filters.

LIST OF REFERENCES

LIST OF REFERENCES

1. Bradley, N. C., Murray, W. P., Googe, J. M., Errors in Measuring Level and Density with Purged Dip-Tubes, Oak Ridge National Laboratory TL-5995, U.S. Department of Energy, 1977.

2. Buckley, Page S., Techniques of Process Control, New York, New York: Wiley, 1964.

3. Gray, Dwight, E., American Institute of Physics Handbook, New York, New York: McGraw-Hill, Inc., 1972.

4. Irvine, Thomas, F., Jr., Hartnett, James, P., Steam and Air Tapes in SI Units, New York, New York: McGraw-Hill, Inc., 1976.

5. Kuo, Benjamin, C., Digital Control Systems, New York, New York: Holt, Rinehart and Winston, Inc., 1980.

6. Memories and Peripherials, Maynard, Massachusetts: Digital Equipment Corp., 1978.

7. Perry, Robert H., Chilton, Cecil H., Chemical Engineers Handbook, New York, New York: McGraw-Hill, Inc., 1973.

8. Stanley, William D., Digital Signal Processing, Reston, Virginia: Reston Publishing Company, Inc., 1975.

APPENDIXES

APPENDIX A

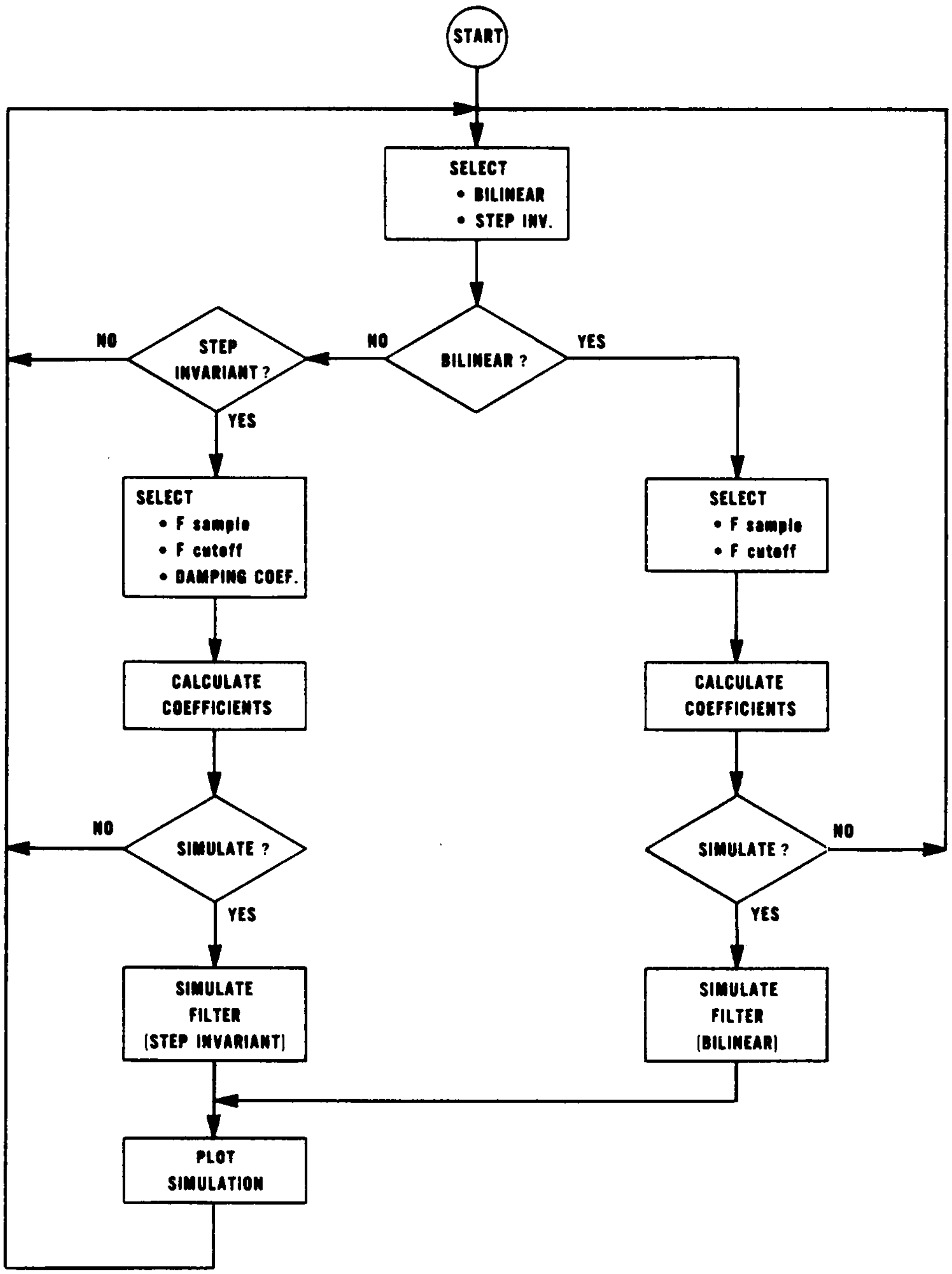


Figure A-1. Second-order filter coefficient generation program block diagram.

```
FORTRAN IV      V02.1-1    Sat 25-Jun-83 12:57:32                 PAGE 001

0001            PROGRAM SIMFIL
0002            COMMON /BLK1/ DATACT(5000)
0003    200     WRITE(5,100)
0004    100     FORMAT(1H ,'        THIS PROGRAM COMPUTES THE COEFFICIENTS FOR',
             #     ' A DIGITAL FILTER'/,
             #     '         EQUIVALENT TO A SECOND ORDER ANALOG FILTER.'//,
             #     '         TWO OPTIONS ARE AVAILABLE:'//,
             #     '                 1)        BILINEAR DERIVATION METHOD'/,
             #     '                 2)        STEP INVARIANT DERIVATION METHOD'//,
             #     '         WHICH METHOD DO YOU WANT? (0 FOR BILINEAR, 1 FOR,',
             #     ' STEP INV.)'//)
0005            READ(5,*) OPTION
0006            IF(OPTION .EQ. 0) CALL BILCOF
0008            IF( OPTION .EQ. 1) CALL GENFIL
0010            IF(OPTION .NE. 1 .AND. OPTION .NE. 0) GO TO 201
0012            GO TO 200
0013    201     WRITE( 5,101)
0014    101     FORMAT( 1H ' ERRONEOUS ENTRY, PLEASE TRY AGAIN..',/)
0015            GO TO 200
0016            END
```

Figure A-2. Listing of second-order filter coefficient generation program.

```
FORTRAN IV      V02.1-1    Sat 25-Jun-83 12:57:42                PAGE 001

0001            SUBROUTINE BILCOF
0002            REAL PI
0003            PI=3.141592654
0004            WRITE(5,100)
0005      100   FORMAT(1H ,' ENTER THE SAMPLING RATE IN HERTZ..')
0006            READ(5,*) SFREQ
0007            WRITE(5,101)
0008      101   FORMAT(1H ,' ENTER THE DESIRED 3db CUTOFF FREQUENCY...')
0009            READ(5,*) CFREQ
0010            W=2*PI*CFREQ
0011            C=1/(SIN(W/2/SFREQ)/COS(W/2/SFREQ))
0012            A0=1.0
0013            A1=0.0
0014            A2=0.0
0015            B0=1.0
0016            B1=SQRT(2.0)
0017            B2=1.0
0018            Q=B0+B1*C+B2*C**2
0019            AA0=(A0+A1*C+A2*C**2)/Q
0020            AA1=(2*A0-2*A2*C**2)/Q
0021            AA2=(A0-A1*C+A2*C**2)/Q
0022            BB1=(2*B0-2*B2*C**2)/Q
0023            BB2=(B0-B1*C+B2*C**2)/Q
0024            WRITE(5,200) AA0,AA1,AA2,BB1,BB2
0025      200   FORMAT(1H ,
               #     ' A=  ',G15.7,/,
               #     ' B=  ',G15.7,/,
               #     ' C=  ',G15.7,/,
               #     ' D=  ',G15.7,/,
               #     ' E=  ',G15.7,/)
0026            CALL GETOPT (OPTION)
0027            IF( OPTION .EQ. 1)CALL BSIMULATE(AA0,AA1,AA2,BB1,BB2)
0029            RETURN
0030            END
```

Figure A-2. (Continued)

```
FORTRAN IV      V02.1-1     Sat 25-Jun-83 12:57:57               PAGE 001

           SUBROUTINE BSIMULATE(A0,A1,A2,B1,B2)
           COMMON /BLK1/ DATACT(5000)
           Z1=0.0
           Z2=0.0
           DO 20 I=1,5000
           IF (I.LT.500) CVTDAT=0
           IF (I.GE.500) CVTDAT=1.0
           CALL BFILTER(CVTDAT,Z1,Z2,A0,A1,A2,B1,B2)
           DATACT(I)=CVTDAT
     20    CONTINUE
           CALL NEWPLT
           WRITE(5,103)
     103   FORMAT(1H ,5(/),' SIMULATION COMPLETE')
           RETURN
           END
```

```
FORTRAN IV      V02.1-1     Sat 25-Jun-83 12:58:07               PAGE 001

           SUBROUTINE BFILTER (X,Z1,Z2,A1,A2,A3,C,D)
           Z=X-C*Z1-D*Z2
           X=A1*Z+A2*Z1+A3*Z2
           Z2=Z1
           Z1=Z
           RETURN
           END
```

Figure A-2. (Continued)

```
FORTRAN IV      V02.1-1    Sat 25-Jun-83 12:58:16               PAGE 001

            SUBROUTINE GENFIL
            REAL PI,N1T1,N1T2
            PI=3.141592654
      301   WRITE(5,100)
      100   FORMAT(1H ,/,' ENTER THE SAMPLING RATE IN HERTZ..')
            READ(5,*) SFREQ
            WRITE(5,101)
      101   FORMAT(1H ,' ENTER THE DESIRED 3db CUTOFF FREQUENCY...')
            READ(5,*) CFREQ
            WRITE(5,102)
      102   FORMAT(1H ,' ENTER THE DESIRED DAMPING COEFFICIENT...')
            READ(5,*) B
            IF(-1.LT.B.AND.B.LT.1) GO TO 300
            WRITE(5,103)
      103   FORMAT(1H ,' DAMPING COEFFICIENT OUT OF RANGE ( -1<B<1 )..')
            GO TO 301
      300   WN=2*PI*CFREQ
            W=WN*SQRT(1-B**2)
            A=B*WN
            AA=EXP(-A/SFREQ)*COS(W/SFREQ)
            CC=EXP(-2*A/SFREQ)
            BB=(B/SQRT(1-B**2))*EXP(-A/SFREQ)*SIN(W/SFREQ)
            RN1=1-AA-BB
            RN2=BB+CC-AA
            D1T1=2*AA
            D2T1=CC
            WRITE(5,200) RN1,RN2,D1T1,D2T1
            CALL GETOPT(OPTION)
            IF(OPTION .EQ. 1)CALL GSIMULATE(RN1,RN2,D1T1,D2T1)
      200   FORMAT(1H ,
          #     'A=   ',G15.7,
          #     'B=   ',G15.7,
          #     'C=   ',G15.7,
          #     'D=   ',G15.7,/)
            RETURN
            END
```

Figure A-2. (Continued)

```
FORTRAN IV      V02.1-1     Sat 25-Jun-83 12:58:31               PAGE 001

            SUBROUTINE GSIMULATE(A1,A2,B1,B2)
            COMMON /BLK1/ DATACT(5000)
            Z1=0.0
            Z2=0.0
            DO 20 I=1,5000
            IF (I.LT.500) CVTDAT=0
            IF (I.GE.500) CVTDAT=1.0
            CALL GFILTER(CVTDAT,Z1,Z2,A1,A2,B1,B2)
            DATACT(I)=CVTDAT
      20    CONTINUE
            CALL NEWPLT
            WRITE(5,103)
      103   FORMAT(1H ,5(/),' SIMULATION COMPLETE')
            RETURN
            END
```

```
FORTRAN IV      V02.1-1     Sat 25-Jun-83 12:58:41               PAGE 001

            SUBROUTINE GFILTER (X,Z1,Z2,A,B,C,D)
            Z=X+C*Z1-D*Z2
            X=A*Z1+B*Z2
            Z2=Z1
            Z1=Z
            RETURN
            END
```

Figure A-2. (Continued)

```
FORTRAN IV      V02.1-1    Sat 25-Jun-83 12:58:50              PAGE 001

              SUBROUTINE GETOPT (OPTION)
              WRITE(5,100)
        100   FORMAT( 1H ,/,' DO YOU WANT TO SIMULATE FILTER
             #     OPERATION? (0 FOR NO, 1 FOR YES)',/)
              READ(5,*) OPTION
              RETURN
              END
```

```
FORTRAN IV      V02.1-1    Sat 25-Jun-83 12:58:59              PAGE 001

              SUBROUTINE SIMOPT(LENGTH,OUTINC)
              WRITE(5,100)
        100   FORMAT(1H ' HOW MANY SAMPLES DO YOU WANT FOR SIMULATION?  ')
              READ(5,*)LENGTH
              WRITE(5,101)
        101   FORMAT(1H ,' HOW MANY SAMPLES TO EACH OUTPUT LIST DO YOU WANT?  ')
              READ(5,*)OUTINC
              RETURN
              END
```

Figure A-2. (Continued)

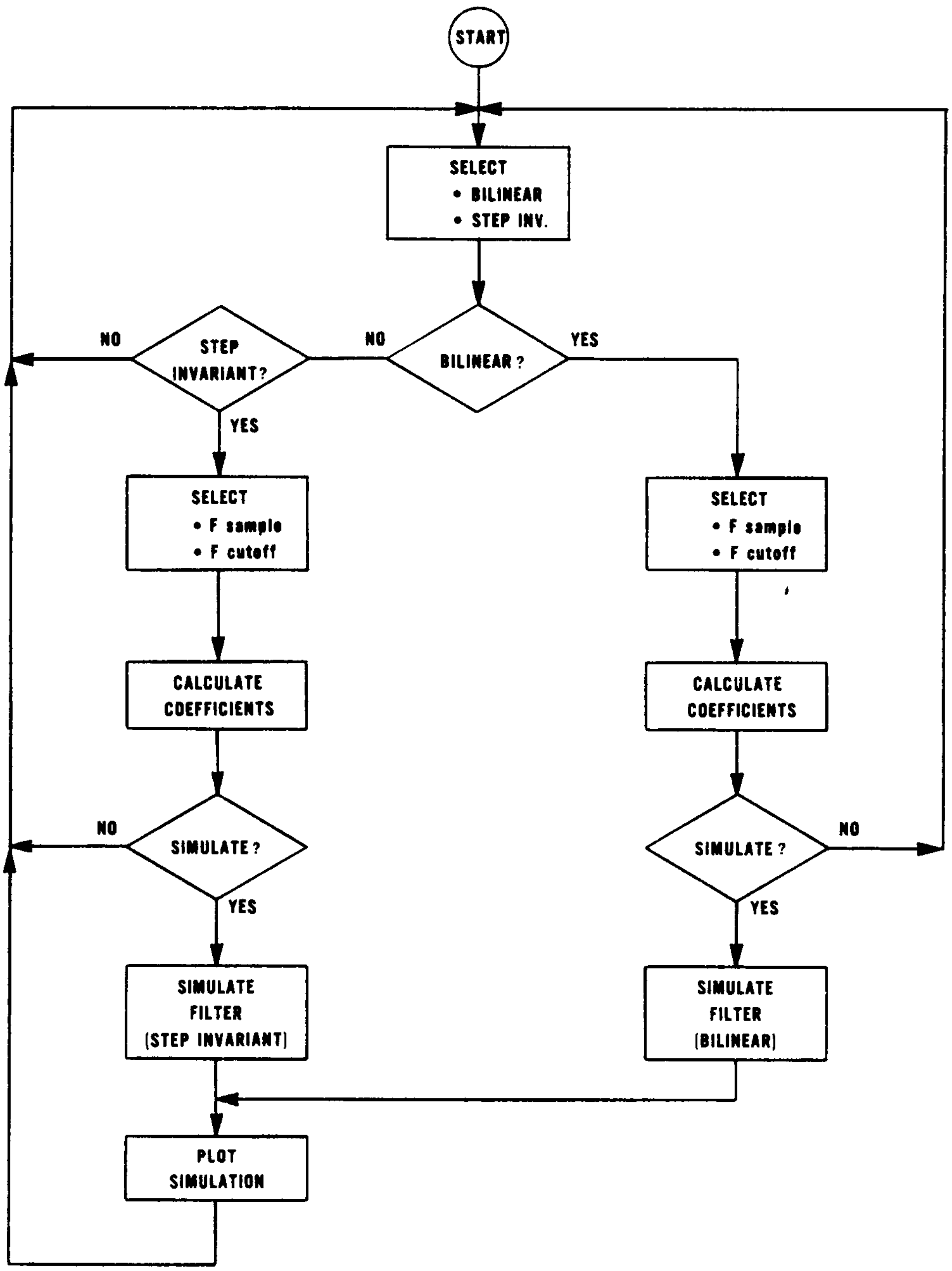


Figure A-3. Block diagram of first-order filter coefficient generation program.

```
FORTRAN IV      V02.1-1    Sat 25-Jun-83 12:55:54              PAGE 001

              PROGRAM SIMFIL
              COMMON /BLK1/ DATACT(5000)
        200   WRITE(5,100)
        100   FORMAT(1X     THIS PROGRAM COMPUTES AND SIMULATES THE',
            #     ' COEFFICIENTS FOR',
            #     ' A DIGITAL FILTER'/,
            #     '       EQUIVALENT TO A FIRST ORDER ANALOG FILTER.'//,
            #     '       TWO OPTIONS ARE AVAILABLE:'//,
            #     '                1)      BILINEAR DERIVATION METHOD'/,
            #     '                2)      STEP INVARIANT DERIVATION METHOD'//,
            #     '       WHICH METHOD DO YOU WANT? (0 FOR BILINEAR, 1 FOR,',
            #     ' STEP INV.)'//)
              READ(5,*) OPTION
              IF(OPTION .EQ. 0) CALL BILCOF
              IF( OPTION .EQ. 1) CALL GENFIL
              IF(OPTION .NE. 1 .AND. OPTION .NE. 0) GO TO 201
              GO TO 200
        201   WRITE( 5,101)
        101   FORMAT( 1H ' ERRONEOUS ENTRY, PLEASE TRY AGAIN..',/)
              GO TO 200
              END
```

Figure A-4. Listing of first-order filter coefficient generation program.

```
FORTRAN IV      V02.1-1    Sat 25-Jun-83 12:56:04            PAGE 001

          SUBROUTINE BILCOF
          REAL PI
          PI=3.141592654
          WRITE(5,100)
    100   FORMAT(1H ,' ENTER THE SAMPLING RATE IN HERTZ..')
          READ(5,*) SFREQ
          WRITE(5,101)
    101   FORMAT(1H ,' ENTER THE DESIRED 3db CUTOFF FREQUENCY...')
          READ(5,*) CFREQ
          W=2*PI*CFREQ
          C=1/(SIN(W/2/SFREQ)/COS(W/2/SFREQ))
          A0=1.0
          A1=0.0
          B0=1.0
          B1=1.0
          Q=B0+B1*C
          AA0=(A0+A1*C)/Q
          AA1=(A0-A1*C)/Q
          BB1=(B0-B1*C)/Q
          WRITE(5,200) AA0,AA1,BB1
    200   FORMAT(1H ,
         #     ' A0=  ',G15.7,/,
         #     ' A1=  ',G15.7,/,
         #     ' B1=  ',G15.7,/)
          CALL GETOPT (OPTION)
          IF( OPTION .EQ. 1)CALL BSIMULATE(AA0,AA1,BB1)
          RETURN
          END
```

Figure A-4. (Continued)

```
FORTRAN IV      V02.1-1     Sat 25-Jun-83 12:56:17                    PAGE 001

            SUBROUTINE BSIMULATE(A0,A1,B1)
            COMMON /BLK1/ DATACT(5000)
            REAL*4 OLD
            Z1=0.0
            DO 20 I=1,5000
            IF (I.LT.500) CVTDAT=0.0
            IF (I.GE.500) CVTDAT=1.0
            CALL BFILTER(CVTDAT,Z1,A0,A1,B1)
            DATACT(I)=CVTDAT
      20    CONTINUE
            CALL NEWPLT
            WRITE(5,103)
      103   FORMAT(1H ,5(/),' SIMULATION COMPLETE')
            RETURN
            END
```

```
FORTRAN IV      V02.1-1     Sat 25-Jun-83 12:56:27                    PAGE 001

            SUBROUTINE BFILTER (X,Z1,A0,A1,B1)
            Z=X-B1*Z1
            X=A0*Z+A1*Z1
            Z1=Z
            RETURN
            END
```

Figure A-4. (Continued)

```
FORTRAN IV      V02.1-1     Sat 25-Jun-83 12:56:36              PAGE 001

              SUBROUTINE GENFIL
              REAL PI
              PI=3.141592654
              WRITE(5,100)
        100   FORMAT(1H ,/,' ENTER THE SAMPLING RATE IN HERTZ..')
              READ(5,*) SFREQ
              WRITE(5,101)
        101   FORMAT(1H ,' ENTER THE DESIRED 3db CUTOFF FREQUENCY...')
              READ(5,*) CFREQ
        300   W=2*PI*CFREQ
              A=EXP(-W/SFREQ)
              WRITE(5,200) W,A
              CALL GETOPT(OPTION)
              IF(OPTION .EQ. 1)CALL GSIMULATE(W,A)
        200   FORMAT(1H ,
             #     'W=   ',G15.7,
             #     'A=   ',G15.7,/)
              RETURN
              END
```

Figure A-4. (Continued)

```
FORTRAN IV      V02.1-1     Sat 25-Jun-83 12:56:47                PAGE 001

0001          SUBROUTINE GSIMULATE(W,A)
0002          COMMON /BLK1/ DATACT(5000)
0003          Z1=0.0
0004          DO 20 I=1,5000
0005          IF (I.LT.500) CVTDAT=0.0
0007          IF (I.GE.500) CVTDAT=1.0
0009          CALL GFILTER(CVTDAT,Z1,W,A)
0010          DATACT(I)=CVTDAT
0011    20    CONTINUE
0012          CALL NEWPLT
0013          WRITE(5,103)
0014    103   FORMAT(1H ,5(/),' SIMULATION COMPLETE')
0015          RETURN
0016          END
```

```
FORTRAN IV      V02.1-1     Sat 25-Jun-83 12:56:57                PAGE 001

0001          SUBROUTINE GFILTER (X,Z1,W,A)
0002          Z=X+A*Z1
0003          X=(1-A)*Z
0004          Z1=Z
0005          RETURN
0006          END
```

Figure A-4. (Continued)

```
FORTRAN IV      V02.1-1    Sat 25-Jun-83 12:57:15                PAGE 001

            SUBROUTINE SIMOPT(LENGTH,OUTINC)
            WRITE(5,100)
      100   FORMAT(1H ' HOW MANY SAMPLES DO YOU WANT FOR SIMULATION?  ')
            READ(5,*)LENGTH
            WRITE(5,101)
      101   FORMAT(1H ,' HOW MANY SAMPLES TO EACH OUTPUT LIST DO YOU WANT?  ')
            READ(5,*)OUTINC
            RETURN
            END
```

```
FORTRAN IV      V02.1-1    Sat 25-Jun-83 12:57:06                PAGE 001

            SUBROUTINE GETOPT (OPTION)
            WRITE(5,100)
      100   FORMAT( 1H ,/,' DO YOU WANT TO SIMULATE AND PLOT FILTER
           #     OPERATION? (0 FOR NO, 1 FOR YES)',/)
            READ(5,*) OPTION
            RETURN
            END
```

Figure A-4. (Continued)

```
FORTRAN IV      V02.1-1     Sat 25-Jun-83 12:59:17                PAGE 001

              SUBROUTINE NEWPLT
              COMMON /BLK1/ DATACT(5000)
              CALL HEADER
              T=2.E-3
              NUMPTS=5000
              CALL SELPEN(3)
              DO 200 J=1,NUMPTS
              X=T*(J-1)
              Y=DATACT(J)
              CALL PLOT(X,Y)
              IF(J.EQ.1)CALL PEN(1)
        200   CONTINUE
              CALL PEN(0)
              CALL SELPEN(0)
              RETURN
              END
```

```
FORTRAN IV      V02.1-1     Sat 25-Jun-83 12:59:27                PAGE 001

              SUBROUTINE HEADER
              INTEGER ANS
              LOGICAL*1 XSTRNG(80),YSTRNG(80)
              DATA XSTRNG/'T','I','M','E',' ',' ',' ','(',' ','S',
             #     'E','C','S',' ',')',67*3/
              DATA YSTRNG/'O','U','T','P','U','T',75*3/
              WRITE(5,300)
        300   FORMAT(1H ,' DO YOU WANT A HEADER DRAWN (1 FOR YES,',
             #     ' 0 FOR NO?')
              READ(5,*) ANS
              CALL SELPEN(1)
              CALL CHDIR(0)
              CALL SCALE(0.,10.0,-.25,1.5)
              IF(ANS.NE.1) GO TO 200
              CALL XAXIS(-.25,.5,0.,10.,2.)
              CALL YAXIS(0.,.25,-.25,1.5,1.)
              CALL PEN(0)
              CALL PLOT(4.0,-.50)
              CALL LABEL(XSTRNG)
              CALL CHDIR(90)
              CALL PLOT(-1.0,.50)
              CALL LABEL(YSTRNG)
        200   CALL PEN(0)
              CALL SELPEN(0)
              RETURN
              END
```

Figure A-5. Listing of plotting subroutines.

APPENDIX B

```
FORTRAN IV      V02.1-1    Sat 25-Jun-83 12:55:36                PAGE 001

        SUBROUTINE FILTER(X,Y)
        COMMON /DELAY/ Z1
        INTEGER X,Y
        DATA A0 /.627924E-3/
        DATA A1 /.627924E-3/
        DATA B1 /-.9987441/
        Z=X-B1*Z1
        Y=INT(A0*Z+A1*Z1+0.5)
        Z1=Z
        RETURN
        END
```

Figure B-1. Listing of noise reduction filter subroutine.

APPENDIX C

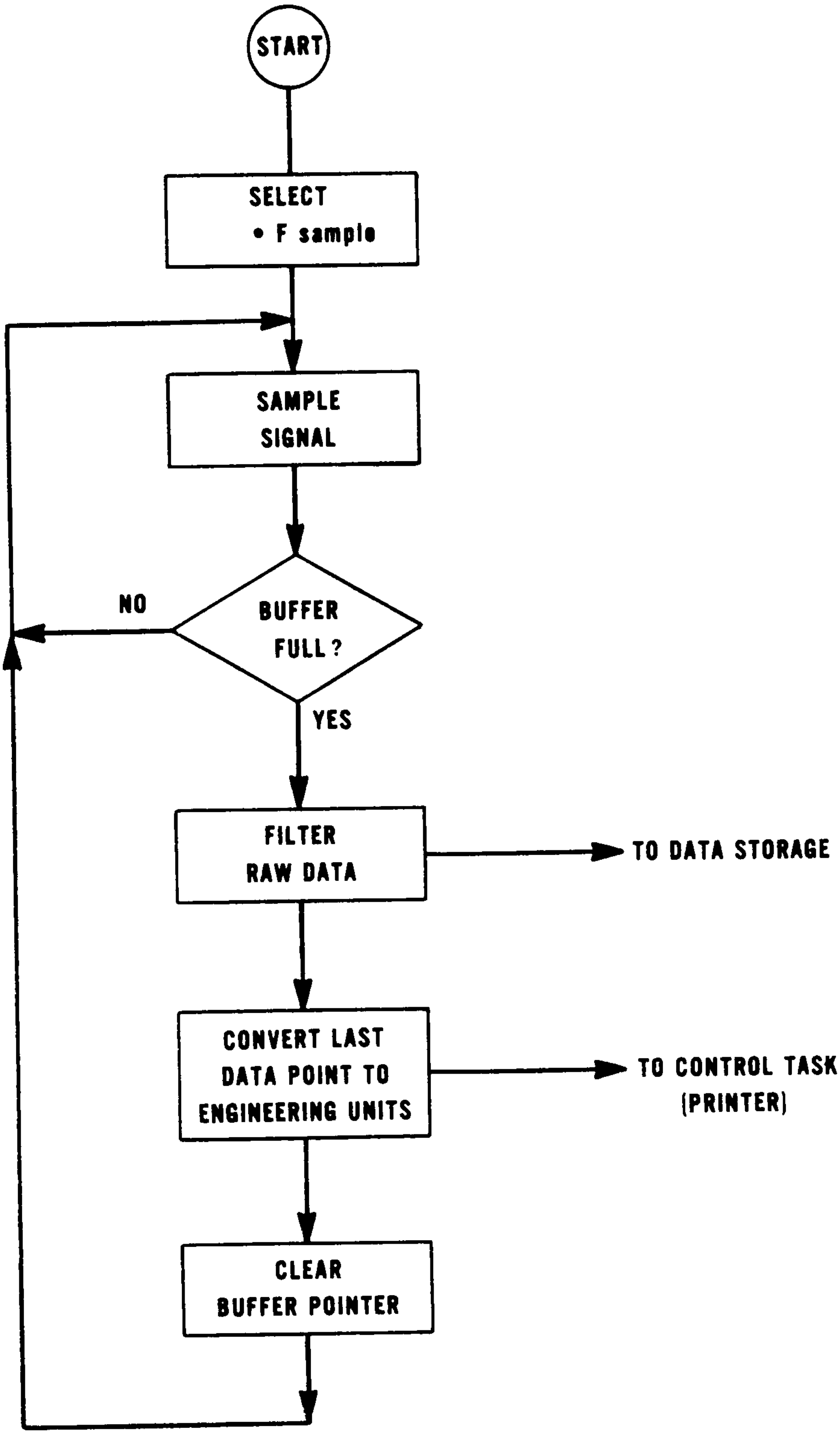


Figure C-1. Block diagram of real-time filter implementation program.

```
FORTRAN IV      V02.1-1     Sat 25-Jun-83 08:52:03              PAGE 001

      C      DATA ACQUISITION PROGRAM TO READ DATA FROM DT2882 ANALOG
      C      INPUT BOARD AND DIGITALLY FILTER DATA FOR NOISE REDUCTION.
      C      WRITTEN BY G. R. WETHERINGTON JR.    JUNE 25,1983
      C
      C
0001         PROGRAM IMPFIL
0002         COMMON /BLOCK1/IBUF(5112),IUNIT,IBEF,ICMF,IBFCT,IBLK,BLKLN,
            *    ISIZ
0003         EXTERNAL DISDAT
0004         EQUIVALENCE (ICMF,IICMF),(IBEF,IIBEF)
0005         DATA IRATE,MODEA,ICMF1/1,9,0/
0006         DATA ISIZ,NBUF,ISTCHN,NCCH,IGAIN/5112,2,0,1,1/
0007         DATA ICMF,IBEF,INTCT,IPRIO,MODEB/0,0,2556,0,2/
0008         DATA NREAD,IBLK/-1,1/
0009         WRITE(5,100)
0010  100    FORMAT(1H ,///,' SAMPLE PROGRAM FOR REAL-TIME A/D OPERATION'//)
0011         WRITE(5,210)
0012  210    FORMAT(1H ,/,' ENTER NUMBER OF MILLISECONDS PER INTERRUPT....')
0013         READ(5,*) PRESET
0014         PRESET=PRESET*1000.0
0015         CALL SETR(IRATE,,MODEA,PRESET,ICMF1)
0016         CALL RTS(IBUF,ISIZ,NBUF,NREAD,ISTCHN,NCCH,,IGAIN,MODEB,
            *    ICMF,IBEF,INTCT,DISDAT,IPRIO)
0017         WRITE(5,110)
0018         WRITE(5,111)
0019  110    FORMAT(1H ,'DATA ACQUISITION IN PROGRESS ....'/)
0020  111    FORMAT(1H ,///,'          RAW DATA:',12X,'FILTERED DATA:',/)
0021  10     IF(ICMF.EQ.-1) WRITE(5,200)
0023  200    FORMAT(1H ,'DATA OVERRUN',/)
0024         IF(IBEF.LE.0) STOP
0026         IF(ICMF.LT.1) GO TO 10
0028         WRITE(5,115)
0029  115    FORMAT(1H ,'DATA ACQUISITION COMPLETE'//)
0030         STOP
0031         END
```

Figure C-2. Listing of real-time filter implementation program.

```
FORTRAN IV      V02.1-1     Sat 25-Jun-83 08:52:29              PAGE 001

            SUBROUTINE WRIOUT(IN,OUT)
            INTEGER OUT
            DATA ANLSB/4.8828125E-3/
            X=(ANLSB*IN*(14.0/10.0))/9.9904
            Y=(ANLSB*OUT*(14.0/10.0))/9.9904
            WRITE(5,500) X,Y
      500   FORMAT(1H ,8X,F12.6,14X,F12.6)
            RETURN
            END
```

```
FORTRAN IV      V02.1-1     Sat 25-Jun-83 08:52:17              PAGE 001

            SUBROUTINE DISDAT
            COMMON /BLOCK1/IBUF(5112),IUNIT,IBEF,ICMF,IBFCT,IBLK,BLKLN,
           *    ISIZ
            COMMON /BLOCK2/ IPNTR
            INTEGER OUT
      20    IF(IBEF.GT.2) GO TO 100
            IF(IBEF.LE.0) GO TO 200
            DO 30 I=5,2560
            IPNTR=IPNTR+1
            CALL FILTER (IBUF(I-4),OUT)
            IF(IPNTR.LT.500) GO TO 30
            IPNTR=0
            CALL WRIOUT(IBUF(I-4),OUT)
      30    CONTINUE
            IBEF=IBEF+1
            IBFCT=IABS(IBFCT-1)
            IF(IBEF.LT.2) GO TO 20
      100   RETURN
      200   WRITE(5,1001)
      1001  FORMAT(1H ,'DATA OVERRUN',/)
            RETURN
            END
```

Figure C-2. (Continued)

```
SAMPLE PROGRAM FOR REAL-TIME A/D OPERATION


ENTER NUMBER OF MILLISECOND PER INTERRUPT....
2
DATA ACQUISITION IN PROGRESS ....
```

RAW DATA:	FILTERED DATA:
0.996953	0.465975
1.008585	0.714358
1.001743	0.847102
1.005848	0.917580
0.998322	0.955214
0.997637	0.975741
0.994900	0.986005
0.973689	0.992163
0.987374	0.995585
1.001059	0.996953
0.995585	0.997637
0.999006	0.998322
1.003796	0.998322
1.001059	0.998322
1.022955	0.999006
1.003796	0.998322
0.996269	0.998322
0.997637	0.998322
0.998322	0.998322
0.994900	0.998322
0.996953	0.999006
0.997637	0.999006
1.000374	0.998322
0.998322	0.998322
1.007901	0.998322
0.998322	0.999006
0.997637	0.999006
0.992163	0.999006
0.996953	0.998322
0.991479	0.999006
1.000374	0.999006
1.025007	0.999006
1.006533	0.998322
0.987374	0.998322
1.000374	0.999006

Figure C-3. Real-time filter implementation program output.

VITA

Grady Randall Wetherington, Jr., was born August 7, 1954, in Jacksonville, Florida. He graduated from Jean Ribault Senior High School and entered the Georgia Institute of Technology in 1972. He graduated with honors in June of 1977, receiving a Bachelor of Electrical Engineering Degree. He entered the Graduate School at The University of Tennessee, Knoxville, in the fall of 1977.

During his undergraduate schooling he worked as a cooperative student with the Georgia Power Company and the Georgia Tech Engineering Experiment Station. Since 1977 he has been employed as a development engineer at the Oak Ridge National Laboratory, which is operated by the Nuclear Division of Union Carbide Corporation for the United States Department of Energy. Responsibilities there have included process systems and instrumentation development, instrumentation and controls applications, and process engineering support. In addition, he has supervised the instrumentation and electrical design efforts of several engineering subcontractors on conceptual and detailed system designs.

He has been married to the former Sarah Cornett since 1979, and they have a son, Eric.