VERTICAL REACTION FORCE RECONSTRUCTION IN HUMAN LOCOMOTION BY DEVELOPING A DIGITAL DATA ACQUISITION SYSTEM AND A HEEL STRIKE ALGORITHM

by

G. Wayne Brodland

A Thesis

Submitted to the Faculty of Graduate Studies

in Partial Fulfillment of the Requirements for the Degree

of Master of Science

The University of Manitoba

Department of Mechanical Engineering

Winnipeg, Manitoba

June, 1982

VERTICAL REACTION FORCE RECONSTRUCTION IN

HUMAN LOCOMOTION BY DEVELOPING A DIGITAL DATA ACQUISITION SYSTEM

AND A HEEL STRIKE ALGORITHM

BY

GERALD WAYNE BRODLAND

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

ABSTRACT

A digital data acquisition, storage and analysis system was developed to process the six reaction force signals induced in a triangular force plate by a subject walking across the force plate. The system made use of a TI 9900 based Technico 9900 microcomputer which was available for the project. An expansion bus with interface cards was designed and built to allow interfacing of the computer to a 16 channel 12 bit sample and hold analogue to digital converter, a digital plotter, and other peripherals. Two floppy disc drives and a printing video terminal are also used.

An algorithm which determines the instant of heel strike from the vertical force records has been developed and is presented. The algorithm is then used to reconstruct the vertical time varying force of particular foot strike patterns (eg. a left foot and right foot striking the plate) by combining the vertical force records of other foot strike patterns (eg. a left foot only striking the force plate, and a right foot only striking the force plate). An extension of this force reconstruction technique allows the center of pressure during double stance to be reconstructed. It also allows the weight transfer during double support to be analyzed. Both reconstruction techniques are especially useful when force plates large enough for multiple successive foot strikes are not available.

The heel strike algorithm proved extremely accurate (within 4 msec) when compared with simultaneous data taken from shoes equipped with heel contact switches. The reconstruction procedures produced records of vertical force and center of pressure which compared very well (typically within 3% and 5% respectively) with measured records of the same types.

ACKNOWLEDGEMENTS

I wish to gratefully acknowledge the guidance provided by my advisor Dr. A.B. Thornton-Trump. I also wish to thank Dr. S. Balakrishnan for his many hours of help with the mechanical equipment, and for many valuable discussions we have had together. Thanks is also due to my brother Dale, to Jack Sill and to several other technicians for their help with the electronic equipment; to Mr. Allan Lee for his work on the data acquisition program; and to the many who have acted as subjects for the experiments.

I wish to thank my wife, Heather, for her encouragement and understanding, especially during the many and often late nights I have been working on this project.

Finally, I wish to gratefully acknowledge the financial support of the National Science and Engineering Research Council without whose help my graduate studies and this thesis would not have been possible.

DEDICATION

This thesis is dedicated to the Glory of God and in honor of my parents, Mr. and Mrs. Alfred Brodland, who taught me both to love God and to love knowledge.

TABLE OF CONTENTS

Table of Contents (Continued)

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A to D	Analogue to Digital converter
A_1	Anterior-posterior reaction force
C1-C4	Circuit capacitors (in microfarads)
CPU	Central processing unit
D_{LR}, D_{RL}	Period between heel strikes
dB	Decibels
DC	Direct current; used to refer to a fixed voltage offset
Disc	A non-volatile magnetic storage device
F_c	Filter cutoff frequency (in Hz)
F_x, F_y, F_z	Cartesian components of force
$F_L(t)$, $F_R(t)$, $F_{LR}(t)$, $F_{LRL}(t)$	Vertical force as a function of time
G	Gain of amplifier stage
Hz	Hertz
LS	Low power Schottky type integrated circuit
msec	Millisecond
M_x, M_y, M_z	Cartesian components of moment
M_1, M_2	Medio-lateral reaction forces
N	Newton
Nm	Newton meter
P	Gait cycle period
Q_1, Q_2	Time scaling factors
R1-R23	Circuit resistors (in ohms)
RAM	Random access (computer) memory
RL ratio	Ratio of the period from right heel strike to left heel strike and the period between successive heel strikes of the same foot

Nomenclature (Continued)

S1-S6	Circuit switches
S_1-S_3	Instants of heel strike
t	Time variable (in sec)
TI	Texas Instruments Incorporated (Registered Trade Mark)
TI 9900	TI type 9900 central processing unit
TTL	Transistor-transistor logic family integrated circuits
V_1 - V_3	Vertical reaction forces
V_7	Input voltage to conditioning amplifier channel 7
V+	Positive supply voltage (+15 volts)
V-	Negative supply voltage (-15 volts)
X	Space coordinate in direction of walking
Y	Space coordinate directed to the right from the direction of walking
Z	Space coordinate directed downwards
&	Denotes a hexadecimal number or memory location
*	Denotes multiplication
741	Type 741 operational amplifier

CHAPTER 1

Introduction

1.1 Gait Studies in Perspective

Locomotion in the form of walking or running is one of the most fundamental of human activities. Studies of the basic mechanism of human gait were recorded as early as the 17th century [1]. Investigations had to be made on the basis of what could be observed directly. Inadequate instrumentation made systematic studies almost impossible. The flourishing of science and invention in the late 19th and early 20th centuries resulted in the development of many ingenious devices to measure foot contact forces and body segment displacements [2-7]. These were usually crude devices which presented only an approximate measure of gait parameters. Analysis of the results was a very long and tedious process. The advent of cinematography made the first accurate displacement measurements possible in 1934 [8]. Electronic instrumentation to measure forces allowed the first accurate force records to be taken in 1952 [9]. Systematic studies became possible at this time, and resulted in such classic descriptions of gait such as that of Murray [10] (1967). The advent of computers to do the calculations and digital data acquisition systems to automatically record measurements has made systematic analysis far easier and has made more complex analyses possible.

Modern gait studies have demonstrated the value of force and displacement records in evaluating normal and abnormal gait [10-16]. Since automatic displacement measuring equipment is considerably more expensive than similar force measuring equipment, much current research has been devoted to force record studies. There are strong indications that force records contain sufficient information to adequately assess the subjects posture, general stability [11], and gait [12]; and that displacement records may not be necessary for most clinical evaluations [13]. Consequently, force plates will probably become a standard piece of equipment in hospitals and rehabilitation centers.

1.2 Problem Statement

While a good triangular force plate was available for the project, a satisfactory method for analyzing data collected using the plate was not. Data had been conditioned by analogue means and stored on magnetic tape. Data analysis was therefore slow and tedious. A digital data acquisition, storage and processing system is necessary to enable the more complex analyses which are proving useful to be done, and to make the system clinically practical.

Films of the subject's gait were necessary to time the force records with the patient's gait cycle. Analysis of the films is also very time consuming. If an algorithm can be developed to determine a particular instant (eg. heel strike) in the gait cycle from the force records and perhaps walkway contact sensors, filming will not be necesary for routine patient studies.

Most hospitals and rehabilitation centers only have access to the small force plates which are commercially available, and which are large enough to measure the forces of a single foot only. Consequently they do not allow studies of double support. Methods to reconstruct the vertical force which would be produced from multiple foot falls, and which would allow the motion of the center of pressure to be reconstructed from single foot force records would allow studies of double support. Current research [14,15,17,18] has shown the value of such studies in assessing both normal and abnormal gait. A clinical system to perform these functions is not presently available but would be a great aid in clinical assessment of a patient's gait during such times as therapeutic treatment, natural recovery, and prosthesis fitting and evaluation.

1.3 Statement of Objectives

The objectives of this project are:

(1) To develop a digital computer system which will acquire, store and process gait force data. The data acquisition system must produce minimal interchannel time skew.

(2) To develop an algorithm which will determine the instants of heel strikes from the plate reaction forces and from foot contact with force sensitive areas of the walkway.

(3) To use the heel strike algorithm to reconstruct the vertical time varying force of particular foot strike patterns by combining the vertical force records of other foot strike patterns.

(4) To reconstruct the motion of the center of pressure during double support using the heel strike algorithm, and the vertical force records of separate left and right feet.

(5) To work towards a system suitable for clinical use.

CHAPTER 2

Literature Survey

2.1 Introduction

The purpose of this chapter is to review the main literature which deals with gait force measurement and with the clinical usefulness of gait force and center of pressure records. The extent of the literature and the number of recent articles indicate much current interest in these subjects.

2.2 Force Measurement

The first reported attempts to measure foot reaction forces were made by the French researchers Carlet [2] and Marey [3] in 1872. They used a pneumatic cell attached to the sole of the subject's shoe to measure his vertical floor reaction. Three ingenious techniques for measuring foot reaction forces were later developed by German researchers. In 1882, Beely [4] had his subjects walk across a sack of plaster of paris. This produced a permanent footprint which actually provided more information about the foot shape than the foot reaction forces. A technique which measured forces was developed by Abrahamson [5] in 1925. A hard surface was covered with steel shot and then with a thin sheet of lead. The shot indentations in the lead sheet produced by the subject's traverse of the sheet were taken as proportional to the local maximum load. In 1926, Basler [6] developed a 10 string harp on which the subject stood. By plucking the strings, the string tension and hence the local contact pressure could be determined.

Time records of forces became possible with the advent of cinematography. In 1934 Elftman [8] filmed the deflections of a corrugated rubber mat from the underside of the glass plate on which it rested as a subject traversed the mat. The mat deflections were taken to represent the applied load. In 1954, Barnett [19] filmed the deflection of vertical perspex rods into a rubber support as the subject walked across the top of the rods.

The first force plates capable of measuring all three floor reaction components were reported by Amar [7] in 1916 and by Elftman [20] in 1938. While Elftman's rectangular plate measured all three reaction components and was the most accurate device available at the time, its large deflections, together with the inertia and friction of the linkages produced large errors in the measured forces. Also, no permanent force record was produced.

Modern foot reaction force measurements are carried out using either force plates or instrumented shoes. Force plates are generally easier to use as the subject need not be encumbered with special footwear, instrumentation and an umbilical cord. He can walk comfortably in his own shoes. Force plates are also generally less expensive and easier to use as one plate will fit all subjects. Also, the instrumentation need not be miniaturized so that it will fit inside a shoe. These advantages are largely responsible for the current popularity of force plates. Modern force plates are normally interfaced to analogue or digital data acquisition systems which among other things, produce a permanent force-time record. Some plates such as the triangular plate of Yamashita and Katoh [15] measure only vertical forces. Others, such as the rectangular plates developed by Cunningham and Brown [9] and Grundy et al. [14], and the triangular force plate developed by Balakrishnan and Thornton-Trump [21] allow vertical and horizontal forces to be measured. These plates act as rigid bodies and only allow the total foot force and center of pressure to be measured. Grundy et al. [14] have developed a plate which incorporates a glass window covered with a thin oil film. The glass plate and oil film allow the instantaneous foot contact areas to be filmed from beneath. The contact areas are then correlated with the measured forces to provide a more complete picture of the subject's gait. In 1976, Scranton and McMaster [22] made use of a liquid crystal whose colour is pressure dependent. The liquid crystal material is incorporated into a sheet which is spread on a supporting plexiglass sheet. The sheet is filmed from beneath as the subject walks on it. The liquid crystal sheet colour is then decoded to determine instantaneous contact pressures.

The primary advantages of force measuring footware over force plates are that the former do not necessarily require the subject to step on a certain part of the floor or walkway and that the number of sequential steps measured is not limited to a few at most by the force plate size. A great variety of footware has been developed. Shoes developed by Spolek [23] have strain gauge load cells in the heel and sole to measure all contact forces and moments exerted at these two areas. A more detailed description of foot reactions is available through shoes having a matrix of piezoelectric transducers incorporated into them such as developed by Hennig [24]. These however, only measure vertical force, and require very expensive instrumentation.

Studies abound on the use of force records produced using instrumented shoes [23,24], and especially those produced using force plates [14-18,21,22,25,26]. These works have corporately demonstrated the value of force records in gait study, and in evaluation of prosthetic devices and therapy. Many of these studies are not detailed here as they provide only a general basis for this work.

2.3 Center of Pressure

Several studies have been done to determine the center of pressure from force plate reactions. Studies of this type have been done by Yamashita and Katoh [15], Cunningham [17], Murray et al. [11,18] and Grundy et al. [14]. Center of pressure is used by these researchers to describe normal and abnormal gait. Murray et al. [11] also used the center of pressure to evaluate postural stability and steadiness. Grundy et al. [14] have found center of pressure useful in studying weight transfer during normal and abnormal gait. They are also using it to study barefoot walking and the effect of footwear on gait. In another investigation [18], center of gravity and center of pressure are studied during such activities as rising from a seated position and jumping. It has been suggested by Cavanagh [27], that special techniques using vertical force and resultant moments must be used if center of pressure paths of several subjects are to be averaged together.

The literature does not show that any work to date has used center of pressure calculations to identify the instant of heel strike. Nor is there any indication that others have reconstructed the force record of particular foot fall patterns from force records of other foot fall patterns. For example, reconstruction of the force records produced by successive left and right foot strikes on a force plate from a left foot force record (ie. where only the left foot struck the plate) and a right foot force record.

CHAPTER 3

Data Acquisition System

3.1 Introduction

Analogue data acquisition and processing is expensive. In this study at least four channels of information must be recorded and seven channels are preferred. Further processing to determine the centre of pressure requires considerable expensive analogue hardware. The alternative is to develop a digital data acquisition and processing system.

The first objective of this project was thus to develop a system to measure, store, and process gait floor reaction forces. A block diagram of the system which was built is shown in Figure 3.1. The system has three main parts: the force plate which measures the gait reaction forces, and its approaches; analogue components which provide scaled voltage signals proportional to these forces; and digital components which sample, store and process these signals. The author built or assembled all of the equipment shown except for the force plate and strain bridge amplifiers, which were built by Dr. S. Balakrishnan.

3.2 The Walkway and Force Plate

The 1.22 m x 2.44 m triangular force plate shown in Figure 3.2 was instrumented to measure the six corner reactions shown. These reactions provided sufficient information to calculate all three force components Fx, Fy and Fz and all three moment components, Mx, My and Mz exerted on the plate by the patient. The details of the plate and its transducers are given by Balakrishnan [13].

To allow the subject to traverse the force plate with his normal steady state gait pattern, 1.22 m wide by 5.64 m long approaches were provided leading up to and away from the force plate. To provide additional heel strike information, the section of each approach immediately adjacent to the force plate was made force sensitive. This was done by hinging these sections as shown in Figure 3.2. Springs were provided to raise the force plate edge slightly above its rigid support. When the subject exerted a moment of more than approximately 6 Nm about the hinge axis (a force of about 20 N at the center of the hinged section) the springs were compressed sufficiently to open a microswitch and thereafter make the hinged section rest on its rigid support. This deflection was very small and did not interfere with the gait of the subject.

3.3 Analogue Equipment

3.3.1 Strain Bridge Amplifiers

Since the force plate is mounted on steel transducer elements, the strain of these elements will be linearly proportional to the plate reaction forces. The transducer strains are converted in turn to voltages, all of which have linear dependence on the strains, by six strain bridge amplifiers; one for each transducer. The three transducers to measure the vertical reactions V_1, V_2 and V_3 were wired as full bridge systems having two active and two temperature compensating gauges mounted on the transducer.

The three horizontal transducers which measure the medio-lateral reactions M_1, M_2 and the anterior-posterior reaction A_1 were full bridge systems having two active gauges mounted on the transducer and two compensating gauges mounted in the amplifier rack. All six amplifiers were calibrated to have a zero offset and a gain of 44.5 mV/N. This bridge arrangement was found to be quite satisfactory. All channels had a full range linearity within $\pm 1/2\%$. Unfortunately, the use of 741 type op-amps on the input causes a slight amount of drift and very low frequency noise. Since neither of these factors affected the amplifier gain or linearity, it was possible to overcome these problems digitally as outlined in Section 5.3.

3.3.2 Switch Conditioning

Section 3.2 describes how an area of each approach immediately adjacent to the force plate is equipped with a microswitch which opens when more than approximately 20 N is exerted at its center. Since the data acquisition program is designed to sample a maximum of seven channels, and there are six independent force plate signals, only one channel is left to record switch positions. The circuit shown in Figure 3.3 produces four distinct voltages dependent on the four possible binary states of the two approach switches. The resulting voltages are shown in Table 3.1. These voltage levels are easily decoded to determine the switch positions. The kill switch shown in Figure 3.3 is used to ground the trigger channel input and prevent normal triggering.

3.3.3 Conditioning Amplifiers

The data acquisition system is designed to be as versatile as practical. It can be configured as shown in Figure 3.1 for real time gait data acquisition. It can also be used to digitize and analyse data from other sources such as a tape recorder.

The conditioning amplifiers are designed to condition signals so that they are suitable for conversion to digital form at the analogue to digital conversion (A to D) unit. Each of the eight amplifier channels is composed of three stages: a variable gain non-inverting amplifier, a second order low pass filter and a unity gain inverter, as shown in Figure 3.4. Switches S3 to S6 allow the filter and inverter stages to be independently switched in and out of the circuit. Further, if switches S5 and S6 are both open, the BNC connector which is usually used to monitor the output can be used to input signals to the A to D converter through the ribbon cable. This feature has proved useful for input of such signals as that from a pair of instrumented shoes used to check the heel strike algorithm (See Section 6.3). The printed circuit board was designed as shown in Figure 3.5 so that all components could be mounted in DIP sockets to allow components to be changed easily. This feature proved to be useful on several occasions during circuit development and caused no circuit reliability problems.

The first stage of each amplifier channel is a variable gain non-inverting amplifier. This configuration allows a high (1 Megohm) input impedance. Switch S1 is provided to isolate the coarse and fine DC offset circuit R7 to R10 thereby allowing the offset null R5 to be set. The coarse DC offset R8 is provided to eliminate large offsets on incoming signals and/or to offset the output by as much as several volts if necessary. The fine DC offset R10 is operated from the front panel and is used to compensate for strain bridge amplifier drift and for fine DC offset adjustment in general. The gain potentiometer R6, also operated from the front panel, sets the stage gain to 1 + R6/(R1 + R2). With the resistors shown, gains from unity to about 50 are possible.

When data are transferred from a tape recorder, a considerable amount of high frequency noise (50 KHz) is observed. This seems to be tape bias. The second order low pass filter is incorporated to remove this noise. The circuit component values are determined from the chosen cut-off frequency F_c by

$$R11 = R12 = R$$

$$R14 = 2\ R$$

$$C2 = \frac{0.707}{6.28\ F_c\ R}$$

$$C1 = 2\ C2 \qquad [28]$$

The values shown in Figure 3.4 are chosen to make the cut-off frequency approximately 110 Hz, and to give a roll-off of 40dB/decade above this frequency. The design proved very effective. The general configuration allows two different cut-off frequencies to be used if necessary for other applications.

The final stage of the system shown in Figure 3.4 is an inverting amplifier, whose gain given by G = R16/R15, has been set to unity. A DC offset circuit with isolation switch S2 has been provided for but not installed as it was not needed for the present work. This stage is useful for inverting reversed polarity tape recorded signals.

The amplifier was used in most of its possible configurations with gains of 4, 20 and 30. It maintained output drifts of less than $\pm$ 5 mV and gain variations of less than $\pm$.05% from day to day when it was properly warmed up, and hence, only required occasional recalibration.

3.4 Digital Equipment

3.4.1 Introduction

The digital system is designed around a TI 9900 based Technico 9900 micro-computer. The system was chosen on the basis of its low cost and 16 bit CPU. Also, this system was one of the few 16 bit systems available at the time it was acquired. This turned out to be an unfortunate choice, made before the author became involved in the project. The two main problems with the system are its very poor reliability which is a consequence of its marginal hardware design and its severe lack of software support since the CPU is non-standard. These and other lesser unnecessary problems have been the cause of hundreds of hours of frustrating extra work to the author. After several years of work these problems have been surmounted and a flexible data acquisition system has been built.

The computer bus is non-standard, several years obsolete and already overloaded. Hence, an eight slot bus expansion with power supply in accordance with the new TI bus structure was fabricated. This allows any of the several boards designed for this bus to be used. Since the fanout of certain bus driving integrated circuits in the main computer is already excessive, a high impedence bi-directional interface is used to drive the expansion bus. The main custom digital components, namely the bus interface and the plotter interface are detailed in the following subsections. They have both proved to be very reliable.

3.4.2 Expansion Bus Interface

The expansion bus is interfaced to the main computer with two custom interface cards, one of which plugs into the 122 pin non-standard computer bus. A block diagram of this interface is shown in Figure 3.6. The card loads the computer bus with at most 2 additional LS TTL gates. The expansion bus is designed primarily for memory mapped devices. Hence the address decoder is designed to decode the six most significant address lines and read data from the expansion bus when addresses &E400 to &E7FF are read by the CPU. These buffers drive a four foot long ribbon cable which goes to an electronically identical card which accepts the ribbon cable and plugs into the expansion bus. In this way, buffers are provided on both ends of the cable to reduce noise and guarantee reliable operation. The interrupt lines are connected without buffering though they are not used in the present system.

3.4.3 Plotter Interface

The plotter interface shown in Figure 3.7 is quite similar in structure to the bus interfaces. The plotter is mapped to memory location &E7FF, and hence its address decoder generates a high device select signal only when &E7FF is addressed. This signal together with the Write Enable and Data Bus In control signals are used by the decoder to produce read and write signals for the buffers. This configuration allows writing 16 bit instructions to the plotter and reading the plotter status bits.

CHAPTER 4

Analysis

4.1 Introduction

Like most things when considered in retrospect, the analysis presented here may seem straightforward and perhaps even obvious. However, the procedure is new, and many of the conclusions which are drawn from it are new, far reaching and of considerable practical importance.

There was a need for a uniform time base which could be used to compare force records. A single subject, for instance, may walk at slightly different speeds on several successive traverses of the force plate and he will almost certainly vary his gait speed from one day to the next. A uniform time base allows the effect of speed variations to be almost completely eliminated, and gives a proper base for calculating the harmonic frequency components of gait.

The dynamic forces caused by acceleration of body segments are proportional to the second time derivative of their position. Hence, these dynamic forces should be scaled by the inverse square of the time scale. It has been shown [12,13] that these dynamic forces usually constitute only 10% of the total vertical reaction; the main vertical component being due to quasi-static body support forces. For example, if the time scale is expanded by 10%, the dynamic forces are reduced by 21%, and the total force is altered by approximately 2%. Hence, unless very large dynamic forces are present or large time scale adjustments are made, little error is introduced when the force amplitude is not adjusted. Such a time base together with a force scaling based on the subject's weight provide a means of comparing the gait forces of one person to another since in general they walk at different speeds and have different weights. That the force scale factor should be the inverse of the subject's weight is obvious. It is also easy to calculate.

It is more difficult to choose and calculate the time scale factor. The heel strike is the most easily and accurately measured instant in the gait cycle as outlined in Section 4.3. Since the period between left and right heel strikes and right and left heel strikes is usually different during abnormal gait, the period which should be chosen is from one heel strike to the next heel strike of the same foot. The ratio of the period from right heel strike to left heel strike and the period between successive heel strikes of the same foot is defined as the RL ratio. It is useful for time scaling abnormal gait when a full gait cycle is not available. Identification of these heel strikes from force records is discussed in the next section.

Solution of the heel strike problem allows the transfer of weight from one foot to the other to be determined completely. It also allows the reconstruction of any foot force pattern from other foot force patterns. For example, the force produced by a sequence of left and then right feet hitting the plate (a left-right force record) can be reconstructed by adding the force records of a single left foot on the plate (a left force record) and a single right foot on the plate (a right force record). In a similar fashion, a left force record can be reconstructed by subtraction of a right force record from either a left-right force record or a right-left force record. These procedures are discussed in the following sections, and are applicable both to normal and abnormal gait.

4.2 Weight Transfer

During single support, the center of force of the foot moves from the heel position to the toe position. During double support the effective center of vertical force moves from this toe position to the heel position of the other foot. This double support weight transfer is modelled by the left foot force, $F_L(t)$, applied at the left toe position and the right foot force, $F_R(t)$, applied at the right heel position as shown in Figure 4.1(b). This approach will model the weight transfer in both the direction of walking (x) and normal to it (y). Only the center of force in the direction of walking is shown.

4.3 Identification of Heel Strikes

The total vertical force exerted on the force plate is simply the algebraic sum of the vertical forces exerted by the left and right feet separately. Figure 4.1(a) shows the vertical forces exerted by the left (solid line) and right (dashed line) feet of a normal subject. It also shows the sum of a left and right record (shown with small circles), which is the force produced by a left-right sequence. The first heel strike, S_1, shown in Figure 4.1(a) occurs at the beginning of this composite record and hence is easily found.

Heel strikes occuring when both feet are on the plate, such as S_2, are more difficult to identify. Fortunately, the force exerted by one foot starts very suddenly and increases almost linearly until near the first peak. The transfer of weight to the forward foot causes the center of force to begin moving suddenly from the toe of the rear foot forward towards the forward foot heel position. This produces a sharp change of slope in the center of force curve, as indicated by point A in Figure 4.1(b). Center of force curves of this shape are also reported by Yamashita and Katoh [15]. These sudden changes in slope in the center of force curve allow heel strikes occuring when both feet are on the force plate to be identified. When more than two successive steps hit the force plate, this algorithm is repeated as required. The instant of toe off is not as easily identified by this method as the foot force decreases smoothly to zero.

The first heel strike off the force plate is found using the approach switch channel shown in Figure 4.1(c) which is decoded to determine the approach switch positions. For the case shown in Figure 4.1, the third heel strike S_3 is found by the voltage step which indicates closure of the right approach switch.

4.4 Force Reconstruction

In steady state, the vertical forces exerted by left and right feet are periodic time functions. The force plate record is the algebraic sum of a certain sampling of these functions. It follows, therefore, that certain records can be reconstructed from other records. For example, a left-right vertical force record can be reconstructed from separate left and right records. Both single step records must be scaled to the same gait cycle period as the composite step using:

Gait cycle period, P = period between succcessive heel strikes of the same foot

or

$$P = \frac{1}{\text{RL ratio}} * \text{period between successive right and left heel strikes}$$

or

$$P = \frac{1}{(1\text{-RL ratio})} * \text{period between successive left and right heel strikes}$$

The right force record must also be time shifted by the time scaled period between successive left and right heel strikes. Mathematically, this can be written as:

$$F_{LR}(t) = F_L\ (Q_1t) + F_R\ (Q_2t - Q_2D_{LR})$$
$$= F_L\ (Q_1t) + F_R\ (Q_2\ (t-D_{LR}))$$

where: $F_{LR}(t)$ is the left-right vertical force time function

$F_L(t)$ is the left vertical force time function

$F_R(t)$ is the right vertical force time function

D_{RL} is the period between left and right heel strikes

Q_1,Q_2 are the time scaling factors of the left and right vertical force functions, given for example, by

$$Q_1 = \frac{\text{Left record gait cycle period}}{\text{Combination record gait cycle period}}$$

The force function F_{LRL} of a left-right combination can be reconstructed in a similar fashion as

$$F_{LRL}(t) = F_L\ (Q_1t) + F_R\ (Q_2(t-D_{LR})) + F_L\ (Q_1(t-P))$$

Re-arrangement of the above equations allows a single step force function to be found by substracting the other single step force function from a combination force function. For example, a left record can be found from a left-right combination and a right record by

$$F_L(t) = F_{LR}(\frac{1}{Q_1} t) - F_R \left(\frac{Q_2}{Q_1}(t-D_{LR})\right)$$

These reconstruction procedures are useful when a particular single or combination record is not taken or cannot be easily taken. These procedures are especially useful since they apply to both normal and abnormal gait.

CHAPTER 5

Experimental Procedure

5.1 Introduction

One of the objectives of this project was to produce a general data acquisition system which is easy to use. The system developed is described in Chapter 3. It was used only for gait data acquisition in this project. It proved most effective in this mode. The patient can walk in his own shoes. He does not have to aim for a small plate as is the case for most gait analysis systems. He is unencumbered with wires, lights or markers; and video and film cameras are unnecessary.

The data acquisition program is loaded and started. The patient then traverses the force plate six times. The plate force voltages and walkway switch voltages of these traverses are stored in RAM in real time and then are automatically stored to disc at the end of six traverses by this program. A further advantage of the data acquisition system is that the results are stored in a permanent digital form which is easily accessible for further digital processing and analysis.

5.2 Data Acquisition

A set of seven time varying voltages corresponding to the six plate reactions and the approach switch positions are sent to the A to D board where these voltages are sampled and converted to digital form. A sampling rate of 125 Hz is suitable for normal gait speeds as a typical gait cycle period is 1.2 sec, and only the first 10 harmonics are of significant amplitude. Since this sampling speed is too fast to be handled by a BASIC program, an assembler program is necessary. The Appendix contains the assembler program developed for this purpose. It is capable of sampling all seven channels at 125 Hz and other selected frequencies from 50 to 250 Hz.

A simplified flow chart of the program is shown in Figure 5.1. One of the software faults of the computer operating system is that it is not possible to go from BASIC to a user assembler routine and then back to BASIC under software control. This fault necessitates the tedious task of writing the entire program in assembler, including functions which can be written and executed more easily and efficiently in BASIC, such as file-keeping, user prompts, messages and inputs. Flow chart block A which denotes the filekeeping routine consequently constitutes by far the bulk of the program. Since the details of this section are not important to the main thrust of this project, they are not detailed in the text. The actual data acquisition routine begins at block B which sets all pointers and indices for taking a series of samples. One channel known as the trigger channel is chosen to initiate and terminate the sampling process. Since a significant vertical force V_2 is observed whenever the subject is in contact with the plate, this channel is chosen as the trigger channel. Monitoring of this channel allows sampling to occur only when significant information is being generated by the plate. A threshold of 100 mV corresponding to a 45 N reaction is used. Blocks C and D sample this trigger channel until three consecutive samples in excess of the threshold voltage are found. This prevents accidental triggering by noise spikes. Blocks E and F form the core of the data acquisition program. They cause the A/D board to sample and digitize the voltages of all 7 channels. The program is fast enough that all 7 channels can be sampled in less than 1 msec thereby minimizing the time skew between samples. Digitized values are stored in RAM from hexadecimal addresses &2000 to &7FFF according to the format of Table 5.1. For example, the channel 1 values from traverse 1 are stored consecutively beginning at RAM location &2000. The sampling routine is terminated when more than 32 consecutive samples of the trigger channel fall below the chosen threshold value. Requiring 32 samples below threshold prevents false termination due to noise spikes which may momentarily drive the signal below threshold. It also ensures that the signal is read until it reaches its steady state (zero force) value again. This provides a method of determining the DC offset which is present at the time the sample is taken, and as discussed in the next section, allows this offset to be removed digitally. Block G checks whether all 6 runs have been taken. If not, control is transferred to Block B and another run is taken. Otherwise, block H transfers the data stored in RAM to disc according to the format of Table 5.1. The data from channel 1 of traverse 1 for example, is stored in disc record 1.

Channel 8 is not used for sample storage. Instead, locations 0 and 1 are used to record the number of samples taken and the sampling frequency, as shown in Table 5.2. The other values stored here will be discussed in the following section on editing.

5.3 Data Editing

It is useful to edit the data before it is used for analysis. That the recorded signal is somewhat different than the true signal is clear from Figure 5.2. The editing procedure described here restores the signal as nearly as possible to its original form.

The DC offset can be found from the last few samples taken, as shown in Figures 5.2 and 5.3(a). The first step in editing is to correct the data for possible DC offset as shown in Figure 5.3(b). Fortunately, the vertical foot force starts suddenly and initially increases linearly with time. This fact allows the initial sub-threshold points to be reconstructed by extrapolation from the first several points which were taken. A sample of the final edited data is shown in Figure 5.3(c). The editing program, Edit3, included in the Appendix performs these functions on all six force channels. The number of initial points to be reconstructed is determined by extrapolation of the total vertical force, given by the sum of V_1, V_2 and V_3, to zero. The program also allows rearranging the channel order and independent channel scaling. To ensure that the editing procedure is reversible, the original offsets, and the number of samples before editing are stored in channel 8. The editing program uses the heel strike algorithm discussed in section 5.2 to calculate instants of heel strike. These are also stored in the channel 8 area as shown in Table 5.2. The number of steps on the plate, their side (left or right) and the RL index are input from the user and are also stored.

5.4 Data Plots

Several utility programs were written to assist in the data analysis and presentation. One of these programs, called DP44, which plots the disc data is included in the Appendix. The program is very flexible. By choosing default 0 (zero), the user can select any particular channels and runs from one disc file for plotting. It is often useful to plot all seven channels of all six runs or the total component forces (V_1 + V_2 + V_3), (M_1 + M_2) and A_1 of all 6 runs. The program will produce these plots, for example, if defaults 1 and 2 respectively are specified. These plots are useful in identifying bad data, as for example when the subject steps on the edge of the force plate and part of his weight is carried by the approach. The traverse direction and number of steps can be determined from the plots of V_1 and V_3. These plots also provide easily assimilated visual records of the runs for future reference.

CHAPTER 6

Results and Discussion

6.1 Introduction

The purpose of this chapter is to present the results obtained by applying the algorithms discussed in Chapter 4 to real subject data obtained using the data acquisition system. All data are edited by the procedure described in Section 5.3; the DC offsets of the six force channels are removed and the initial subthreshold data missing from the original record are reconstructed.

The heel strike algorithm described in Section 4.3 is built into the editing program. The heel strikes found by it are used to calculate the offsets and time scales needed to combine force records to form new force records. The same records are used to demonstrate reconstruction of total vertical force and center of force motion.

6.2 Digital Data Acquisition System

The digital data acquisition system proved to be effective and reliable. The bus expansion proved to be a good structure for interfacing peripherals. Not only did it provide physical space and connection slots, but it minimized the additional load on the processing unit. All channels were sampled at 125 Hz. This proved to be a high enough sampling rate as a typical gait cycle period is 1.2 sec and only the first 10 harmonics are significant. Even at this high sampling rate, the time skew between the first and last channels was only 10% of the period between successive samples of the same channel. This allowed samples to be treated as though taken simultaneously.

The data acquisition system and plotter were tested for their accuracy and reliability by employing various waveforms produced by a signal generator. These signals were digitized, printed and plotted for examination. The results of these tests indicate that signals can be read, recorded and plotted accurately.

That the digitizing and editing error is negligible can be seen from Figure 6.1. Figure 6.1(a) shows two vertical force records made using an x-y analogue plotter, while Figure 6.1(b) shows the same force records taken digitally, edited and then plotted digitally. The slight differences are due to a bit of noise in the analogue equipment used to plot Figure 6.1(a) and the limited frequency response of the analogue plotter due to high frequency rolloff.

6.3 The Heel Strike Algorithm

A test was devised to determine the accuracy of the heel strike algorithm. A special pair of shoes having heel and toe contact switches were available. The switches were connected to a special amplifier which produced a discrete voltage for each combination of switch closures, similar to that described in Section 3.3.2 for the approach switches. Several test runs were done in which a subject wearing the special shoes traversed the plate. The three vertical reactions, V_1, V_2 and V_3, the approach switch voltage and the shoe switch voltage were recorded. The resulting record is shown in Figure 6.2. The decoded foot switch positions of run 2 are also shown. The heel strikes as determined from the foot switch record and as determined by the heel strike algorithm are shown in Table 6.1. One program scanned the foot switch record and identified the points where voltage changes represented heel switch closures and hence heel strikes. These sample points are listed in the first part of the table. The heel strike algorithm described in Section 4.3 calculated the heel strike points shown in the last 4 columns of the table.

The extremely good correlation between the heel strike times measured using instrumented shoes, and the times calculated by the heel strike algorithm, shows that the algorithm not only determines heel strike, but that it is extremely accurate as well. Indeed, the data in Table 6.1 show that the algorithm is accurate to within one half of the sample period, or 4 msec. in this case. The accuracy of the algorithm is further verified by the accuracy of the force reconstructions of several different patients (some with abnormal gait) which are discussed in the next section and which rely on the heel strike algorithm for time scaling and offsets.

The fourth heel strike of run 4 is the only figure in the table which is adjusted. The approach switch voltage of this run has only a downward spike at this point rather than a downward step. This is because the approach switches were not properly adjusted when this early run was taken. As a result, they did not remain closed from a force applied to any part of the plate. Their timing fortunately, is not affected by this adjustment. The algorithm value was slightly in error, as it incorrectly used the trailing edge of the spike rather than the leading edge. This value was re-calculated manually from an examination of the data.

The significance of this algorithm is that the period of single foot support for each foot can be determined from force plate data alone. Previously this had been impossible. Accurate heel strike determination also allows determination of weight transfer from foot to foot during double support. Having the forces for single foot support also allows the previously indeterminant problem of joint moments during double support to become determinant (if the limb segment positions are known).

6.4 Force Reconstruction

The second phase of the project is to reconstruct the vertical force records of particular foot fall patterns from the force records of other foot fall patterns. For example, Figure 6.3(a) shows the separate left and right vertical force records of a subject who has an abnormal gait. The subject had previously fractured his right femur. His right foot force pattern is easily identified by the missing first hump which is normally produced by a firm heel strike (compare with force record of a subject with normal gait shown in Figure 6.4). His left foot force record shown in Figure 6.3(a) is more normal. These records are time scaled using the calculated heel strike points as outlined in Section 4.4. In this case they are scaled to match the time scale of the composite record shown with a solid line in Figure 6.3(b). The left record is then time shifted by the right-left heel strike period. This figure shows that the reconstructed right-left record (shown with a dotted line) matches the measured right-left record with less than 3% maximum error. This is very good agreement as variations of up to 10% in successive records of the same subject are not uncommon.

Figure 6.4 shows the reconstruction of a left-right-left record from separate left and right records for a subject with normal gait. Exactly the same time scaling and offset technique were used here. As a final example, Figure 6.5 shows the reconstruction of a right record of another subject having normal gait obtained by subtracting a left record from a right-left record. Both of these final examples also show extremely good correlation (3% maximum error) between the reconstructed record and a measured record of the same type. The examples demonstrate the accuracy of the technique. They also inductively infer its applicability to the addition and subtraction of single and/or multiple steps to reconstruct almost any desired single or multiple step foot force record.

6.5 Center of Force Reconstruction

It seems almost ironic that this project whose analysis initially stemmed from a study of centre of force motion should produce a method to reconstruct the locus of the centre of force. The analysis presented in Section 4.4 is designed to model the center of force position during double stance. The present model treats the foot as an unmoving time-varying point load and hence does not account for the weight transfer from the heel to the toe which occurs during single support. A more refined model would include this second order effect.

Figure 6.3(c) shows as a solid line the center of pressure motion calculated from the vertical force plate reactions, V_1, V_2 and V_3. The dotted line shows the center of force motion reconstructed from the separate left and right foot vertical forces. The figure shows that the reconstructed center of force is, during double support, very similar to the measured one. It is also clear that the difference between the two curves is almost entirely a result of weight transfer from the heel to the toe during single support and which is not accounted for in the present reconstruction.

The center of force reconstruction of a left-right-left sequence is shown in Figure 6.4(c). Here again, the center of force reconstruction is almost indistinguishable from the measured one during most of double support. The difference between the two curves close to and during single support is again caused by the fact that the reconstruction model does not account for the heel to toe weight transfer which occurs during single support.

CHAPTER 7

Conclusions

7.1 Data Acquisition System

The data acquisition, storage and processing system in its final form proved to be very useful and reliable as a laboratory tool. Further development is however necessary before it will be suitable for clinical installations. Before this hardware and software development can be done, the existing CPU must be replaced with any one of the standard and reliable systems which are presently available.

The analogue and digital peripherals and their interfaces proved to be effective, flexible and reliable. These peripherals and the extension bus are suitable for use with any processing system. However, the interface board which plugs into the main computer would have to be replaced with a board which would interface to the new computer bus. The seven channel sampling frequency of 125 Hz is adequate. Even at this high sampling frequency, the inter-channel time skew is sufficiently small that samples across the seven channels can be considered as simultaneous.

7.2 Algorithms and Procedures

It was possible to identify the instants of heel strike from the vertical force record only. By equipping the section of approach immediately adjacent to the force plate with microswitches, at least one additional heel strike could be identified. Comparison with data taken simultaneously from shoes equipped with heel contact switches demonstrated the extremely good accuracy of the algorithm (within 4 msec).

The vertical force records of particular foot patterns (eg. left foot followed by right foot striking the force plate) could be reconstructed from the force patterns of other foot patterns (eg. left foot only striking plate and right foot only striking plate). Additive as well as substractive reconstructions are possible. Comparison with measured force records of the same types demonstrates that the reconstructions are typically accurate within 3%.

The center of pressure can be reconstructed using a method similar to that used for force reconstruction. The reconstruction is often indistinguishable from and usually within 5% of corresponding measured center of pressure records during double support. The error is due almost entirely to motion of the center of pressure during single support. The reconstruction could be made much more accurate if the center of pressure motion during single support is modelled by moving the load application point from the heel to the toe position during the single support period. However, further research and a more complex model would be required.

The reconstruction procedures developed here are useful for generating the force record of any particular foot pattern desired by the clinician. They are especially useful in cases where force plates large enough for multiple successive foot strikes on the plate are not available, as the algorithms allow any force record to be constructed and they make double support weight transfer studies possible. Also, gait time asymmetries can be accurately determined using only a force plate. Neither instrumented shoes nor cine/video methods are necessary.

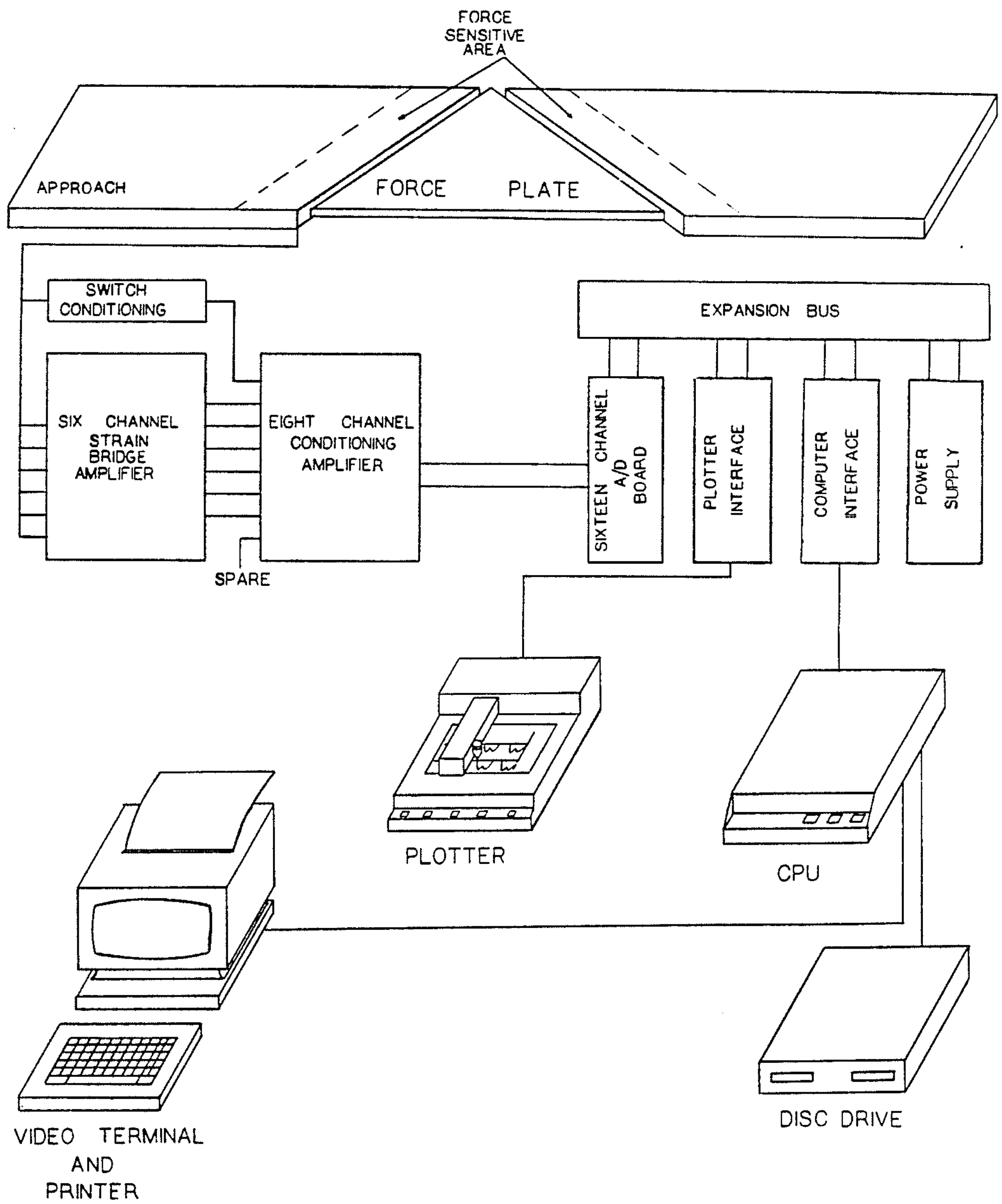
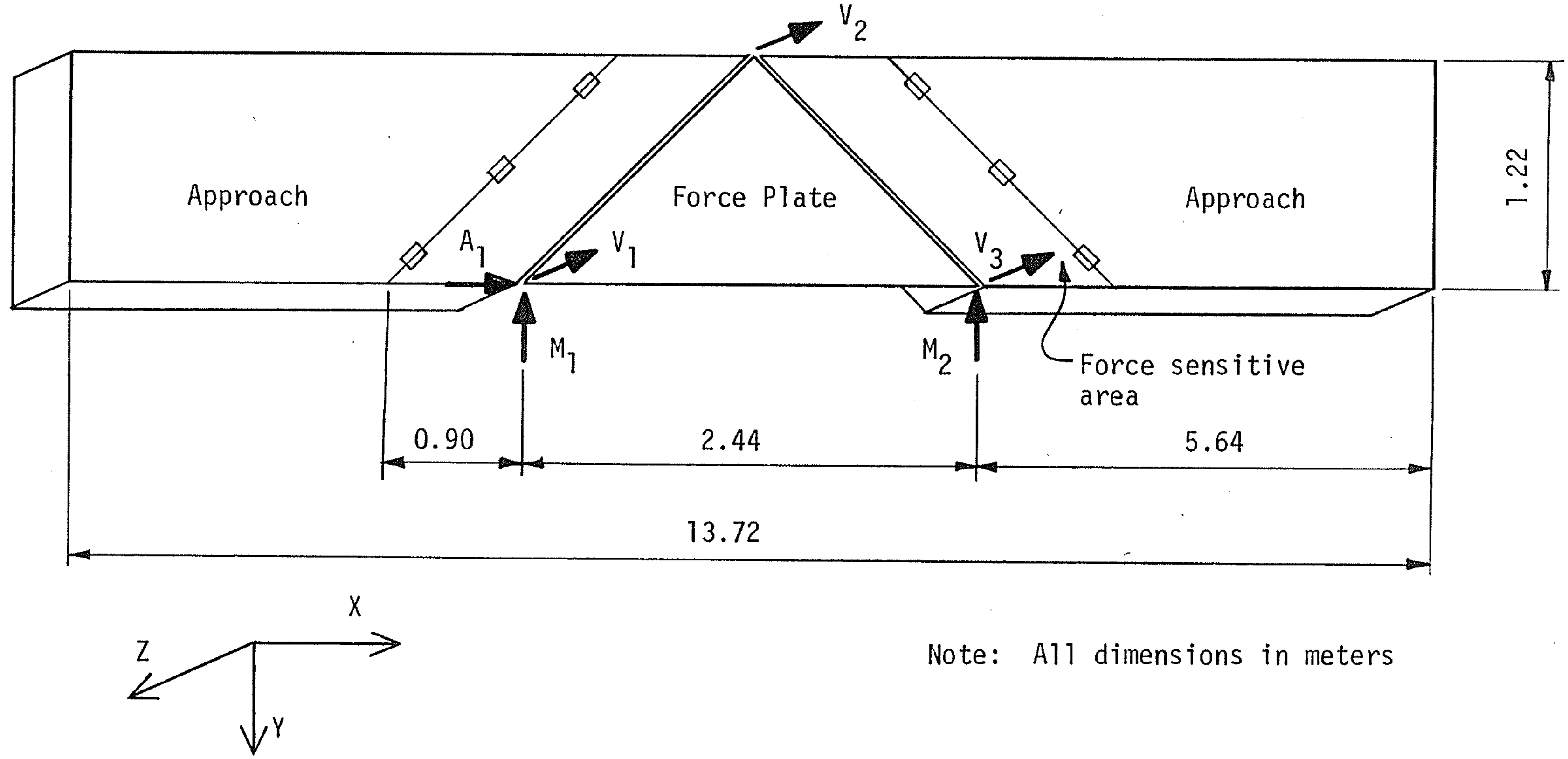


Fig. 3.1 System Configuration



Note: All dimensions in meters

Fig. 3.2 Walkway and Force Plate

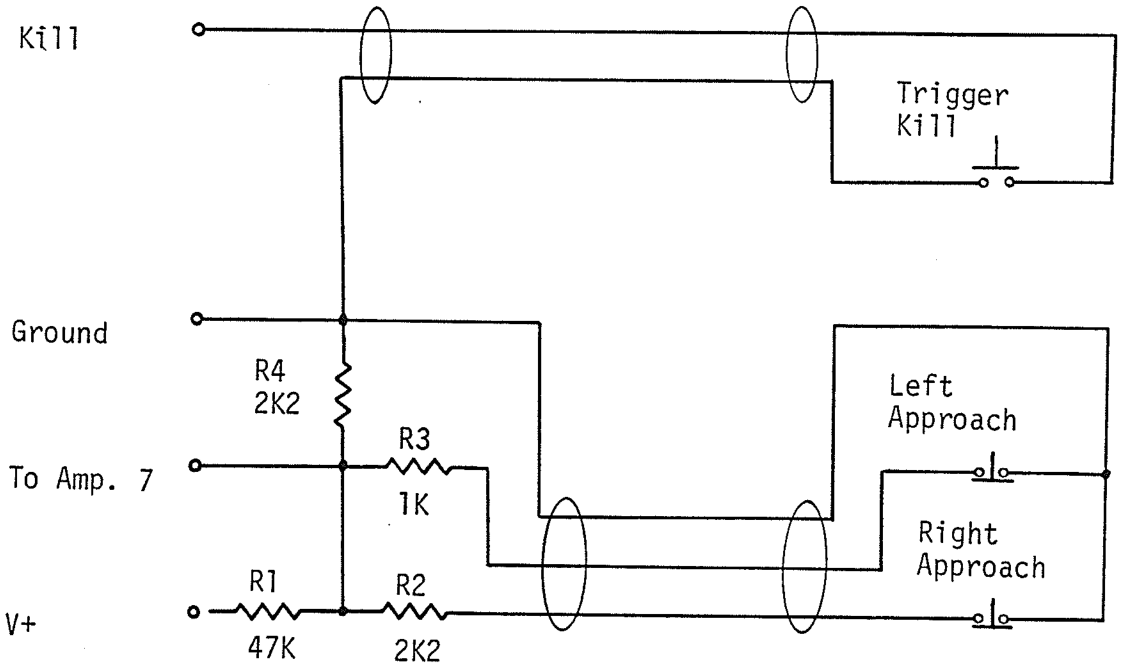


Fig. 3.3 Auxiliary Switches

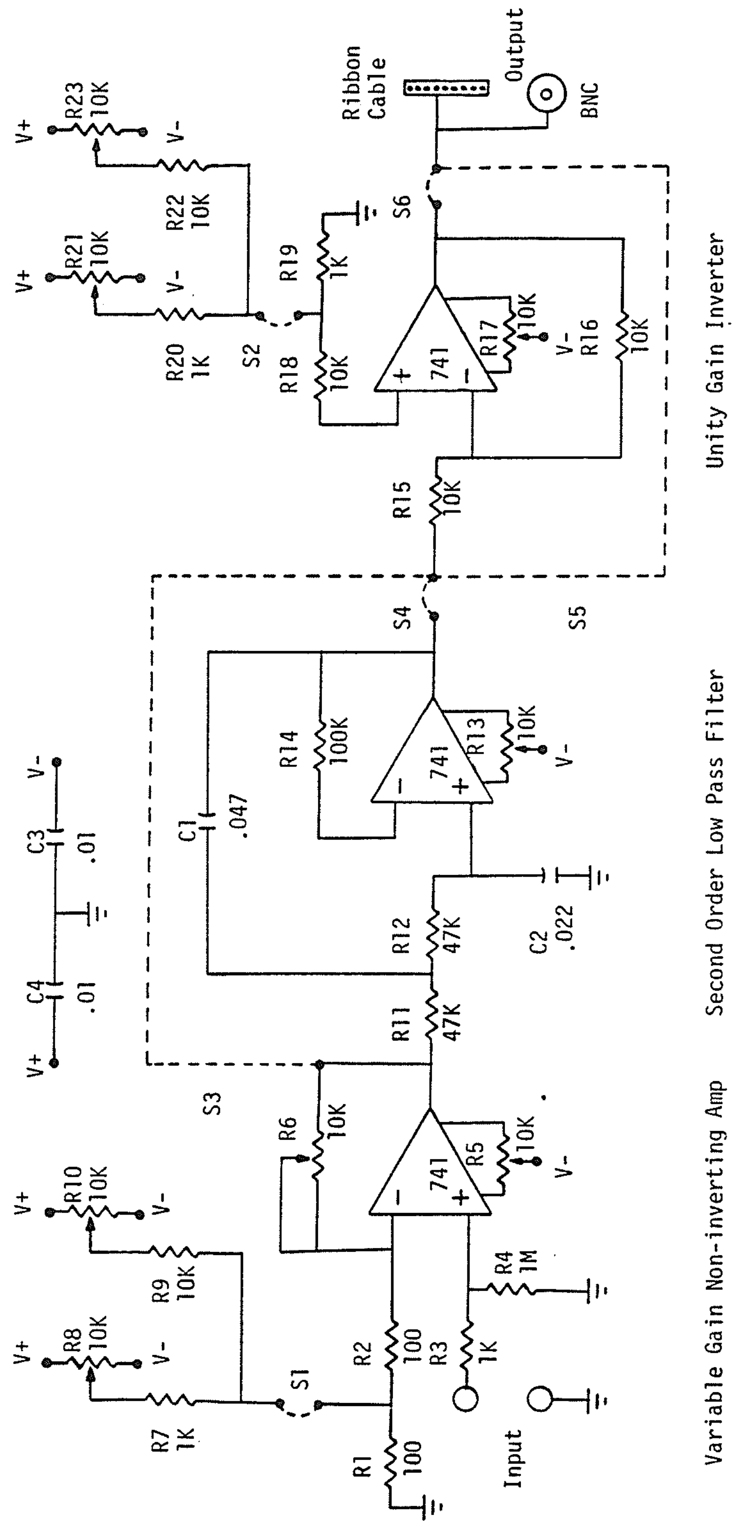


Fig. 3.4 Conditioning Amplifier Schematic

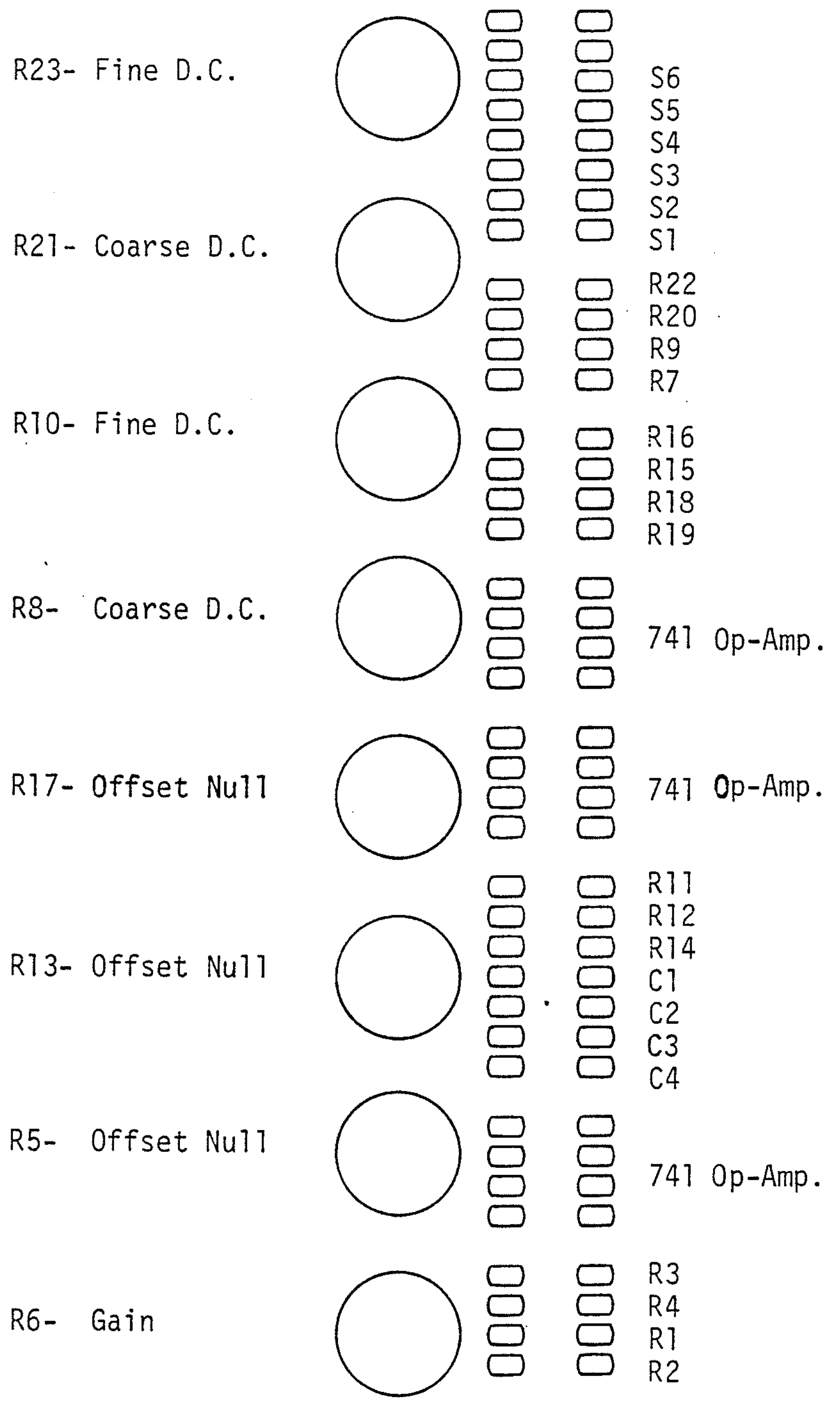


Fig. 3.5 Conditioning Amplifier Layout

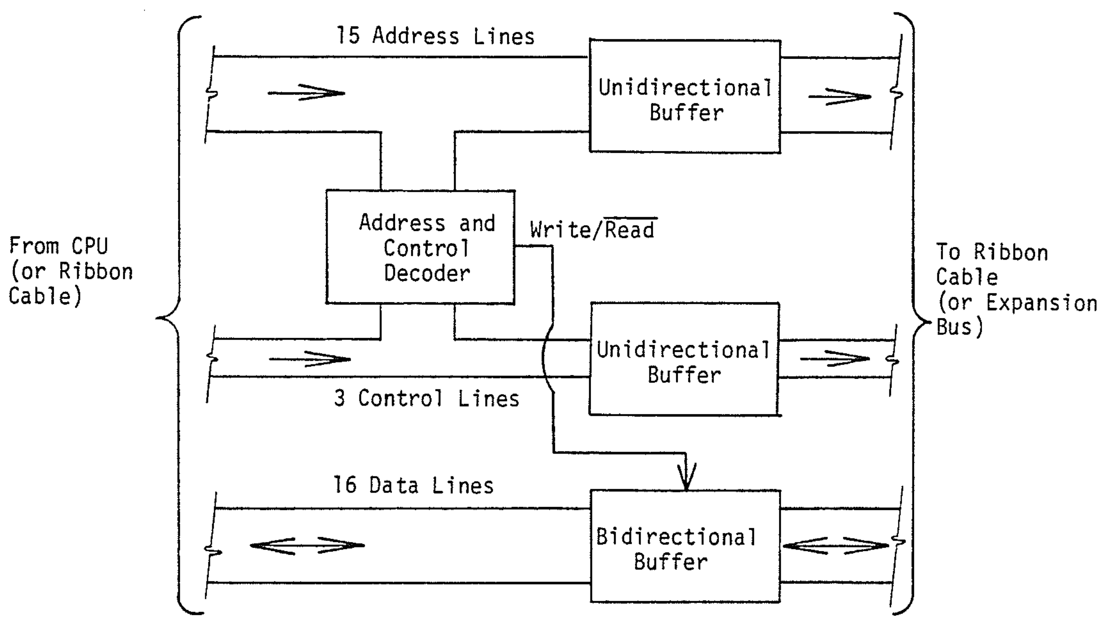


Fig. 3.6 Buffer Boards

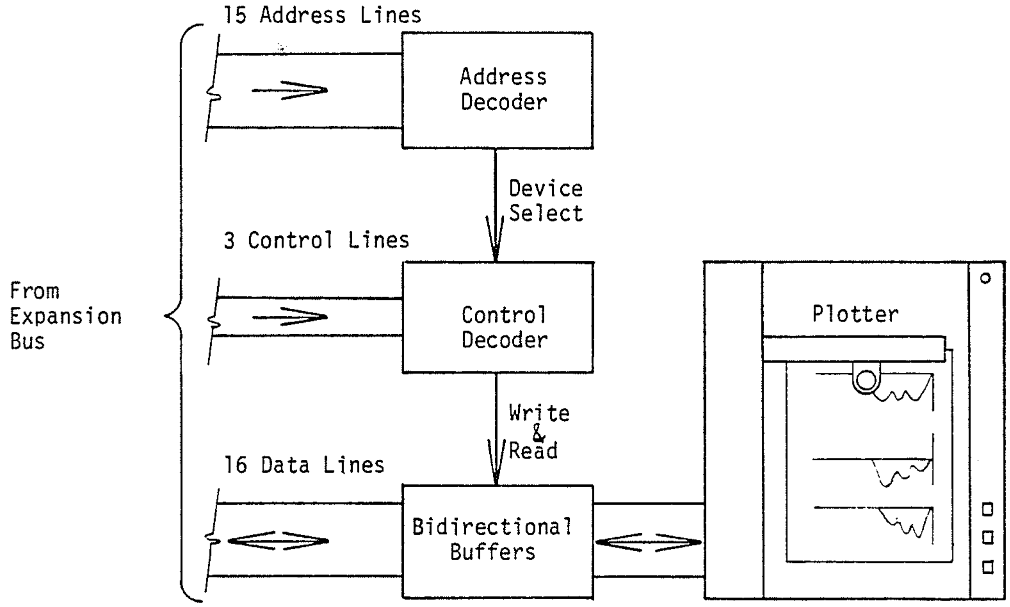
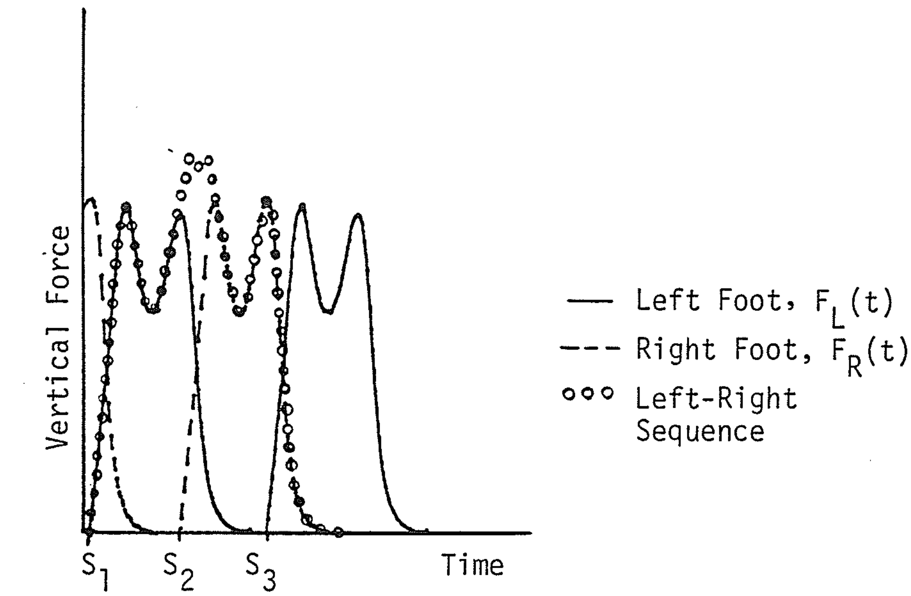
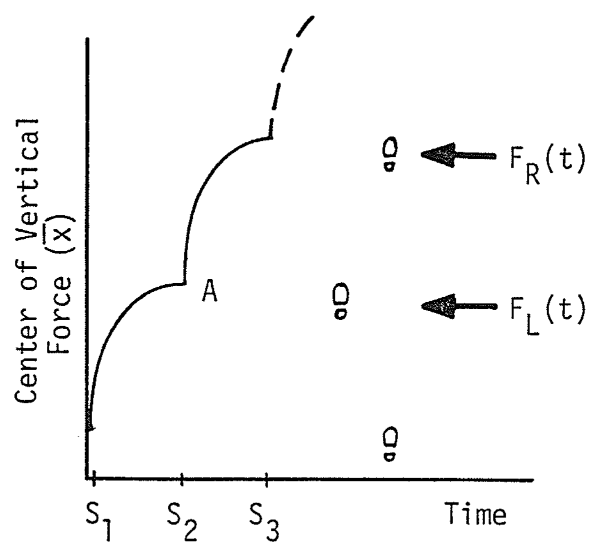


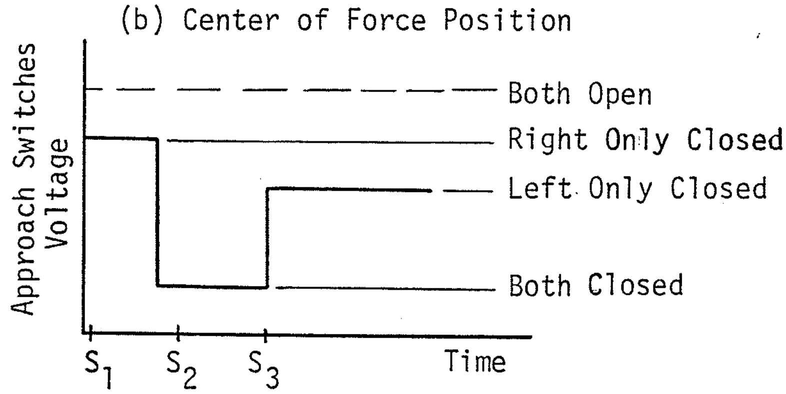
Fig. 3.7 Plotter Interface



(a) Vertical Force



(b) Center of Force Position



(c) Approach Switch Voltage

<u>Fig. 4.1 Typical Gait Record</u>

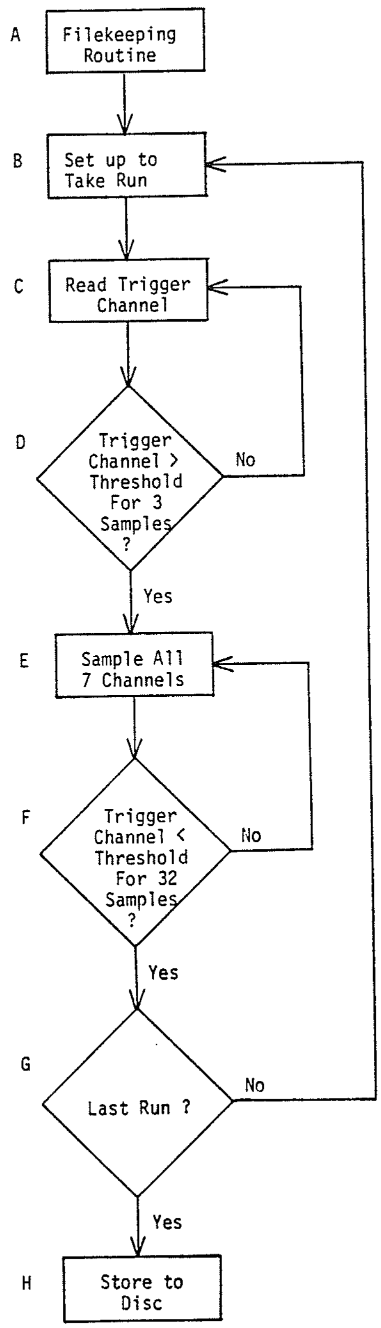


Fig. 5.1 Data Acquisition Program Flow Chart

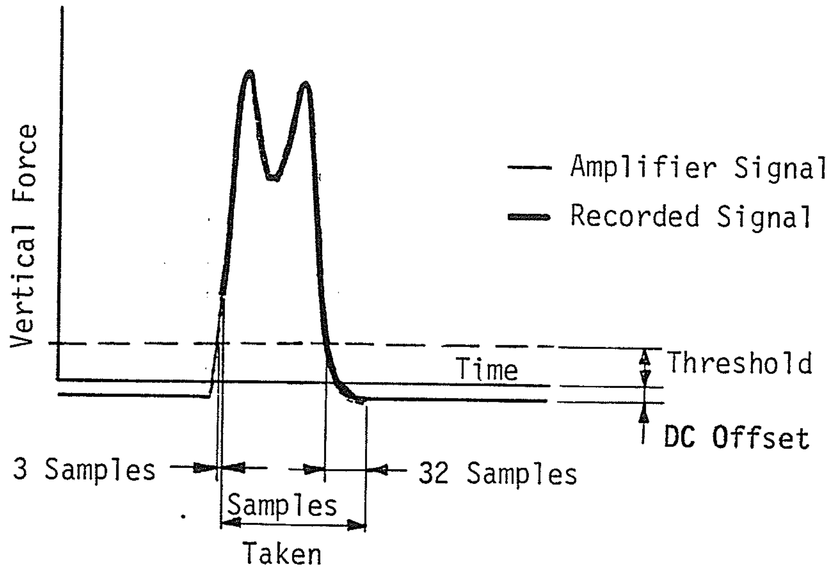


Fig. 5.2 Typical Vertical Force Signal

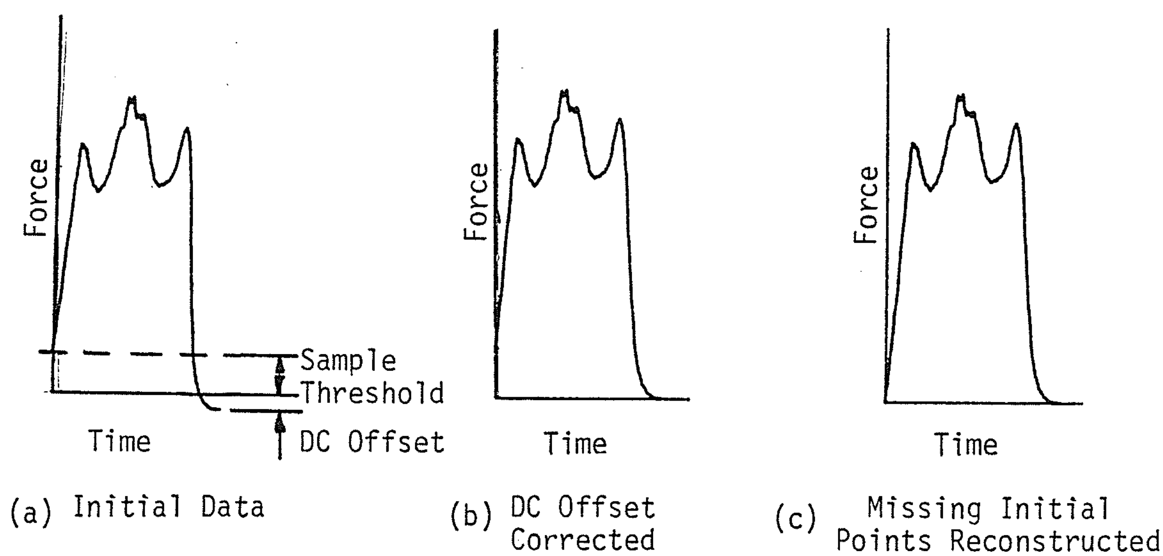
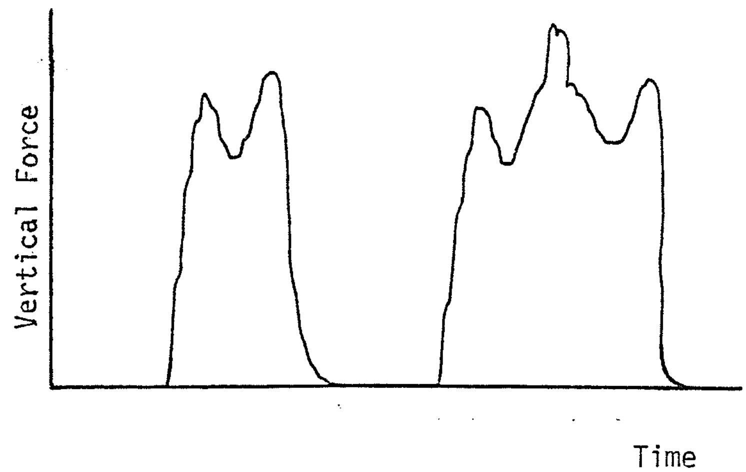
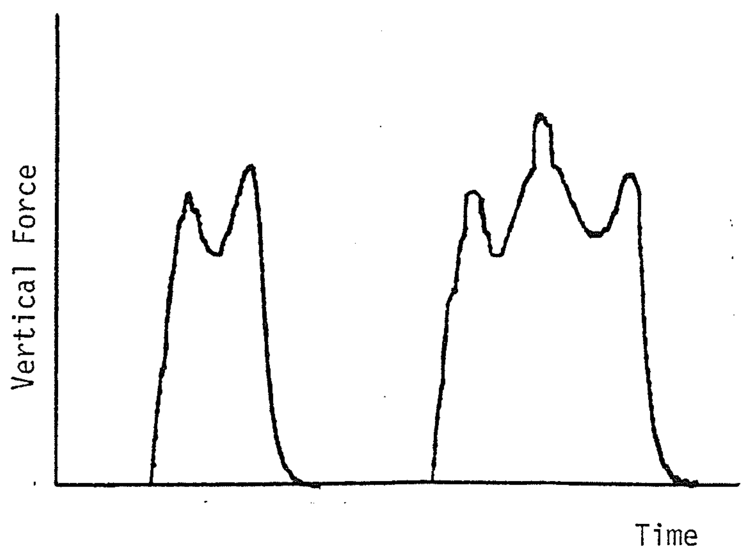


Fig. 5.3 Data Editing Procedure



(a) Analogue Signal



(b) Digital Signal

Fig. 6.1 Comparison of Analogue and Digital Records

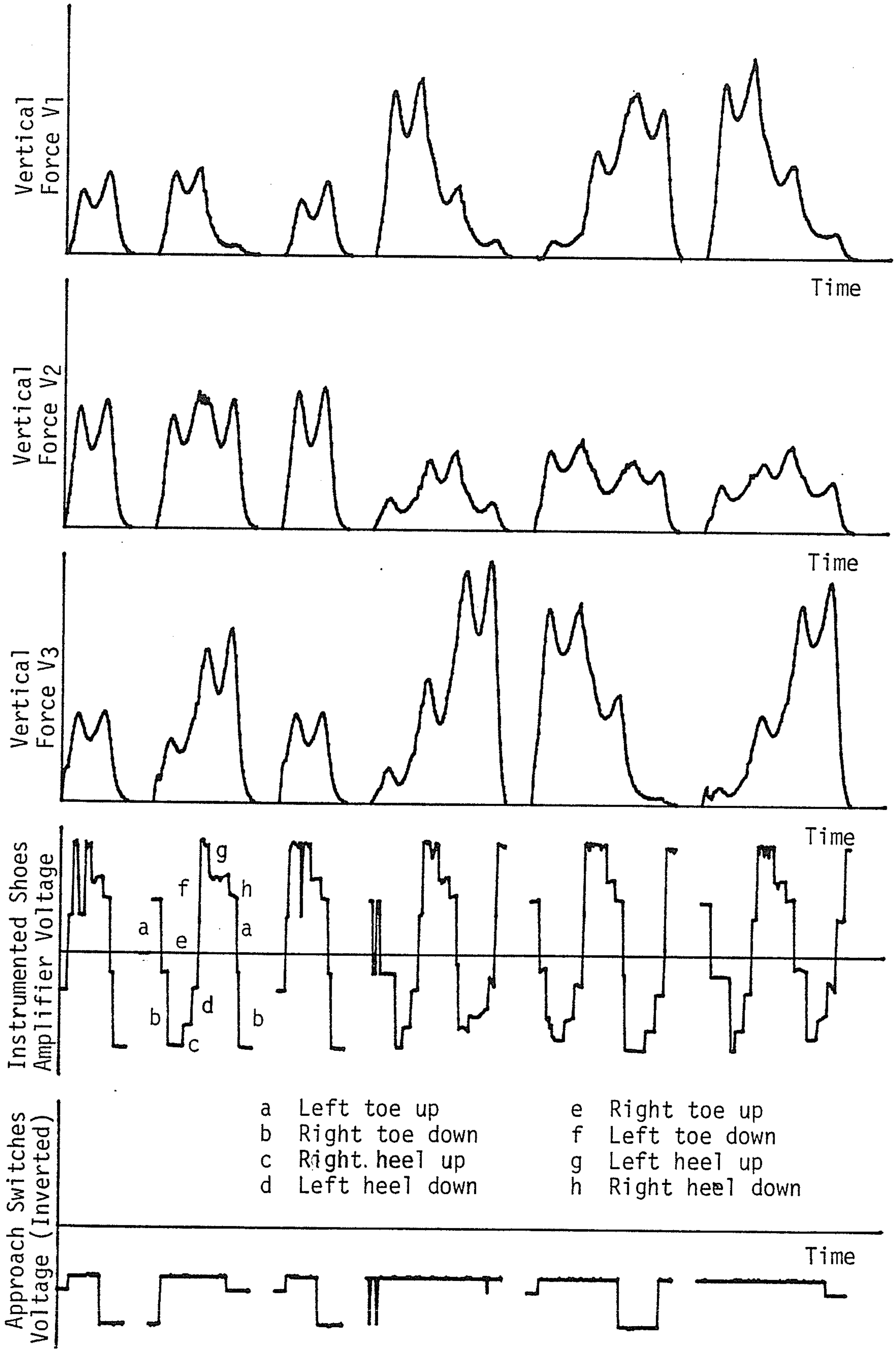


Fig. 6.2 Record to Test Heel Strike Algorithm

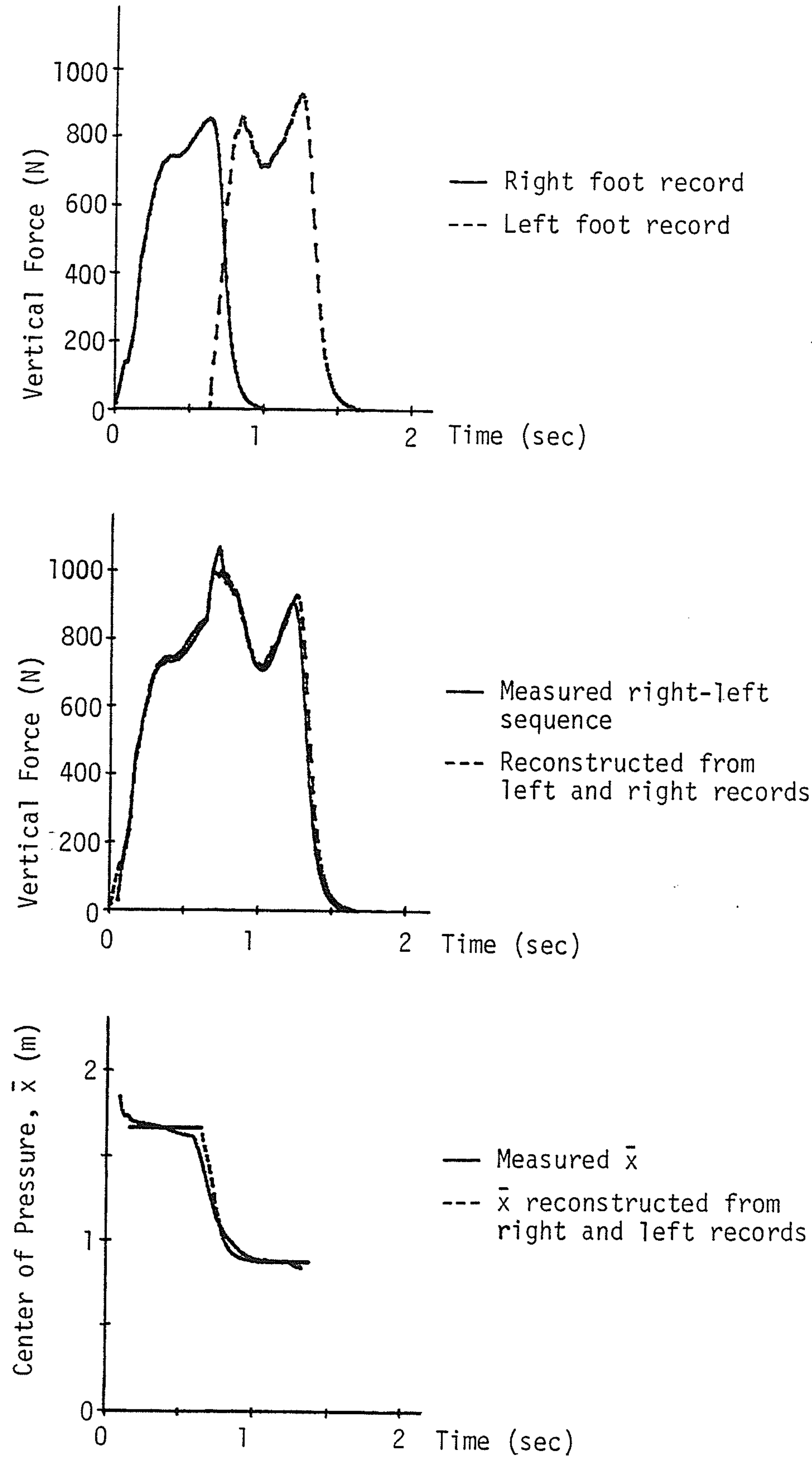


Fig. 6.3 Reconstruction of a Right-Left Record

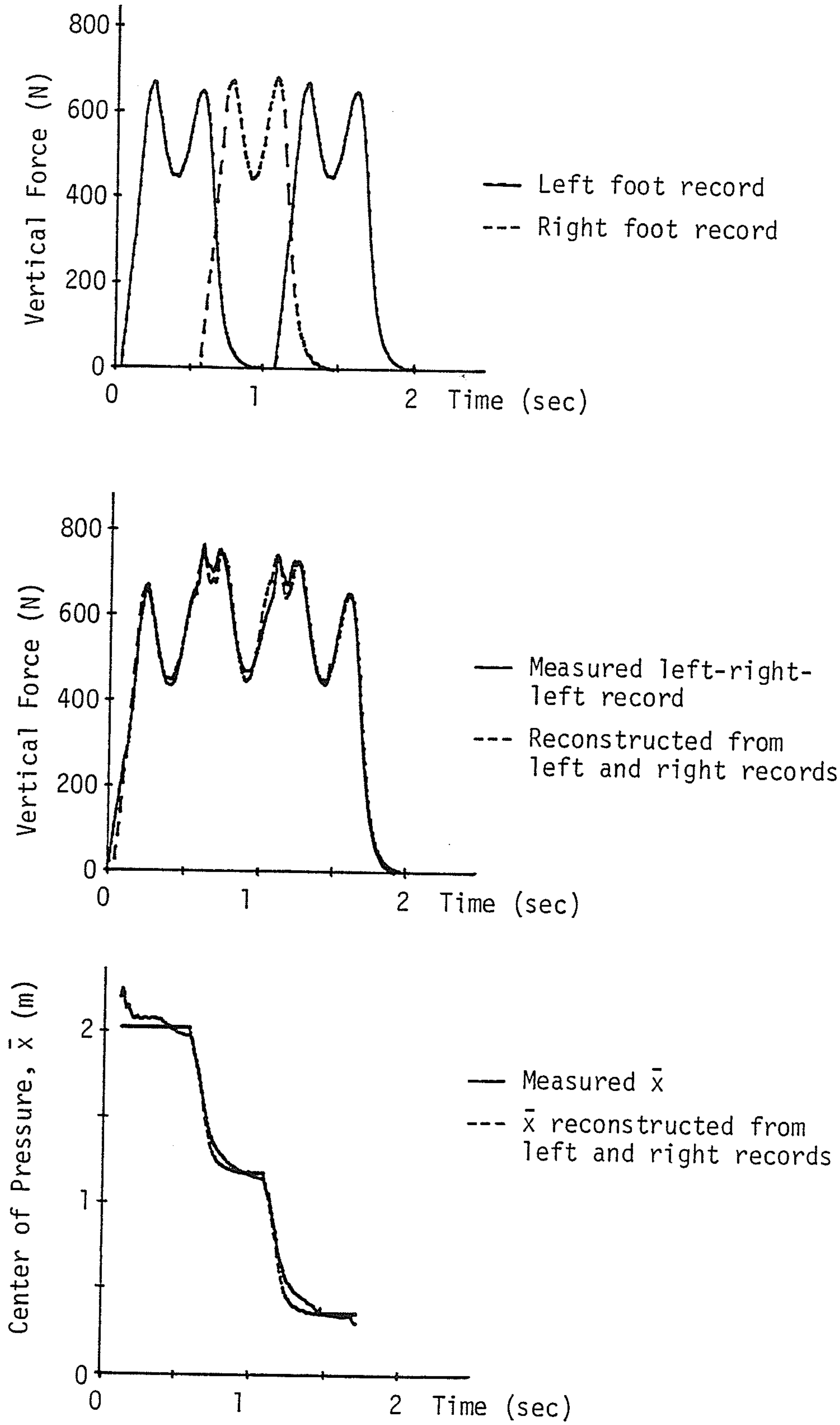


Fig. 6.4 Reconstruction of a Left-Right-Left Record

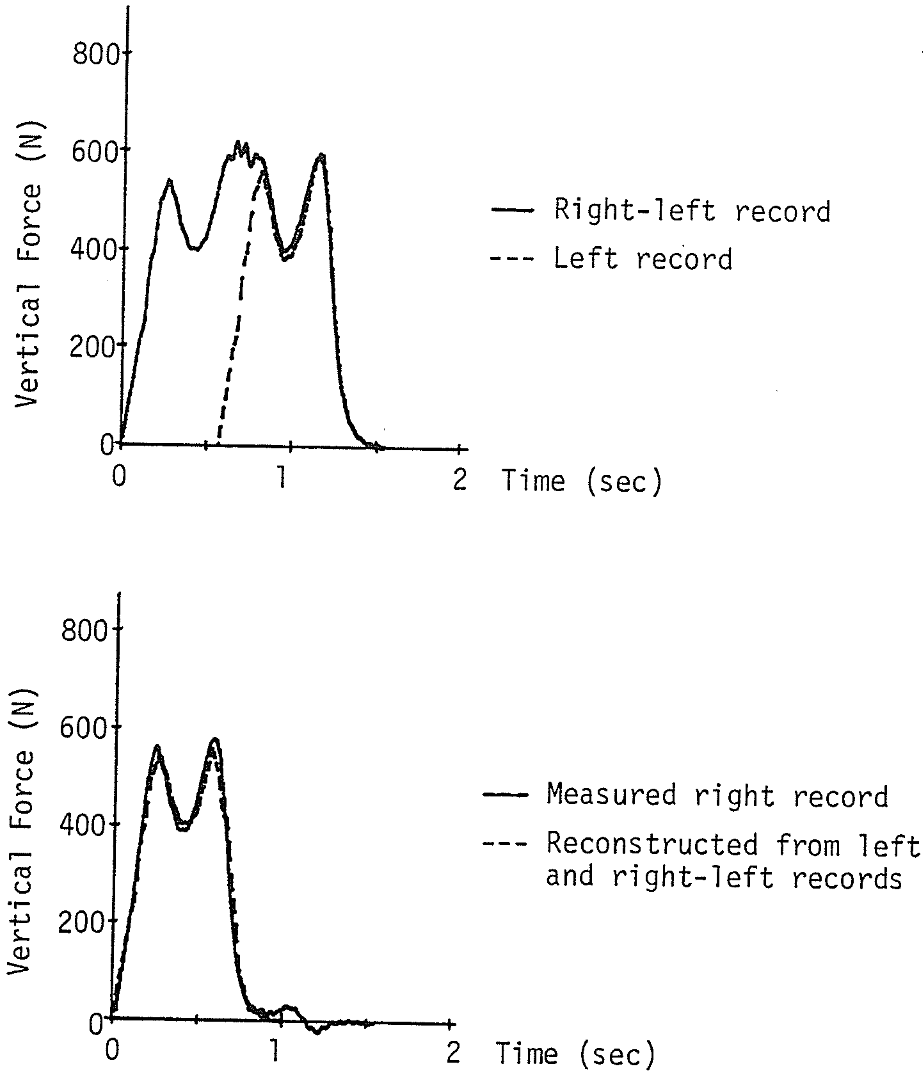


Fig. 6.5 Reconstruction of a Right Record

Table 3.1

Approach Switch Voltages

Left Switch	Right Switch	Parallel Resistance (Ohms)	V_7 / V^+	V_7 For V^+ = 15V
Open	Open	2200	0.0447	0.671
Closed	Open	688	0.0144	0.216
Open	Closed	1100	0.0229	0.344
Closed	Closed	524	0.0110	0.165

Table 5.1

Memory Map and Disc Record Allocation

CHANNEL	Traverse					
	1	2	3	4	5	6
1	&2000-21FF 1	&3000-31FF 9	&4000-41FF 17	&5000-51FF 25	&6000-61FF 33	&7000-71FF 41
2	&2200-23FF 2	&3200-33FF 10	&4200-43FF 18	&5200-53FF 26	&6200-63FF 34	&7200-73FF 42
3	&2400-25FF 3	&3400-35FF 11	&4400-45FF 19	&5400-55FF 27	&6400-65FF 35	&7400-75FF 43
4	&2600-27FF 4	&3600-37FF 12	&4600-49FF 20	&5600-57FF 28	&6600-67FF 36	&7600-77FF 44
5	&2800-29FF 5	&3800-39FF 13	&4800-49FF 21	&5800-59FF 29	&6800-69FF 37	&7800-79FF 45
6	&2A00-2BFF 6	&3400-3BFF 14	&4A00-4BFF 22	&5A00-5BFF 30	&6A00-6BFF 38	&7A00-7BFF 46
7	&2C00-2DFF 7	&3C00-3DFF 15	&4C00-4DFF 23	&5C00-5DFF 31	&6C00-6DFF 39	&7C00-7DFF 47
8	&2E00-2FFF 8	&3E00-3FFF 16	&4E00-4FFF 24	&5D00-5FFF 32	&6E00-6FFF 40	&7E00-7FFF 48

Table 5.2

"Channel 8" Format

Entry #	Description
0	No. of data points after editing
1	Sampling frequency
2	No. of data points before editing
3 - 8	Channel DC offsets before editing
11	No. of steps
12 - 15	Step details (Right = 4, Left = 2)
16	RL Symmetry Index
21 - 25	Heel strike sample numbers

Table 6.1

Comparison of Heel Strike Times

Run	No. of Steps	Strike Times From Foot Switches				Strike Times From Algorithm			
		1	2	3	4	1	2	3	4
1	1	1	71			1	72		
2	2	1	67	128		1	67	129	
3	1	1	72			1	72		
4	3	1	71	132	197	1	72	132	198*
5	3	1	74	140	208	1	75	141	**
6	3	1	73	141	212	1	73	141	213

Times shown measured in 1/125 sec. (sample number) from first heel strike.

* See Text, Page 29.

** Foot struck edge of force plate.

REFERENCES

1. Borelli, De Motu Animalium, 1682.

2. Carlet, M.G. 'Essai experimental sur la locomotion humain', Etude de la marche. Ann. d. Scien. mat., Zool. 16: pp. 1-92, 1872.

3. Marey, E.J. 'De la locomotion terrestre chez les bipedes et les quadrupedes,' J. de L'Anat. et. de la Physiol. 9: 42-80, 1873.

4. Beely, F., 'Zur mechanik des stehens: uber die bedentung des fusage wolbens beim stehen,' Archiv fur Klinische Chirurgie 27457, 1882.

5. Abrahamson, E., 'Zur kenntnis der mechanik des mittelfusses,' Skord. Arch. Physiol. Bol. 51, pp. 175-235, 1927.

6. Basler, 'Bestimmurg des auf die eingelmen schlenbegirke wirkerden teilgewichlis des mimehlichen korpers,' Abdertaldens Handbuch, Abt. V, Teil 51 Hift 3, pp. 559-574, 1927.

7. Amar, J. 'Trottior dynamographique', Compt. rend. Acad. d. Sc. 163: 130, 1916.

8. Elftman, H., 'A cinematic study of the distribution of pressure in the human foot', Anat. Rec. 59: pp. 481-490, 1934.

9. Cunningham, D.M. and G.W. Brown, 'Two devices for measuring the forces acting on the human body during walking', Proc. Soc. Exp. Stress Anal. IX, 2,75, 1952.

10. Murray, M.P., Gait as a total pattern of movement, American Journal of Physical Medicine, Vol. 46, No. 1, pp. 290-333, 1967.

11. Murray, M.P. et al., Normal postural stability and steadiness: quantitative assessment, The Journal of Bone and Joint Surgery. Vol. 57-A, No. 4, pp. 510-516, 1975.

12. Jacobs, N.A., J. Skorecki and J. Charnley, Analysis of the vertical component of force in normal and pathological gait, J. Biomechanics, Vol. 5, pp. 11-34, 1972.

13. Balakrishnan, S., The interpretation of medio-lateral reaction forces in human locomotion using body coordinate systems, Ph.D. Thesis, University of Manitoba, 1982.

14. Grundy, Michael, et al., 'An investigation of the centres of pressure under the foot while walking', The Journal of Bone and Joint Surgery, Vol 57-B, No. 1, pp. 98-103, Feb. 1975.

15. Yamashita, T. and R. Katoh, Moving pattern of point of application of vertical resultant force during level walking, J. Biomechanics, Vol. 9, pp. 93-99, 1976.

16. Schwartz, R.P. et al., A quantitative analysis of recorded variables in the walking pattern of "normal" adults, The Journal of Bone and Joint Surgery, Vol. 46-A, No. 2, pp. 324-334, 1964.

17. Cunningham, D.M., Components of floor reactions during walking. Prosthetic Devices Res. Rep. Inst. of Eng. Res. Univ. of California, Berkeley, Series II, Issue 14, 1958.

18. Murray, M.P. et al., Center of gravity, center of pressure, and supportive forces during human activities, Journal of Applied Physiology, Vol. 23, No. 6, pp. 831-838, 1967.

19. Barnett, C.H., 'A plastic pedograph', Lancet, 2, 273, 1954.

20. Elftman, H. 'The measurement of the external force in walking', Science 84, pp. 152-153, 1938.

21. Balakrishnan, S. and A.B. Thornton-Trump, 'Measurement of reaction forces between foot and ground using a modified force plate', CANCAM, 1979.

22. Scranton, P.E., and J.H. McMaster, Momentary distribution of forces under the foot, J. Biomechanics, Vol. 9, pp. 45-48, 1976.

23. Spolek, G.A., and F.G. Lippert, An instrumented shoe - A portable force measuring device, J. Biomechanics, Vol. 9, pp. 779-783, 1976.

24. Hennig, E.M., High resolution in-shoe pressure distribution measurements by piezoelectric transducers, Proceedings of the Special Conference of the Canadian Society for Biomechanics, pp. 120-121, 1980.

25. Murray, M.P. et al., Comparison of free and fast speed walking patterns of normal men, American Journal of Physical Medicine, Vol. 45, No. 1, pp. 8-24, 1966.

26. Cavanagh, P.R. and M.L. Lafortune, 'Ground reaction forces in distance running', J. Biomechanics, Vol. 13, pp. 397-406, 1980.

27. Cavanagh, P.R., A technique for averaging center of pressure paths from a force platform, J. Biomechanics, Vol. 11, pp. 487-491, 1978.

28. Coughlin, R.F., and F.F. Driscoll, Operational amplifiers and linear integrated circuits, Prentice Hall Inc., Englewood Cliffs, N.J., 1977.

29. Alexander, R., and A.S. Jayes, Fourier analysis of forces exerted in walking and running, J. Biomechanics, Vol. 13, pp. 383-390, 1980.

30. Bouman, J.H., and P.W. Brand; 'Measurement of pressure between foot and shoe', Lancet 1: pp. 629-632, 1963.

31. Brodland, G.W., Motion of the center of vertical force during walking, University of Manitoba, Unpublished, 1978.

32. Godfrey, C.M. et al., 'A method for determination of pedal pressure changes during weight bearing', Prelimiary observations in normal and arthritic feet', Arthritis and Rheumatism, 10, pp. 135-140, 1967.

33. Kausand, T., M. Milner, A.O. Quanbury and D.A. Winter, 'Computers and the kinesiology of gait', Comput. Biol. Med. Vol. 6, pp. 111-120, 1976.

34. Kluger, J. 'The human machine', Science Digest, pp. 64-71, June 1982.

35. Moreton, D.N. and M.J. Maxted, A technique to determine the load distribution in human feet, Strain, pp. 140-143, Oct. 1979.

36. Quanbury, A.O., Gait assessment; Description or explanation, proceedings of the special conference of the canadian society for biomechanics, pp. 82-83, 1980.

APPENDIX

Computer Program Listings

```
                            TITL  'GAIT DATA ACQUISITION - PART 1 (VER 08/09/81)'
 085C              PART1    IDT
                            DREG
                   *
                   * EXTERNAL REFERENCES
                   *
                            REF    LIST,UPDATE,SAMPLE
                   *
                   * EXTERNAL DEFINITIONS
                   *
                            DEF    INPUT,SETUP1,SETUP2,MOD,WRAPUP
                            DEF    MSGOUT,MSG11,DTH,LABEL1,OPEN,DTH01,DTH02
                            DEF    BUFFER,RNUM,MSG30,MSG31,LABEL4,MSGLP2
                            DEF    MSG10,DTH03,DTH07,MSG40,MSG50,DATAWP,CR
                   *
                   * RAM DATA BASE
                   * NOTE: ORDER IS IMPORTANT: CHANGE WITH CARE.
                   *
 0100                       AORG   >100            ; STARTING ADDRESS OF RAM
 0100              MAINWP   BSS    32              ; MAIN PROGRAM WORK SPACE
 0120              DATAWP   BSS    32              ; WORKSPACE FOR SAMPLE
 0140              BUFFER   BSS    130             ; TEXT BUFFER
 01C2              DTHCMD   BSS    32              ; BUFFER FOR DTH COMMANDS
 01E2              FNAME    BSS    9               ; FILE NAME FOR DTH COMMANDS
 01EB              DATE     BSS    7               ; DATE
 01F2              DATE1    BSS    9               ; DD/MM/YY
 01FC                       EVEN
 01FC              RNUM     BSS    2               ; RUN NUMBER
 01FE              RNUM1    BSS    4               ; CURRENT RUN NO.
 0140              BOUND1   EQU    BUFFER          ; LOWER BOUND FOR DTH CALLS
 01C0              BOUND2   EQU    DTHCMD-2        ; UPPER BOUND FOR DTH CALLS
                   *
                   * MONITOR INTERFACE
                   *
 0000                       RORG   *
 0000 0460 0786    START    B      @MAIN
                   *
                   * ROM DATA BASE
                   *
 0004 0D           CR       BYTE   >0D             ; CARRIAGE RETURN
 0005 0A           LF       BYTE   >0A             ; LINE FEED
 0006 40           ENSIGN   BYTE   '@'             ; SIGN FOR END OF TEXT
 0007 2F           SLASH    BYTE   '/'             ; SLASH
 0008 59           YES      BYTE   'Y'             ; YES
 0009 20           BLANK    BYTE   ' '             ; BLANK
 000A 41           AF       BYTE   'A'             ; ALLOCATE FILE
 000B 43           CF       BYTE   'C'             ; COPY FILES
 000C 46           DU       BYTE   'F'             ; DEFINE UNIT
 000D 4B           ID       BYTE   'K'             ; INITIALIZE DISK
 000E 52           IR       BYTE   'R'             ; INPUT RECORD
 000F 57           OR       BYTE   'W'             ; OUTPUT RECORD
 0010 50           PF       BYTE   'P'             ; POSITION UNIT
 0011 58           RD       BYTE   'X'             ; RESET DISK
 0012 53           SF       BYTE   'S'             ; SAVE FILE (DISK)
 0013 30           ZERO     BYTE   '0'
 0014 31           ONE      BYTE   '1'
 0015 32           TWO      BYTE   '2'
 0016 33           THREE    BYTE   '3'
```

```
0017 34             FOUR    BYTE  '4'
0018 35             FIVE    BYTE  '5'
0019 36             SIX     BYTE  '6'
001A 37             SEVEN   BYTE  '7'
001B 38             EIGHT   BYTE  '8'
001C 39             NINE    BYTE  '9'
001E 000A           TEN     DATA  10
0020 0064           HUNDRD  DATA  100
0022 2020           BLANKS  DATA  '  '
0024 0608 0044      TABCMD  DATA  DATACQ,'D'
0028 0000 0055              DATA  UPDATE,'U'
002C 0000 004C              DATA  LIST,'L'
0030 0000                   DATA  0
                    *
                    * LABEL FILE
                    *
0032 02B3           LABEL1  DATA  MSG20
0034 02C3                   DATA  MSG21
0036 02C9                   DATA  MSG22
0038 02D2                   DATA  MSG23
003A 02DB                   DATA  MSG24
003C 02EA                   DATA  MSG25
003E 02FB                   DATA  MSG26
0040 0309                   DATA  MSG27
0042 031A                   DATA  MSG28
0044 032A                   DATA  MSG29
0046 0000                   DATA  0
0048 3131           LABEL2  DATA  '11'          ; SECTOR 11
004A 3132                   DATA  '12'          ; SECTOR 12
004C 3133                   DATA  '13'          ; SECTOR 13
004E 3134                   DATA  '14'          ; SECTOR 14
0050 3135                   DATA  '15'          ; SECTOR 15
0052 3136                   DATA  '16'          ; SECTOR 16
0054 3230           LABEL3  DATA  '20'          ; SECTOR 20
0056 3234                   DATA  '24'          ; SECTOR 24
0058 3238                   DATA  '28'          ; SECTOR 28
005A 3332                   DATA  '32'          ; SECTOR 32
005C 3336                   DATA  '36'          ; SECTOR 36
005E 3430                   DATA  '40'          ; SECTOR 40
0060 036B           LABEL4  DATA  MSG33
0062 0375                   DATA  MSG34
0064 037F                   DATA  MSG35
0066 0000                   DATA  0
                    *
                    * DTH COMMAND FILE
                    *
0068 5020 3120      DTH01   TEXT  'P 1 0 0'     ; SET FILE1 TO THE BEGINNING
006C 3020 30
006F 0D                     BYTE  >0D
0070 5220 3120      DTH02   TEXT  'R 1 128'     ; INPUT A RECORD (128 BYTES)
0074 3132 38
0077 0D                     BYTE  >0D
0078 5720 3120      DTH03   TEXT  'W 1 128'     ; OUTPUT A RECORD (128 BYTES)
007C 3132 38
007F 0D                     BYTE  >0D
0080 58             DTH04   TEXT  'X'           ; RESET DISK DRIVES
0081 0D                     BYTE  >0D
0082 46             DTH05   TEXT  'F'           ; CLOSE ALL OPENED FILES
```

PAGE-3 GAIT DATA ACQUISITION - PART 1 (VER 08/09/81)

```
0083 0D                     BYTE   >0D
0084 5020 3120   DTH06      TEXT   'P 1 10'          ; POSN FILE TO SECTOR 10
0088 3130
008A 0D                     BYTE   >0D
008B 5020 3120   DTH07      TEXT   'P 1 20 0 '
008F 3230 2030
0093 20
0094 0D                     BYTE   >0D
0095 5020 3120   DTH10      TEXT   'P 1 '            ; POSITION FILE 1
0099 0D                     BYTE   >0D
                 *
                 * SYSTEM MESSAGES
                 *
009A 0A          MSG01      BYTE   >0A
009B 2A2A 2A2A              TEXT   '**** GAIT DATA ACQUISITION SYSTEM ****'
009F 2047 4149
00A3 5420 4441
00A7 5441 2041
00AB 4351 5549
00AF 5349 5449
00B3 4F4E 2053
00B7 5953 5445
00BB 4D20 2A2A
00BF 2A2A
00C2 0A0D                   DATA   >0A0D
00C4 454E 5445   MSG02      TEXT   'ENTER DATE (DD/MM/YY): '
00C8 5220 4441
00CC 5445 2028
00D0 4444 2F4D
00D4 4D2F 5959
00D8 293A 20
00DB 0D                     BYTE   >0D
00DC 0A          MSG03      BYTE   >0A
00DD 4655 4E43              TEXT   'FUNCTIONS AVAILABLE: '
00E1 5449 4F4E
00E5 5320 4156
00E9 4149 4C41
00ED 424C 453A
00F1 20
00F2 0D                     BYTE   >0D
00F3 2020 312E   MSG04      TEXT   '  1.  DATA ACQUISITION (D), '
00F7 2020 4441
00FB 5441 2041
00FF 4351 5549
0103 5349 5449
0107 4F4E 2028
010B 4429 2C20
010F 0D                     BYTE   >0D
0110 2020 322E   MSG05      TEXT   '  2.  UPDATE PATIENT FILE (U), AND '
0114 2020 5550
0118 4441 5445
011C 2050 4154
0120 4945 4E54
0124 2046 494C
0128 4520 2855
012C 292C 2041
0130 4E44 20
0133 0D                     BYTE   >0D
```

PAGE-4 GAIT DATA ACQUISITION - PART 1 (VER 08/09/81)

```
0134 2020 332E  MSG06   TEXT  '  3.  LIST PATIENT FILE (L). '
0138 2020 4C49
013C 5354 2050
0140 4154 4945
0144 4E54 2046
0148 494C 452C
014C 284C 292E
0150 20
0151 0D                 BYTE  >0D
0152 0A         MSG07   BYTE  >0A
0153 4348 4F4F          TEXT  'CHOOSE YOUR FUNCTION BY THE FIRST LETTER: '
0157 5345 2059
015B 4F55 5220
015F 4655 4E43
0163 5449 4F4E
0167 2042 5920
016B 5448 4520
016F 4649 5253
0173 5420 4C45
0177 5454 4552
017B 3A20
017D 0D                 BYTE  >0D
017E 2A2A 2A20  MSG08   TEXT  '*** INVALID INPUT - TRY AGAIN: '
0182 494E 5641
0186 4C49 4420
018A 494E 5055
018E 5420 2D20
0192 5452 5920
0196 4147 4149
019A 4E3A 20
019D 0D                 BYTE  >0D
019E 0A         MSG09   BYTE  >0A
019F 2A2A 2A20          TEXT  '*** DATA ACQUISITION ***'
01A3 4441 5441
01A7 2041 4351
01AB 5549 5349
01AF 5449 4F4E
01B3 202A 2A2A
01B8 0A0D               DATA  >0A0D
01BA 0A         MSG10   BYTE  >0A
01BB 2A2A 2A20          TEXT  '*** UPDATE MASTER FILE ***'
01BF 5550 4441
01C3 5445 204D
01C7 4153 5445
01CB 5220 4649
01CF 4C45 202A
01D3 2A2A
01D6 0A0D               DATA  >0A0D
01D8 0A         MSG11   BYTE  >0A
01D9 2A2A 2A20          TEXT  '*** LIST MASTER FILE ***'
01DD 4C49 5354
01E1 204D 4153
01E5 5445 5220
01E9 4649 4C45
01ED 202A 2A2A
01F2 0A0D               DATA  >0A0D
01F4 4953 2054  MSG12   TEXT  'IS THE PATIENT NEW TO THE SYSTEM? (Y/N) '
01F8 4845 2050
```

PAGE-5 GAIT DATA ACQUISITION - PART 1 (VER 08/09/81)

```
01FC 4154 4945
0200 4E54 204E
0204 4557 2054
0208 4F20 5448
020C 4520 5359
0210 5354 454D
0214 3F20 2859
0218 2F4E 2920
021C 0D                    BYTE   >0D
021D 494E 5345  MSG13      TEXT   'INSERT A NEW DISKETTE INTO DRIVE NO. 1'
0221 5254 2041
0225 204E 4557
0229 2044 4953
022D 4B45 5454
0231 4520 494E
0235 544F 2044
0239 5249 5645
023D 204E 4F2E
0241 2031
0243 0D                    BYTE   >0D
0244 494E 5345  MSG14      TEXT   'INSERT THE PATIENT FILE INTO DRIVE NO. 1'
0248 5254 2054
024C 4845 2050
0250 4154 4945
0254 4E54 2046
0258 494C 4520
025C 494E 544F
0260 2044 5249
0264 5645 204E
0268 4F2E 2031
026C 0D                    BYTE   >0D
026D 454E 5445  MSG15      TEXT   'ENTER THE PATIENT LAST NAME: '
0271 5220 5448
0275 4520 5041
0279 5449 454E
027D 5420 4C41
0281 5354 204E
0285 414D 453A
0289 20
028A 0D                    BYTE   >0D
028B 0A         MSG16      BYTE   >0A
028C 434F 4E54             TEXT   'CONTINUE WITH ANOTHER FUNCTION? (Y/N) '
0290 494E 5545
0294 2057 4954
0298 4820 414E
029C 4F54 4845
02A0 5220 4655
02A4 4E43 5449
02A8 4F4E 3F20
02AC 2859 2F4E
02B0 2920
02B2 0D                    BYTE   >0D
02B3 0A         MSG20      BYTE   >0A
02B4 5355 424A             TEXT   'SUBJECT NAME: '
02B8 4543 5420
02BC 4E41 4D45
02C0 3A20
02C2 0D                    BYTE   >0D
```

PAGE-6 GAIT DATA ACQUISITION - PART 1 (VER 08/09/81)

```
02C3 4147 453A  MSG21   TEXT  'AGE: '
02C7 20
02C8 0D                 BYTE  >0D
02C9 4845 4947  MSG22   TEXT  'HEIGHT: '
02CD 4854 3A20
02D1 0D                 BYTE  >0D
02D2 5745 4947  MSG23   TEXT  'WEIGHT: '
02D6 4854 3A20
02DA 0D                 BYTE  >0D
02DB 5449 4249  MSG24   TEXT  'TIBIA LENGTH: '
02DF 4120 4C45
02E3 4E47 5448
02E7 3A20
02E9 0D                 BYTE  >0D
02EA 4645 4D55  MSG25   TEXT  'FEMURAL LENGTH: '
02EE 5241 4C20
02F2 4C45 4E47
02F6 5448 3A20
02FA 0D                 BYTE  >0D
02FB 4241 434B  MSG26   TEXT  'BACK LENGTH: '
02FF 204C 454E
0303 4754 483A
0307 20
0308 0D                 BYTE  >0D
0309 4855 4D45  MSG27   TEXT  'HUMERAL LENGTH: '
030D 5241 4C20
0311 4C45 4E47
0315 5448 3A20
0319 0D                 BYTE  >0D
031A 5241 4449  MSG28   TEXT  'RADIAL LENGTH: '
031E 414C 204C
0322 454E 4754
0326 483A 20
0329 0D                 BYTE  >0D
032A 4849 5020  MSG29   TEXT  'HIP WIDTH: '
032E 5749 4454
0332 483A 20
0335 0D                 BYTE  >0D
0336 0A         MSG30   BYTE  >0A
0337 5255 4E20          TEXT  'RUN NO.  DATE '
033B 4E4F 2E20
033F 2044 4154
0343 4520
0345 0D                 BYTE  >0D
0346 3D3D 3D3D  MSG31   TEXT  '=======  ==== '
034A 3D3D 3D20
034E 203D 3D3D
0352 3D20
0354 0D                 BYTE  >0D
0355 0A         MSG32   BYTE  >0A
0356 434F 4D4D          TEXT  'COMMENT FOR RUN NO. '
035A 454E 5420
035E 464F 5220
0362 5255 4E20
0366 4E4F 2E20
036A 0D                 BYTE  >0D
036B 4C49 4E45  MSG33   TEXT  'LINE 1 : '
036F 2031 203A
```

PAGE-7 GAIT DATA ACQUISITION - PART 1 (VER 08/09/81)

```
0373 20
0374 0D                  BYTE   >0D
0375 4C49 4E45  MSG34    TEXT   'LINE 2 : '
0379 2032 203A
037D 20
037E 0D                  BYTE   >0D
037F 4C49 4E45  MSG35    TEXT   'LINE 3 : '
0383 2033 203A
0387 20
0388 0D                  BYTE   >0D
0389 0A         MSG40    BYTE   >0A
038A 2A2A 2A20           TEXT   '***  SAMPLE AND STORE DATA  *** '
038E 2053 414D
0392 504C 4520
0396 414E 4420
039A 5354 4F52
039E 4520 4441
03A2 5441 2020
03A6 2A2A 2A20
03AA 0D                  BYTE   >0D
03AB 0A         MSG41    BYTE   >0A
03AC 494E 5345           TEXT   'INSERT A NEW DISKETTE INTO DRIVE NO. 2'
03B0 5254 2041
03B4 204E 4557
03B8 2044 4953
03BC 4B45 5454
03C0 4520 494E
03C4 544F 2044
03C8 5249 5645
03CC 204E 4F2E
03D0 2032
03D2 0D                  BYTE   >0D
03D3 0A         MSG42    BYTE   >0A
03D4 5055 5420           TEXT   'PUT THE INITIALIZED DISKETTE BACK TO DRIVE NO.
1'
03D8 5448 4520
03DC 494E 4954
03E0 4941 4C49
03E4 5A45 4420
03E8 4449 534B
03EC 4554 5445
03F0 2042 4143
03F4 4B20 544F
03F8 2044 5249
03FC 5645 204E
0400 4F2E 2031
0404 0D                  BYTE   >0D
0405 4F4B 3F    MSG43    TEXT   'OK?'
0408 0D                  BYTE   >0D
0409 3D3D 3D3D  MSG50    TEXT   '======================'
040D 3D3D 3D3D
0411 3D3D 3D3D
0415 3D3D 3D3D
0419 3D3D 3D3D
041D 3D3D 3D
0420 0A0D                DATA   >0A0D
                *
                * ROUTINE: INPUT
                * THIS ROUTINE INPUTS A LINE FROM THE TERMINAL AND STORES IT
```

```
                    * INTO THE TEXT BUFFER.
                    *
0422 0200 0140 INPUT   LI    R0,BUFFER       ; GET TEXT BUFFER
0426 2C41      INLP    IN    R1              ; GET ONE CHARACTER
0428 DC01              MOVB  R1,*R0+         ; STORE TO THE TEXT BUFFER
042A 9801 0004         CB    R1,@CR          ; INPUT FINISHED?
042E 16FB              JNE   INLP            ; NO, CONTINUE
0430 D420 0004         MOVB  @CR,*R0         ; SET UP THE ENDING MARK
0434 045B              B     *R11            ; EXIT
                    *
                    * ROUTINE: MSGOUT
                    * THIS ROUTINE DISPLAYS THE MESSAGE WHOSE ADDRESS
                    * FOLLOWS THE CALLS.
                    *
0436 2CA0 0004 MSGOUT  OUT   @CR             ; OUTPUT A CARRIAGE RETURN
043A 2CA0 0005         OUT   @LF             ; OUTPUT A LINE FEED
043E 2C90      MSGLP2  OUT   *R0             ; OUTPUT ONE CHARACTER
0440 0580              INC   R0              ; ADVANCE THE POINTER ONCE
0442 9810 0004         CB    *R0,@CR         ; END OF MESSAGE?
0446 16FB              JNE   MSGLP2          ; NO, CONTINUE
0448 045B              B     *R11            ; EXIT
                    *
                    * ROUTINE: DTH
                    * THIS ROUTINE SETS UP THE PARAMETERS REQUIRED FOR
                    * CALLING A DTH SUBROUTINE; IT THEN BRANCHES TO ONE
                    * OF THE SUBROUTINES DEPENDING ON THE PARAMETER SET
                    * BY THE CALLING ROUTINE.
                    *
044A C03B      DTH     MOV   *R11+,R0        ; GET ADDRESS OF THE COMMAND
044C 0201 0140         LI    R1,BOUND1       ; SET UP LOWER BOUND FOR DTH
0450 0202 01C0         LI    R2,BOUND2       ; SET UP UPPER BOUND
0454 0420 F804         BLWP  @#F804          ; GO TO DTH'S SUBROUTINE
0458 0000              DATA  0
045A 045B              B     *R11            ; EXIT
                    *
                    * ROUTINE: SETUP
                    * THIS ROUTINE INCLUDES ALL THE SUBROUTINES WHICH SET UP
                    * THE NECESSARY PARAMETERS FOR DTH CALLS.
                    *
045C C03B      SETUP1  MOV   *R11+,R0
045E DE90              MOVB  *R0,*R10+
0460 DEA0 0009         MOVB  @BLANK,*R10+
0464 045B              B     *R11
0466 C03B      SETUP2  MOV   *R11+,R0
0468 04C1              CLR   R1              ; LENGTH COUNTER=0
046A DEB0      SLP     MOVB  *R0+,*R10+      ; SET UP FILE NAME
046C 9810 0004         CB    *R0,@CR         ; ENDING MARK?
0470 16FC              JNE   SLP             ; NO, CONTINUE
0472 045B              B     *R11            ; EXIT
                    *
                    * ROUTINE: OPEN
                    * THIS ROUTINE OPENS FILE 1 AS AN ACTIVE I/O FILE.
                    * THE NAME OF FILE 1 FOLLOWS THE CALLING INSTRUCTION
                    * IN THE INVOKING ROUTINE.
                    *
0474 0200 01E2 OPEN    LI    R0,FNAME        ; GET THE FILE NAME
0478 0201 01C2         LI    R1,DTHCMD       ; GET COMMAND FILE
047C C24B              MOV   R11,R9          ; SAVE RETURN ADDRESS
```

```
047E DC60 000C          MOVB   @DU,*R1+          ; SET UP DU COMMAND
0482 DC60 0009          MOVB   @BLANK,*R1+       ; LEAVE A BLANK
0486 DC60 0014          MOVB   @ONE,*R1+         ; FILE 1
048A DC60 0009          MOVB   @BLANK,*R1+       ; LEAVE ANOTHER BLANK
048E DC70        OPENLP MOVB   *R0+,*R1+         ; SET UP FILE NAME
0490 9810 0004          CB     *R0,@CR           ; END OF FILE NAME?
0494 16FC               JNE    OPENLP            ; NO, CONTINUE
0496 DC60 0004          MOVB   @CR,*R1+          ; PLACE AN ENDING LABEL
049A 06A0 044A          BL     @DTH              ; CALL DTH FOR DU COMMAND
049E 01C2               DATA   DTHCMD
04A0 0459               B      *R9               ; EXIT
                 *
                 * ROUTINE: MOD
                 * THIS ROUTINE MODULOES R6 BY 6 AND ADDS ONE TO R6.
                 *
04A2 0205 0006   MOD    LI     R5,6              ; MODULO 6
04A6 8146        MOD1   C      R6,R5             ; R6 < 6 ?
04A8 1102               JLT    MODRTN            ; YES, GO TO MODRTN
04AA 6185               S      R5,R6             ; SUBTRACT 6 FROM R6
04AC 10FC               JMP    MOD1              ; TEST AGAIN
04AE 0A16        MODRTN SLA    R6,1              ; MULTIPLY R6 BY 2
04B0 045B               B      *R11              ; EXIT
                 *
                 * ROUTINE: WRAPUP
                 * THIS ROUTINE READS THE PREVIOUS RUN NO. FROM
                 * SECTOR 11 OF THE MASTER FILE; IT THEN UPDATES
                 * THE RUN NO. AND STORES THE DATE ASSOCIATED WITH
                 * THE NEW RUN NO. BACK TO THE MASTER FILE.  FINALLY
                 * IT QUERIES THE OPERATOR FOR ANY COMMENTS NEEDED
                 * TO BE STORED.
                 *
04B2 C1CB        WRAPUP MOV    R11,R7            ; SAVE RETURN ADDRESS
04B4 06A0 044A          BL     @DTH              ; POSN FILE 1 TO SECTOR 10
04B8 0084               DATA   DTH06
04BA 06A0 044A          BL     @DTH              ; READ SECTOR 10 FOR RUN NO.
04BE 0070               DATA   DTH02
04C0 C1A0 0140          MOV    @BUFFER,R6        ; GET PREV. RUN NO.
04C4 0586               INC    R6                ; UPDATE RUN NO.
04C6 C806 01FC          MOV    R6,@RNUM          ; STORE PREV. RUN NO.
04CA C806 0140          MOV    R6,@BUFFER        ; PUT CUR RUN NO TO BUFFER
04CE 06A0 044A          BL     @DTH              ; POSN FILE 1 TO SECTOR 10
04D2 0084               DATA   DTH06
04D4 06A0 044A          BL     @DTH              ; STORE CURR NUM NUM TO DISK
04D8 0078               DATA   DTH03
04DA 0606               DEC    R6                ; GET PREV. RUN NO.
04DC 06A0 04A2          BL     @MOD              ; MOD(R6,7)
04E0 0201 01C2          LI     R1,DTHCMD         ; GET COMMAND FILE
04E4 0202 0095          LI     R2,DTH10          ; R2 POINTS TO DTH10
04E8 DC72        WLOOP1 MOVB   *R2+,*R1+         ; MOVE ONE CHARACTER
04EA 9812 0004          CB     *R2,@CR           ; END OF COMMAND?
04EE 16FC               JNE    WLOOP1            ; NO, CONTINUE
04F0 CC66 0048          MOV    @LABEL2(R6),*R1+  ; GET SECTOR NO.
04F4 D460 0004          MOVB   @CR,*R1           ; PLACE ENDING LABEL
04F8 06A0 044A          BL     @DTH              ; POSN FILE 1
04FC 01C2               DATA   DTHCMD
04FE 0200 0140          LI     R0,BUFFER         ; GET TEXT BUFFER
0502 0201 01FE          LI     R1,RNUM1          ; R1 POINTS TO RNUM1
0506 CC20 0022          MOV    @BLANKS,*R0+      ; LEAVE TWO BLANKS
```

PAGE-10 GAIT DATA ACQUISITION - PART 1 (VER 08/09/81)

```
050A 04C4                CLR    R4                ; CLEAR R4 FOR DIVISION
050C C160 01FC           MOV    @RNUM,R5          ; PUT RUN NO IN R5
0510 3D20 0020           DIV    @HUNDRD,R4        ; DIV R4,R5 BY 100
0514 DC24 0013           MOVB   @ZERO(R4),*R0+    ; GET HUNDREDTH
0518 DC64 0013           MOVB   @ZERO(R4),*R1+    ; GET HUNDREDTH FOR RNUM1
051C 04C4                CLR    R4                ; CLEAR R4 FOR DIVISION
051E 3D20 001E           DIV    @TEN,R4           ; DIV R4,R5 BY 10
0522 DC24 0013           MOVB   @ZERO(R4),*R0+    ; GET TENTH
0526 DC64 0013           MOVB   @ZERO(R4),*R1+    ; GET TENTH FOR RNUM1
052A DC25 0013           MOVB   @ZERO(R5),*R0+    ; GET UNITH
052E DC65 0013           MOVB   @ZERO(R5),*R1+    ; GET UNITH FOR RNUM1
0532 D460 0004           MOVB   @CR,*R1           ; PLACE ENDING LABEL
0536 DC20 0009           MOVB   @BLANK,*R0+       ; LEAVE ONE BLANK
053A CC20 0022           MOV    @BLANKS,*R0+      ; LEAVE TWO BLANKS
053E 0201 01F2           LI     R1,DATE1          ; R1 POINTS TO DD/MM/YY
0542 DC31         WLOOP2 MOVB   *R1+,*R0+         ; MOVE ONE CHARACTER
0544 9811 0004           CB     *R1,@CR           ; END OF DATE?
0548 16FC                JNE    WLOOP2            ; NO, CONTINUE
054A D420 0004           MOVB   @CR,*R0           ; PLACE ENDING MARK
054E 06A0 044A           BL     @DTH              ; OUTPUT ONE RECORD
0552 0078                DATA   DTH03
0554 0200 01C2           LI     R0,DTHCMD         ; R0 POINTS TO DTHCMD
0558 0201 0095           LI     R1,DTH10          ; R1 POINTS TO DTH10
055C DC31         WLOOP3 MOVB   *R1+,*R0+         ; MOVE ONE CHARACTER
055E 9811 0004           CB     *R1,@CR           ; END OF COMMAND?
0562 16FC                JNE    WLOOP3            ; NO, CONTINUE
0564 CC26 0054           MOV    @LABEL3(R6),*R0+  ; GET SECTOR NUMBER
0568 D420 0004           MOVB   @CR,*R0           ; PLACE ENDING LABEL
056C 06A0 044A           BL     @DTH              ; POSITION FILE 1
0570 01C2                DATA   DTHCMD
0572 0200 01C2           LI     R0,DTHCMD         ; R0 POINTS TO DTHCMD
0576 DC20 0012           MOVB   @SF,*R0+          ; SETUP COMMAND KEY
057A DC20 0009           MOVB   @BLANK,*R0+       ; LEAVE A BLANK
057E 0201 01E2           LI     R1,FNAME          ; R1 POINTS TO FNAME
0582 DC31                MOVB   *R1+,*R0+         ; GET THE 1ST LETTER
0584 DC31                MOVB   *R1+,*R0+         ; GET THE 2ND LETTER
0586 0201 01EB           LI     R1,DATE           ; R1 POINTS TO DATE
058A DC31         LOOP01 MOVB   *R1+,*R0+         ; GET ONE DIGIT OF THE DATE
058C 9811 0004           CB     *R1,@CR           ; END OF DATE?
0590 16FC                JNE    LOOP01            ; NO, CONTINUE
0592 0201 01FE           LI     R1,RNUM1          ; GET ASCII RUN NO.
0596 DC31         LOOP02 MOVB   *R1+,*R0+         ; GET ONE DIGIT AT A TIME
0598 9811 0004           CB     *R1,@CR           ; END OF RUN NO.?
059C 16FC                JNE    LOOP02            ; NO, GET ONE MORE
059E DC20 0009           MOVB   @BLANK,*R0+       ; LEAVE A BLANK
05A2 DC20 0004           MOVB   @CR,*R0+          ; PUT ENDING MARK
05A6 0200 01C2           LI     R0,DTHCMD         ; GET COMMAND
05AA 0201 2000           LI     R1,>2000          ; LOWER BOUND
05AE 0202 7FFF           LI     R2,>7FFF          ; UPPER BOUND
05B2 0420 F804           BLWP   @#F804            ; CALL DTH SUB-PROGRAM
05B6 0000                DATA   0                 ; ANY ERROR IS FATAL
05B8 0200 0355           LI     R0,MSG32          ; R0 POINTS TO MSG32
05BC 0201 0140           LI     R1,BUFFER         ; R1 POINTS TO TEXT BUFFER
05C0 DC70         WLOOP4 MOVB   *R0+,*R1+         ; MOVE ONE CHARACTER
05C2 9810 0004           CB     *R0,@CR           ; END OF STRING?
05C6 16FC                JNE    WLOOP4            ; NO, CONTINUE
05C8 0200 01FE           LI     R0,RNUM1          ; R0 POINTS TO RNUM1
05CC DC70         WLOOP5 MOVB   *R0+,*R1+         ; MOVE ONE CHARACTER
```

```
05CE 9810 0004          CB    *R0,@CR        ; END OF RUN NUMBER?
05D2 16FC               JNE   WLOOP5         ; NO, CONTINUE
05D4 D460 0004          MOVB  @CR,*R1        ; PUT ENDING LABEL
05D8 0200 0140          LI    R0,BUFFER      ; R0 POINTS TO TEXT BUFFER
05DC 06A0 0436          BL    @MSGOUT        ; OUTPUT TEXT BUFFER
05E0 06A0 044A          BL    @DTH           ; STORE TEXT TO DISK
05E4 0078               DATA  DTH03
05E6 0200 0409          LI    R0,MSG50
05EA 06A0 0436          BL    @MSGOUT        ; OUTPUT MSG50
05EE 0204 0060          LI    R4,LABEL4      ; R4 POINTS TO LABEL4
05F2 C034       WLOOP6  MOV   *R4+,R0        ; GET MESSAGE POINTER
05F4 1308               JEQ   WFIN           ; IF END OF MSG GOTO WFIN
05F6 06A0 0436          BL    @MSGOUT        ; OUTPUT MSG POINTED BY R0
05FA 06A0 0422          BL    @INPUT         ; GET USER'S ENTRY
05FE 06A0 044A          BL    @DTH           ; STORE TO DISK
0602 0078               DATA  DTH03
0604 10F6               JMP   WLOOP6         ; AGAIN
0606 0457       WFIN    B     *R7            ; EXIT
                *
                * ROUTINE: DATACQ
                * THIS ROUTINE PERFORMS DATA ACQUISITION AND UPDATES
                * THE RUN NUMBER IN THE PATIENT MASTER FILE.  IF
                * THE PATIENT IS NEW TO THE SYSTEM, A NEW MASTER FILE
                * FOR THE PATIENT WILL BE CREATED.
                *
0608 C20B       DATACQ  MOV   R11,R8         ; SAVE RETURN ADDRESS
060A 0200 019E          LI    R0,MSG09       ; R0 POINTS TO MSG09
060E 06A0 0436          BL    @MSGOUT        ; OUTPUT MSG09
0612 0200 01F4          LI    R0,MSG12       ; R0 POINTS TO MSG12
0616 06A0 0436          BL    @MSGOUT        ; OUTPUT MSG12
061A 06A0 0422          BL    @INPUT         ; GET REPLY
061E 9820 0140          CB    @BUFFER,@YES   ; NEW PATIENT?
0622 0008
0624 1650               JNE   OLD            ; NO, GO TO OLD
0626 0200 021D          LI    R0,MSG13       ; R0 POINTS TO MSG13
062A 06A0 0436          BL    @MSGOUT        ; OUTPUT MSG13
062E 020A 01C2          LI    R10,DTHCMD     ; R10 POINTS TO DTHCMD
0632 06A0 045C          BL    @SETUP1        ; SET UP COMMAND
0636 000D               DATA  ID
0638 06A0 0466          BL    @SETUP2        ; SET UP FILE NAME FOR ID
063C 01E2               DATA  FNAME          ; FILE NAME
063E D6A0 0004          MOVB  @CR,*R10       ; PUT ENDING MARK
0642 06A0 044A          BL    @DTH           ; CALL DTH FOR ID COMMAND
0646 01C2               DATA  DTHCMD         ; ADDRESS FOR COMMAND
0648 020A 01C2          LI    R10,DTHCMD     ; R10 POINTS TO DTHCMD
064C 06A0 045C          BL    @SETUP1        ; SET UP AF COMMAND
0650 000A               DATA  AF
0652 06A0 0466          BL    @SETUP2        ; SET UP FILE NAME
0656 01E2               DATA  FNAME
0658 DEA0 0009          MOVB  @BLANK,*R10+   ; LEAVE A BLANK
065C DEA0 0017          MOVB  @FOUR,*R10+    ; SET THE FILE TO 44 SECTORS
0660 DEA0 0017          MOVB  @FOUR,*R10+
0664 D6A0 0004          MOVB  @CR,*R10       ; SET UP THE ENDING MARK
0668 06A0 044A          BL    @DTH           ; CALL DTH FOR AF COMMAND
066C 01C2               DATA  DTHCMD
066E 06A0 0474          BL    @OPEN          ; OPEN FILE NO. 1
0672 06A0 044A          BL    @DTH           ; CALL DTH FOR PF COMMAND
0676 0068               DATA  DTH01
```

```
0678 0204 0032           LI    R4,LABEL1        ; R4 POINTS TO LABEL TABLE
067C 04C5                CLR   R5               ; INITIALIZE COUNTER
067E C034      DLOOP1    MOV   *R4+,R0          ; POSITION TO PROPER LABEL
0680 06A0 0436           BL    @MSGOUT          ; OUTPUT MSG POINTED BY LABEL
0684 06A0 0422           BL    @INPUT           ; GET OPERATOR'S ENTRY
0688 06A0 044A           BL    @DTH             ; CALL DTH TO OUTPUT 1 RECORD
068C 0078                DATA  DTH03
068E 0585                INC   R5
0690 0285 000A           CI    R5,10            ; END OF LABEL TABLE?
0694 11F4                JLT   DLOOP1           ; NO. CONTINUE
0696 06A0 044A DLOOP2    BL    @DTH             ; INITIALIZE OTHER SECTORS
069A 0078                DATA  DTH03
069C 0585                INC   R5               ; UPDATE COUNTER
069E 0285 002C           CI    R5,44            ; END OF FILE?
06A2 11F9                JLT   DLOOP2           ; NO, CONTINUE
06A4 06A0 044A           BL    @DTH             ; POSN FILE 1 TO SECTOR 10
06A8 0084                DATA  DTH06
06AA 04C0                CLR   R0               ; INITIALIZE RUN NO.
06AC C800 0140           MOV   R0,@BUFFER       ; STORE RUN NO. TO BUFFER
06B0 D820 0004           MOVB  @CR,@BUFFER+2    ; SET UP ENDING LABEL
06B4 0142
06B6 06A0 044A           BL    @DTH             ; STORE RUN NO. TO FILE
06BA 0078                DATA  DTH03            ; OUTPUT A RECORD
06BC 0420 0000           BLWP  @SAMPLE          ; SAMPLE AND STORE DATA
06C0 06A0 04B2           BL    @WRAPUP          ; SET DATE AND COMMENTS
06C4 105F                JMP   DEXIT            ; GO TO EXIT
06C6 0200 0244 OLD       LI    R0,MSG14         ; R0 POINTS TO MSG14
06CA 06A0 0436           BL    @MSGOUT          ; OUTPUT MSG14
06CE 06A0 0474           BL    @OPEN            ; OPEN FILE NO. 1
06D2 06A0 044A           BL    @DTH             ; POSITION TO SECTOR 10
06D6 0084                DATA  DTH06
06D8 06A0 044A           BL    @DTH             ; READ PREV RUN NO.
06DC 0070                DATA  DTH02
06DE C1A0 0140           MOV   @BUFFER,R6       ; PLACE PREV RUN NO. TO R6
06E2 0205 0006 OLDLP1    LI    R5,6             ; MAXIMUM 6 RUNS
06E6 8146                C     R6,R5            ; GREATER THAN 6
06E8 1202                JLE   OLDFIN           ; NO, GO TO OLDFIN
06EA 6185                S     R5,R6            ; YES, SUBTRACT 6 FROM R6
06EC 10FA                JMP   OLDLP1           ; TEST AGAIN
06EE 8146      OLDFIN    C     R6,R5            ; R6=6 ?
06F0 1645                JNE   OLD01            ; NO, GO TO OLD01
06F2 0200 03AB           LI    R0,MSG41         ; R0 POINTS TO MSG41
06F6 06A0 0436           BL    @MSGOUT          ; OUTPUT MSG41
06FA 06A0 044A           BL    @DTH             ; CLOSE ALL FILES
06FE 0082                DATA  DTH05
0700 06A0 044A           BL    @DTH             ; RESET DISKS
0704 0080                DATA  DTH04
0706 020A 01C2           LI    R10,DTHCMD       ; R10 POINTS TO DTHCMD
070A 06A0 045C           BL    @SETUP1          ; SET UP ID COMMAND
070E 000D                DATA  ID
0710 06A0 0466           BL    @SETUP2          ; PLACE FILE NAME
0714 01E2                DATA  FNAME
0716 DEA0 0007           MOVB  @SLASH,*R10+     ; PUT A SLASH
071A DEA0 0015           MOVB  @TWO,*R10+       ; DRIVE NO. 2
071E D6A0 0004           MOVB  @CR,*R10         ; ENDING LABEL
0722 06A0 044A           BL    @DTH             ; INITIALIZE A NEW DISK
0726 01C2                DATA  DTHCMD
0728 020A 01C2           LI    R10,DTHCMD       ; R10 POINTS TO DTHCMD
```

```
072C 06A0 045C          BL     @SETUP1            ; SETUP CF COMMAND
0730 000B               DATA   CF
0732 06A0 0466          BL     @SETUP2            ; FIRST FILE NAME
0736 01E2               DATA   FNAME
0738 DEA0 0007          MOVB   @SLASH,*R10+       ; PLACE A SLASH
073C DEA0 0013          MOVB   @ZERO,*R10+        ; DISK DRIVE NO. 0/1
0740 DEA0 0009          MOVB   @BLANK,*R10+       ; LEAVE A BLANK
0744 06A0 0466          BL     @SETUP2            ; SECOND FILE NAME
0748 01E2               DATA   FNAME
074A DEA0 0007          MOVB   @SLASH,*R10+       ; LEAVE A SLASH
074E DEA0 0015          MOVB   @TWO,*R10+         ; DRIVE NO. 2
0752 D6A0 0004          MOVB   @CR,*R10           ; ENDING LABEL
0756 06A0 044A          BL     @DTH               ; GO TO CF COMMAND
075A 01C2               DATA   DTHCMD
075C 0200 03D3          LI     R0,MSG42           ; R0 POINTS TO MSG42
0760 06A0 0436          BL     @MSGOUT            ; OUTPUT MSG42
0764 0200 0405 OLDLP2   LI     R0,MSG43           ; R0 POINTS TO MSG43
0768 06A0 0436          BL     @MSGOUT
076C 06A0 0422          BL     @INPUT             ; GET USER'S RESPONSE
0770 9820 0140          CB     @BUFFER,@YES       ; READY?
0774 00C8
0776 16F6               JNE    OLDLP2             ; WAIT AGAIN
0778 06A0 0474          BL     @OPEN              ; OPEN FILE 1
077C 0420 06BE OLD01    BLWP   @SAMPLE            ; SAMPLE AND STORE DATA
0780 06A0 04B2          BL     @WRAPUP            ; SET DATE AND COMMENTS
0784 0458      DEXIT    B      *R8                ; EXIT
               *
               * MAIN PROGRAM: MAIN
               * THIS IS A BASIC MONITOR WHICH PERFORMS THE
               * INITIALIZATION AND QUERIES THE OPERATOR FOR
               * A DESIRED FUNCTION.  IT GATHERS PARAMETERS AND
               * TRANSFERS CONTROL TO AN APPROPRIATE ROUTINE.
               *
0786 02E0 0100 MAIN     LWPI   MAINWP             ; INITIALIZE WORK SPACE
078A 0200 009A          LI     R0,MSG01           ; R0 POINTS TO MSG01
078E 06A0 0436          BL     @MSGOUT            ; OUTPUT MSG01
0792 0200 00C4          LI     R0,MSG02           ; R0 POINTS TO MSG02
0796 06A0 0436          BL     @MSGOUT            ; OUTPUT MSG02
079A 06A0 0422          BL     @INPUT             ; GET REPLY
079E 0200 01EB          LI     R0,DATE            ; GET DATE BUFFER
07A2 0202 0140          LI     R2,BUFFER          ; POINT TO INPUT BUFFER
07A6 DC32      MLOOP1   MOVB   *R2+,*R0+          ; MOVE ONE CHARACTER
07A8 9812 0007          CB     *R2,@SLASH         ; A SLASH?
07AC 1601               JNE    NEXT               ; YES, GO TO NEXT
07AE 0582               INC    R2                 ; UPDATE COUNTER
07B0 9812 0004 NEXT     CB     *R2,@CR            ; END OF DATE?
07B4 16F8               JNE    MLOOP1             ; NO, CONTINUE
07B6 D420 0004          MOVB   @CR,*R0            ; PLACE ENDING LABEL
07BA 0200 01F2          LI     R0,DATE1           ; POINT TO DD/MM/YY
07BE 0201 0140          LI     R1,BUFFER          ; POINT TO INPUT BUFFER
07C2 DC31      MLOOP2   MOVB   *R1+,*R0+          ; MOVE ONE CHARACTER
07C4 9811 0004          CB     *R1,@CR            ; END OF DD/MM/YY?
07C8 16FC               JNE    MLOOP2             ; NO, CONTINUE
07CA D420 0004          MOVB   @CR,*R0            ; PLACE ENDING LABEL
07CE 0200 00DC          LI     R0,MSG03           ; R0 POINTS TO MSG03
07D2 06A0 0436          BL     @MSGOUT            ; OUTPUT MSG03
07D6 0200 00F3          LI     R0,MSG04           ; R0 POINTS TO MSG04
07DA 06A0 0436          BL     @MSGOUT            ; OUTPUT MSG04
```

```
07DE 0200 0110           LI     R0,MSG05        ; R0 POINTS TO MSG05
07E2 06A0 0436           BL     @MSGOUT         ; OUTPUT MSG05
07E6 0200 0134           LI     R0,MSG06        ; R0 POINTS TO MSG06
07EA 06A0 0436           BL     @MSGOUT         ; OUTPUT MSG06
07EE 0200 0152  MAIN01   LI     R0,MSG07        ; R0 POINTS TO MSG07
07F2 06A0 0436           BL     @MSGOUT         ; OUTPUT MSG07
07F6 06A0 0422  MLOOP4   BL     @INPUT          ; GET REPLY
07FA C020 0140           MOV    @BUFFER,R0      ; STORE REPLY
07FE 0980                SRL    R0,8            ; RIGHT JUSTIFY
0800 0201 0024           LI     R1,TABCMD       ; POINT TO TABLE OF COMMANDS
0804 C231       SEARCH   MOV    *R1+,R8         ; SAVE ADDRESS PTR IN R8
0806 1325                JEQ    ERROR           ; ERROR IF TABLE EXHAUTED
0808 8C40                C      R0,*R1+         ; COMPARE TO USER'S ENTRY
080A 16FC                JNE    SEARCH          ; NOT MATCHED, SERACH AGAIN
080C 0200 026D           LI     R0,MSG15        ; R0 POINTS TO MSG15
0810 06A0 0436           BL     @MSGOUT         ; OUTPUT MSG15
0814 06A0 0422           BL     @INPUT          ; GET REPLY
0818 0200 01E2           LI     R0,FNAME        ; R0 POINTS TO FILE NAME
081C 0201 01EB           LI     R1,DATE         ; PTR FOR END OF FILE NAME
0820 0202 0140           LI     R2,BUFFER       ; R2 POINTS TO BUFFER
0824 DC32       MLOOP3   MOVB   *R2+,*R0+       ; MOVE ONE CHARACTER
0826 9812 0004           CB     *R2,@CR         ; END OF FILE NAME?
082A 1302                JEQ    MNEXT1          ; YES, GO TO MNEXT1
082C 8040                C      R0,R1           ; NO, CHECK MAX LENGTH
082E 11FA                JLT    MLOOP3          ; CONT IF MAX NOT REACHED
0830 DC20 0004  MNEXT1   MOVB   @CR,*R0+        ; SET UP ENDING MARK
0834 0698                BL     *R8             ; GO TO CHOICE
0836 06A0 044A           BL     @DTH            ; CLOSE ALL FILES
083A 0082                DATA   DTH05
083C 0200 028B           LI     R0,MSG16        ; R0 POINTS TO MSG16
0840 06A0 0436           BL     @MSGOUT         ; OUTPUT MSG16
0844 06A0 0422           BL     @INPUT          ; GET OPERATOR'S REPLY
0848 9820 0140           CB     @BUFFER,@YES    ; COMPARE TO USER'S ENTRY
084C 0008
084E 13CF                JEQ    MAIN01          ; IF YES, START AGAIN
0850 2C00                XOP    0,0             ; NO, RETURN TO MONITOR
0852 0200 017E  ERROR    LI     R0,MSG08        ; R0 POINTS TO MSG08
0856 06A0 0436           BL     @MSGOUT         ; OUTPUT MSG08
085A 10CD                JMP    MLOOP4          ; START AGAIN
085C 00                  END    MAIN            ; END OF PROGRAM
```

```
 000A LP  AF        0009 LP  BLANK     0022 LP  BLANKS    0140 LA  BOUND1
 01C0 LA  BOUND2    0140 DA  BUFFER    000B LP  CF        0004 DP  CR
 0608 LP  DATACQ   *0120 DA  DATAWP    01EB LA  DATE      01F2 LA  DATE1
 0784 LP  DEXIT     067E LP  DLOOP1    0696 LP  DLOOP2    044A DP  DTH
 0068 DP  DTH01     0070 DP  DTH02     0078 DP  DTH03     0080 LP  DTH04
 0082 LP  DTH05     0084 LP  DTH06    *008B DP  DTH07     0095 LP  DTH10
 01C2 LA  DTHCMD    000C LP  DU       *001B LP  EIGHT    *0006 LP  ENSIGN
 0852 LP  ERROR    *0018 LP  FIVE      01E2 LA  FNAME     0017 LP  FOUR
 0020 LP  HUNDRD    000D LP  ID        0426 LP  INLP      0422 DP  INPUT
*000E LP  IR        0032 DP  LABEL1    0048 LP  LABEL2    0054 LP  LABEL3
 0060 DP  LABEL4    0005 LP  LF        002C RP  LIST      058A LP  LOOP01
 0596 LP  LOOP02    0786 LP  MAIN      07EE LP  MAIN01    0100 LA  MAINWP
 07A6 LP  MLOOP1    07C2 LP  MLOOP2    0824 LP  MLOOP3    07F6 LP  MLOOP4
 0830 LP  MNEXT1    04A2 DP  MOD       04A6 LP  MOD1      04AE LP  MODRTN
```

PAGE-15 GAIT DATA ACQUISITION - PART 1 (VER 08/09/81)

```
 009A LP   MSG01    00C4 LP   MSG02    00DC LP   MSG03    00F3 LP   MSG04
 0110 LP   MSG05    0134 LP   MSG06    0152 LP   MSG07    017E LP   MSG08
 019E LP   MSG09   *01BA DP   MSG10   *01D8 DP   MSG11    01F4 LP   MSG12
 021D LP   MSG13    0244 LP   MSG14    026D LP   MSG15    028B LP   MSG16
 02B3 LP   MSG20    02C3 LP   MSG21    02C9 LP   MSG22    02D2 LP   MSG23
 02DB LP   MSG24    02EA LP   MSG25    02FB LP   MSG26    0309 LP   MSG27
 031A LP   MSG28    032A LP   MSG29   *0336 DP   MSG30   *0346 DP   MSG31
 0355 LP   MSG32    036B LP   MSG33    0375 LP   MSG34    037F LP   MSG35
*0389 DP   MSG40    03AB LP   MSG41    03D3 LP   MSG42    0405 LP   MSG43
 0409 DP   MSG50    043E DP   MSGLP2   0436 DP   MSGOUT   07B0 LP   NEXT
*001C LP   NINE     06C6 LP   OLD      077C LP   OLD01    06EE LP   OLDFIN
 06E2 LP   OLDLP1   0764 LP   OLDLP2   0014 LP   ONE      0474 DP   OPEN
 048E LP   OPENLP  *000F LP   OR      *085C LA   PART1   *0010 LP   PF
 0000 LA   R0       0001 LA   R1       000A LA   R10      000B LA   R11
*000C LA   R12     *000D LA   R13     *000E LA   R14     *000F LA   R15
 0002 LA   R2      *0003 LA   R3       0004 LA   R4       0005 LA   R5
 0006 LA   R6       0007 LA   R7       0008 LA   R8       0009 LA   R9
*0011 LP   RD       01FC DA   RNUM     01FE LA   RNUM1    077E RP   SAMPLE
 0804 LP   SEARCH   045C DP   SETUP1   0466 DP   SETUP2  *001A LP   SEVEN
 0012 LP   SF      *0019 LP   SIX      0007 LP   SLASH    046A LP   SLP
*0000 LP   START    0024 LP   TABCMD   001E LP   TEN     *0016 LP   THREE
 0015 LP   TWO      0028 RP   UPDATE   0606 LP   WFIN     04E8 LP   WLOOP1
 0542 LP   WLOOP2   055C LP   WLOOP3   05C0 LP   WLOOP4   05CC LP   WLOOP5
 05F2 LP   WLOOP6   04B2 DP   WRAPUP   0008 LP   YES      0013 LP   ZERO
```

PAGE-1 GAIT DATA ACQUISITION - PART 2 (VER 27/04/82)

```
                     TITL  'GAIT DATA ACQUISITION - PART 2 (VER 27/04/82)'
06EC          PART2  IDT
                     DREG
              *
              * EXTERNAL REFERENCES
              *
                     REF   MSGOUT.MSG11.DTH.LABEL1.OPEN,DTH01,DTH02
                     REF   BUFFER.RNUM,MSG30,MSG31,LABEL4
                     REF   MSG10,MSG40,MSG50,MSGLP2,DTH03
                     REF   DTH07.INPUT.CR.DATAWP
              *
              * EXTERNAL DEFINITIONS
              *
                     DEF   LIST.UPDATE.SAMPLE
              *
              * RAM DATA BASE
              *
0202                 AORG  >202.
0202          COUNTL BSS   2            ; # OF THRESHOLDS
0204          COUNTC BSS   2            ; NUM OF CLOCK PULSES
0206          THRESH BSS   2            ; THRESHOLD LEVEL
0208          RUNPTR BSS   2            ; ADDR FOR DIFF RUN
020A          STEPTR BSS   2            ; STEP POINTER
020C          SMRATE BSS   2            ; SAMPLING RATE
020E          NUMSAM BSS   2            ; NUM OF SAMPLES
0210          NUMCHL BSS   2            ; NUM OF CHANNELS
0212          TRICHL BSS   2            ; TRIGGER CHANNEL
0214          DELAY  BSS   2            ; DELAY LOOP INDEX
              *
              * A/D INTERFACES
              *
E7F4          SETUP  EQU   >E7F4        ; SETUP A/D OPTIONS
E7F8          MUXADR EQU   >E7F8        ; MULTIPLEX ADDRESS
E7FA          CONVER EQU   >E7FA        ; CONVERSION COMMAND
E7FC          STATUS EQU   >E7FC        ; A/D STATUS
E7FE          ADCDAT EQU   >E7FE        ; A/D CONVERTED DATA
              *
              * ROM DATA BASE
              *
0000                 RORG  *
0000 59       YES    BYTE  'Y'          ; ASCII CODE FOR YES
0002 0200     HALF   DATA  >0200        ; HALF OF CLOCK +VE VOLTAGE
0004 0002     LIMIT1 DATA  2            ; THRESHOLDS TO START
0006 0020     LIMIT2 DATA  >20          ; THRESHOLDS TO STOP
0008 0023     WIDTH  DATA  >23          ; MIN. VALID SAMPLE WIDTH
000A 0000     NONE   DATA  0            ; NO OPTIONS
000C 0100     WANTED DATA  >0100        ; NUMBER OF SAMPLES WANTED
000E 1000     OFFSET DATA  >1000        ; STEP OFFSET
0010 2000     START  DATA  >2000        ; BUFFER STARTING PTR
              *
              * LABEL FILES
              *
0012 0000     LABEL5 DATA  MSG40
0014 0096            DATA  MSG60
0016 00B6            DATA  MSG62
0018 00D8            DATA  MSG63
001A 0100            DATA  MSG64
001C 011E            DATA  MSG65
```

```
001E 0000                 DATA   0
0020 0180        LABEL7   DATA   MSG101
0022 018A                 DATA   MSG102
0024 0194                 DATA   MSG103
0026 019E                 DATA   MSG104
0028 01A8                 DATA   MSG105
002A 01B2                 DATA   MSG106
002C 0000                 DATA   0
002E 0310        LABEL8   DATA   MSG206
0030 028D                 DATA   MSG203
0032 02B6                 DATA   MSG204
0034 02E0                 DATA   MSG205
0036 0000                 DATA   0
0038 032B        LABEL9   DATA   MSG207
003A 0348                 DATA   MSG208
003C 035E                 DATA   MSG209
003E 0376                 DATA   MSG210
0040 02E0                 DATA   MSG205
0042 0000                 DATA   0
0044 0001 0031   TABLE1   DATA   1,'1'
0048 0002 0032            DATA   2,'2'
004C 0003 0033            DATA   3,'3'
0050 0004 0034            DATA   4,'4'
0054 0005 0035            DATA   5,'5'
0058 0006 0036            DATA   6,'6'
005C 0007 0037            DATA   7,'7'
0060 0000                 DATA   0
0062 0005 0031   TABLE2   DATA   5,'1'              ; 25 MILI VOLTS
0066 000A 0032            DATA   10,'2'             ; 50 MILI VOLTS
006A 000F 0033            DATA   15,'3'             ; 75 MILI VOLTS
006E 0000                 DATA   0
0070 172D 0019   TABLE3   DATA   5933,25,'1'        ; 25 HZ
0074 0031
0076 0B75 0032            DATA   2933,50,'2'        ; 50 HZ
007A 0032
007C 078D 004B            DATA   1933,75,'3'        ; 75 HZ
0080 0033
0082 0599 0064            DATA   1433,100,'4'       ; 100 HZ
0086 0034
0088 0215 00FA            DATA   533,250,'5'        ; 250 HZ
008C 0035
008E 00E9 01F4            DATA   233,500,'6'        ; 500 HZ
0092 0036
0094 0000                 DATA   0
                 *
                 * SYSTEM MESSAGES
                 *
0096 0A          MSG60    BYTE   >0A
0097 5354 414E            TEXT   'STANDARD SYSTEM DEFAULTS ARE: '
009B 4441 5244
009F 2053 5953
00A3 5445 4D2C
00A7 4445 4641
00AB 554C 5453
00AF 2041 5245
00B3 3A20
00B5 0D                   BYTE   >0D
00B6 2020 312E   MSG62    TEXT   '  1.  TRIGGER CHANNEL = CHANNEL 1'
```

PAGE-3 GAIT DATA ACQUISITION - PART 2 (VER 27/04/82)

```
 00BA 2020 5452
 00BE 4947 4745
 00C2 5220 4348
 00C6 414E 4E45
 00CA 4C20 3D20
 00CE 4348 414E
 00D2 4E45 4C20
 00D6 31
 00D7 0D                     BYTE   >0D
 00D8 2020 322E  MSG63       TEXT   '  2.  SIGNAL THRESHOLD = 100 MILI VOLTS'
 00DC 2020 5349
 00E0 474E 414C
 00E4 2054 4852
 00E8 4553 484F
 00EC 4C44 203D
 00F0 2031 3030
 00F4 204D 494C
 00F8 4920 564F
 00FC 4C54 53
 00FF 0D                     BYTE   >0D
 0100 2020 332E  MSG64       TEXT   '  3.  SAMPLING RATE = 125 HZ'
 0104 2020 5341
 0108 4D50 4C49
 010C 4E47 2052
 0110 4154 4520
 0114 3D20 3132
 0118 3520 485A
 011C 0A0D                   DATA   >0A0D
 011E 444F 2059  MSG65       TEXT   'DO YOU WANT TO CHANGE ANY OF THE DEFAULTS? (Y/
N) '
 0122 4F55 2057
 0126 414E 5420
 012A 544F 2043
 012E 4841 4E47
 0132 4520 414E
 0136 5920 4F46
 013A 2054 4845
 013E 2044 4546
 0142 4155 4C54
 0146 533F 2028
 014A 592F 4E29
 014E 20
 014F 0D                     BYTE   >0D
 0150 5452 4947  MSG73       TEXT   'TRIGGER CHANNEL = '
 0154 4745 5220
 0158 4348 414E
 015C 4E45 4C20
 0160 3D20
 0162 0D                     BYTE   >0D
 0163 0A         MSG82       BYTE   >0A
 0164 2A2A 2A20              TEXT   '*** READY TO ACQUIRE DATA'
 0168 5245 4144
 016C 5920 544F
 0170 2041 4351
 0174 5549 5245
 0178 2044 4154
 017C 41
 017E 0A0D                   DATA   >0A0D
 0180 5354 4550  MSG101      TEXT   'STEP 1 - '
```

PAGE-4 GAIT DATA ACQUISITION - PART 2 (VER 27/04/82)

```
0184 2031 202D
0188 20
0189 0D                  BYTE  >0D
018A 5354 4550  MSG102  TEXT  'STEP 2 - '
018E 2032 202D
0192 20
0193 0D                  BYTE  >0D
0194 5354 4550  MSG103  TEXT  'STEP 3 - '
0198 2033 202D
019C 20
019D 0D                  BYTE  >0D
019E 5354 4550  MSG104  TEXT  'STEP 4 - '
01A2 2034 202D
01A6 20
01A7 0D                  BYTE  >0D
01A8 5354 4550  MSG105  TEXT  'STEP 5 - '
01AC 2035 202D
01B0 20
01B1 0D                  BYTE  >0D
01B2 5354 4550  MSG106  TEXT  'STEP 6 - '
01B6 2036 202D
01BA 20
01BB 0D                  BYTE  >0D
01BC 0A         MSG110  BYTE  >0A
01BD 2A2A 204D          TEXT  '** MEMORY OVERFLOWED - SAMPLING RATE TOO FAST'

01C1 454D 4F52
01C5 5920 4F56
01C9 4552 464C
01CD 4F57 4544
01D1 202D 2053
01D5 414D 504C
01D9 494E 4720
01DD 5241 5445
01E1 2054 4F4F
01E5 2046 4153
01E9 54
01EA 0D                  BYTE  >0D
01EB 4649 4E49  MSG111  TEXT  'FINISHED'
01EF 5348 4544
01F3 0D                  BYTE  >0D
01F4 0A         MSG112  BYTE  >0A
01F5 2A2A 2A20          TEXT  '*** STORING DATA TO DISK'
01F9 5354 4F52
01FD 494E 4720
0201 4441 5441
0205 2054 4F20
0209 4449 534B
020D.0D                  BYTE  >0D
020E 0A         MSG120  BYTE  >0A
020F 2A2A 2A20          TEXT  '*** INVALID INPUT - TRY AGAIN: '
0213 494E 5641
0217 4C49 4420
021B 494E 5055
021F 5420 2D20
0223 5452 5920
0227 4147 4149
022B 4E3A 20
022E 0D                  BYTE  >0D
```

PAGE-5 GAIT DATA ACQUISITION - PART 2 (VER 27/04/82)

```
022F 0A            MSG200  BYTE  >0A
0230 4348 414E             TEXT  'CHANGE TRIGGER CHANNEL? (Y/N) '
0234 4745 2054
0238 5249 4747
023C 4552 2043
0240 4841 4E4E
0244 454C 3F20
0248 2859 2F4E
024C 2920
024E 0D                    BYTE  >0D
024F 0A            MSG201  BYTE  >0A
0250 4348 414E             TEXT  'CHANGE THRESHOLD LEVEL? (Y/N) '
0254 4745 2054
0258 4852 4553
025C 484F 4C44
0260 204C 4556
0264 454C 3F20
0268 2859 2F4E
026C 2920
026E 0D                    BYTE  >0D
026F 0A            MSG202  BYTE  >0A
0270 4348 414E             TEXT  'CHANGE SAMPLING RATE? (Y/N) '
0274 4745 2053
0278 414D 504C
027C 494E 4720
0280 5241 5445
0284 3F20 2859
0288 2F4E 2920
028C 0D                    BYTE  >0D
028D 0A            MSG203  BYTE  >0A
028E 2020 312E             TEXT  '  1.   25 HZ;  2.   50 HZ;  3.   75 HZ;'
0292 2020 2032
0296 3520 485A
029A 3B20 2032
029E 2E20 2020
02A2 3530 2048
02A6 5A3B 2020
02AA 332E 2020
02AE 2037 3520
02B2 485A 3B
02B5 0D                    BYTE  >0D
02B6 2020 342E     MSG204  TEXT  '  4.  100 HZ;  5.  250 HZ;  6.  500 HZ.'
02BA 2020 3130
02BE 3020 485A
02C2 3B20 2035
02C6 2E20 2032
02CA 3530 2048
02CE 5A3B 2020
02D2 362E 2020
02D6 3530 3020
02DA 485A 2E
02DE 0A0D                  DATA  >0A0D
02E0 5345 4C45     MSG205  TEXT  'SELECT THE NUMBER ASSOCIATED WITH YOUR CHOICE:
'
02E4 4354 2054
02E8 4845 204E
02EC 554D 4245
02F0 5220 4153
02F4 534F 4349
```

PAGE-6 GAIT DATA ACQUISITION - PART 2 (VER 27/04/82)

```
 02F8 4154 4544
 02FC 2057 4954
 0300 4820 594F
 0304 5552 2043
 0308 484F 4943
 030C 453A 20
 030F 0D                     BYTE   >0D
 0310 0A            MSG206   BYTE   >0A
 0311 5341 4D50              TEXT   'SAMPLING RATES AVAILABLE:'
 0315 4C49 4E47
 0319 2052 4154
 031D 4553 2041
 0321 5641 494C
 0325 4142 4C45
 0329 3A
 032A 0D                     BYTE   >0D
 032B 0A            MSG207   BYTE   >0A
 032C 5448 5245              TEXT   'THRESHOLD LEVELS AVAILABLE:'
 0330 5348 4F4C
 0334 4420 4C45
 0338 5645 4C53
 033C 2041 5641
 0340 494C 4142
 0344 4C45 3A
 0347 0D                     BYTE   >0D
 0348 0A            MSG208   BYTE   >0A
 0349 2020 312E              TEXT   '  1.  25 MILI VOLTS,'
 034D 2020 3235
 0351 204D 494C
 0355 4920 564F
 0359 4C54 532C
 035D 0D                     BYTE   >0D
 035E 2020 322E     MSG209   TEXT   '  2.  50 MILI VOLTS, OR'
 0362 2020 3530
 0366 204D 494C
 036A 4920 564F
 036E 4C54 532C
 0372 204F 52
 0375 0D                     BYTE   >0D
 0376 2020 332E     MSG210   TEXT   '  3.  75 MILI VOLTS.'
 037A 2020 3735
 037E 204D 494C
 0382 4920 564F
 0386 4C54 532E
 038A 0A0D                   DATA   >0A0D
                    *
                    * ROUTINE: MOD6
                    * THIS ROUTINE TESTS A NUMBER AND DECIDES WHETHER
                    * IT IS NECESSARY TO SUBTRACT 6 FROM THE NUMBER.
                    * THE ROUTINE KEEPS ON SUBTRACTING 6 FROM THE NUMBER
                    * UNTIL THE RESULT IS LESS THAN OR EQUAL TO 6.
                    *
 038C 0200 0006     MOD6     LI     R0,6              ; MAXIMUM 6 RUNS PER DISK
 0390 8006                   C      R6,R0             ; OVER 6 ?
 0392 1202                   JLE    MODFIN            ; NO, RETURN
 0394 6180                   S      R0,R6             ; YES, R6=R6-6
 0396 10FA                   JMP    MOD6              ; CONTINUE
 0398 045B          MODFIN   B      *R11              ; EXIT
```

PAGE-7 GAIT DATA ACQUISITION - PART 2 (VER 27/04/82)

```
                   *
                   * ROUTINE: LIST
                   * THIS ROUTINE LISTS A PATIENT'S MASTER FILE.
                   *
039A C20B          LIST    MOV   R11,R8           ; SAVE RETURN ADDRESS
039C 0200 0000             LI    R0,MSG11         ; R0 POINTS TO MSG11
03A0 06A0 0000             BL    @MSGOUT          ; OUTPUT MSG11
03A4 06A0 0000             BL    @OPEN            ; OPEN FILE 1
03A8 06A0 0000             BL    @DTH             ; POSN FILE 1 TO SECTOR 0
03AC 0000                  DATA  DTH01
03AE 0204 0000             LI    R4,LABEL1        ; R4 POINTS TO MSG FILE
03B2 C034          LSTLP1  MOV   *R4+,R0          ; OUTPUT ONE LABEL
03B4 130A                  JEQ   LSTFN1           ; GO TO LSTFN1 IF FINISHED
03B6 06A0 03A2             BL    @MSGOUT
03BA 06A0 03AA             BL    @DTH             ; READ ONE RECORD
03BE 0000                  DATA  DTH02
03C0 0200 0000             LI    R0,BUFFER        ; GET MESSAGE PTR
03C4 06A0 0000             BL    @MSGLP2          ; LIST ONE RECORD
03C8 10F4                  JMP   LSTLP1           ; CONTINUE
03CA 06A0 03BC     LSTFN1  BL    @DTH             ; READ PREV RUN NO.
03CE 03BE                  DATA  DTH02
03D0 C1A0 03C2             MOV   @BUFFER,R6       ; KEEP PREV RUN NO.
03D4 06A0 038C             BL    @MODE6           ; GET LOOP INDEX
03D8 C806 0000             MOV   R6,@RNUM         ; DITTO
03DC 0200 0000             LI    R0,MSG30         ; GET MSG30
03E0 06A0 03B8             BL    @MSGOUT          ; OUTPUT MSG30
03E4 0200 0000             LI    R0,MSG31         ; GET MSG31
03E8 06A0 03E2             BL    @MSGOUT          ; OUTPUT MSG31
03EC 06A0 03CC     LSTLP2  BL    @DTH             ; READ RUN NO. AND DATE
03F0 03CE                  DATA  DTH02
03F2 0200 03D2             LI    R0,BUFFER        ; R0 POINTS TO BUFFER
03F6 06A0 03EA             BL    @MSGOUT          ; LIST RUN NO. AND DATE
03FA 0606                  DEC   R6               ; FINISHED?
03FC 15F7                  JGT   LSTLP2           ; NO, CONTINUE
03FE 06A0 03EE             BL    @DTH             ; POSN FILE 1 TO SECT 20
0402 0000                  DATA  DTH07
0404 C1A0 03DA             MOV   @RNUM,R6         ; GET PREV. RUN NO.
0408 06A0 0400     LSTLP3  BL    @DTH             ; GET COMMENT TITLE
040C 03F0                  DATA  DTH02
040E 0200 03F4             LI    R0,BUFFER
0412 06A0 03F8             BL    @MSGOUT          ; OUTPUT COMMENT TITLE
0416 0200 0000             LI    R0,MSG50         ; R0 POINTS TO MSG50
041A 06A0 0414             BL    @MSGOUT          ; OUTPUT MSG50
041E 020A 0000             LI    R10,LABEL4       ; R10 POINTS LINE NO.
0422 C03A          LSTLP4  MOV   *R10+,R0         ; UPDATE POINTER
0424 130A                  JEQ   LSTFN2           ; IF END OF LST, TO LSTFN2
0426 06A0 041C             BL    @MSGOUT          ; LIST LINE NO.
042A 06A0 040A             BL    @DTH             ; GET COMMENT LINE
042E 040C                  DATA  DTH02
0430 0200 0410             LI    R0,BUFFER        ; POINT TO COMMENT LINE
0434 06A0 03C6             BL    @MSGLP2          ; OUTPUT COMMENT LINE
0438 10F4                  JMP   LSTLP4           ; CONTINUE
043A 0606          LSTFN2  DEC   R6               ; FINISED?
043C 15E5                  JGT   LSTLP3           ; NO, CONTINUE
043E 0458                  B     *R8              ; EXIT
                   *
                   * ROUTINE: UPDATE
                   * THIS ROUTINE UPDATES THE PATIENT'S MASTER FILE:
```

PAGE-8 GAIT DATA ACQUISITION - PART 2 (VER 27/04/82)

```
                        * IT LISTS THE FILE ONE ITEM AT A TIME AND ASKS
                        * THE OPERATOR TO ENTER THE NEW INFORMATION.  THE
                        * INFORMATION WILL REMAIN THE SAME IF ONLY A CARRIAGE
                        * RETURN IS ENTERED.
                        *
0440 C20B               UPDATE  MOV   R11,R8            ; SAVE RETURN ADDRESS
0442 0200 0000                  LI    R0,MSG10          ; R0 POINTS TO MSG10
0446 06A0 0428                  BL    @MSGOUT           ; OUTPUT MSG10
044A 06A0 03A6                  BL    @OPEN             ; OPEN FILE 1
044E 06A0 042C                  BL    @DTH              ; POSN FILE 1 TO START
0452 03AC                       DATA  DTH01
0454 0204 03B0                  LI    R4,LABEL1         ; R4 POINTS TO MSG FILE
0458 C034               UPLP1   MOV   *R4+,R0           ; ADVANCE MSG POINTER
045A 1310                       JEQ   UPFIN1            ; GOTO UPFIN1 IF FINISED
045C 06A0 0448                  BL    @MSGOUT           ; OUTPUT LABEL
0460 06A0 0000                  BL    @INPUT            ; GET USER'S RESPONSE
0464 9820 0432                  CB    @BUFFER,@CR       ; NO CHANGE?
0468 000C
046A 1304                       JEQ   UNEXT1            ; NEXT RECORD
046C 06A0 0450                  BL    @DTH              ; STORE THE INFO TO DISK
0470 0000                       DATA  DTH03
0472 10F2                       JMP   UPLP1             ; CONTINUE
0474 06A0 046E          UNEXT1  BL    @DTH              ; UPDATE SECTOR NO.
0478 042E                       DATA  DTH02
047A 10EE                       JMP   UPLP1             ; CONTINUE
047C 06A0 0476          UPFIN1  BL    @DTH              ; GET PREV RUN NO.
0480 0478                       DATA  DTH02
0482 C1A0 0466                  MOV   @BUFFER,R6        ; STORE PREV RUN NO.
0486 06A0 038C                  BL    @MOD6             ; GET LOOP INDEX
048A 06A0 047E                  BL    @DTH              ; POSN TO SECTOR 20
048E 0402                       DATA  DTH07
0490 06A0 048C          UPLP2   BL    @DTH              ; GET COMMENT LABEL
0494 0480                       DATA  DTH02
0496 0200 000A                  LI    R0,>0A            ; LINE FEED
049A 2C80                       OUT   R0
049C 0200 0484                  LI    R0,BUFFER         ; POINT TO COMMENT LABEL
04A0 06A0 045E                  BL    @MSGOUT           ; OUTPUT LABEL
04A4 0200 0418                  LI    R0,MSG50          ; GET UNDERLINE
04A8 06A0 04A2                  BL    @MSGOUT           ; OUTPUT UNDERLINE
04AC 0204 0420                  LI    R4,LABEL4         ; R4 POINTS TO LINE NO.
04B0 C034               UPLP3   MOV   *R4+,R0           ; GET LINE NO.
04B2 1310                       JEQ   UPFIN2            ; IF END OF LIST, FINISHED
04B4 06A0 04AA                  BL    @MSGOUT           ; OUTPUT LINE NO.
04B8 06A0 0462                  BL    @INPUT            ; GET USER'S RESPONSE
04BC 9820 049E                  CB    @BUFFER,@CR       ; NO CHANGE?
04C0 0468
04C2 1304                       JEQ   UNEXT2            ; YES, GOTO UNEXT2
04C4 06A0 0492                  BL    @DTH              ; STORE COMMENT
04C8 0470                       DATA  DTH03
04CA 10F2                       JMP   UPLP3             ; CONTINUE
04CC 06A0 04C6          UNEXT2  BL    @DTH              ; UPDATE SECTOR NO.
04D0 0494                       DATA  DTH02
04D2 10EE                       JMP   UPLP3             ; CONTINUE
04D4 0606               UPFIN2  DEC   R6                ; UPDATE INDEX
04D6 15DC                       JGT   UPLP2             ; IF MORE, CONTINUE
04D8 0458                       B     *R8
                        *
                        * SUBPROGRAM: SAMPLE
```

PAGE-9 GAIT DATA ACQUISITION - PART 2 (VER 27/04/82)

```
                          *
04DA 0000            SAMPLE  DATA  DATAWP          ; WP FOR SAMPLE
04DC 04DE                    DATA  SAM01           ; STARTING PTR FOR SAMPLE
04DE C020 0010       SAM01   MOV   @START.R0       ; BUFFER STARTING PTR
04E2 04F0            LOOP01  CLR   *R0+            ; CLEAR ONE BUFFER LOCN
04E4 0280 7FFE               CI    R0,>7FFE        ; LAST ONE?
04E8 12FC                    JLE   LOOP01          ; NO, CONTINUE
04EA 0200 0007               LI    R0,7            ; NO. OF CHANNELS = 7
04EE C800 0210               MOV   R0,@NUMCHL
04F2 0200 0014               LI    R0,>0014        ; THRESHOLD = 100 MV
04F6 C800 0206               MOV   R0,@THRESH
04FA 0200 046D               LI    R0,1133         ; DELAY INDEX FOR 125 HZ
04FE C800 0214               MOV   R0,@DELAY       ; SETUP DELAY INDEX
0502 0200 007D               LI    R0,125          ; SAMPLING RATE = 125 HZ
0506 C800 020C               MOV   R0,@SMRATE
050A 0200 0001               LI    R0,1            ; CHANNEL 1
050E C800 0212               MOV   R0,@TRICHL      ; TRIGGER CHANNEL = 1
0512 0204 0012               LI    R4,LABEL5       ; R4 POINTS TO FILE LABEL5
0516 C034            SAMPL1  MOV   *R4+,R0         ; ADVANCE POINTER
0518 1303                    JEQ   SNEXT1          ; IF FINISHED, GOTO SNEXT1
051A 06A0 04B6               BL    @MSGOUT         ; OUTPUT ONE MESSAGE
051E 10FB                    JMP   SAMPL1          ; CONTINUE
0520 06A0 04BA       SNEXT1  BL    @INPUT          ; GET USER'S RESPONSE
0524 9820 04BE               CB    @BUFFER,@YES    ; NO CHANGE?
0528 00C0
052A 166C                    JNE   SNEXT2          ; YES, GOTO SNEXT2
052C 0200 022F       SLOOP1  LI    R0,MSG200
0530 06A0 051C               BL    @MSGOUT         ; OUTPUT MSG200
0534 06A0 0522               BL    @INPUT          ; GET RESPONSE
0538 9820 0526               CB    @BUFFER,@YES    ; NO CHANGE?
053C 00C0
053E 1617                    JNE   NEXT01          ; YES, GOTO NEXT01
0540 0200 0150               LI    R0,MSG73
0544 06A0 0532               BL    @MSGOUT         ; OUTPUT MSG73
0548 06A0 0536       AGN01   BL    @INPUT          ; GET OPERATOR'S ENTRY
054C C020 053A               MOV   @BUFFER,R0      ; GET CHANNEL NO.
0550 0980                    SRL   R0,8            ; RIGHT JUSTIFIED
0552 0201 0044               LI    R1,TABLE1       ; GET TABLE1
0556 C0F1            SLOOP2  MOV   *R1+,R3         ; KEEP ADDRESS
0558 1305                    JEQ   ERROR1          ; IF ERROR GOTO ERROR1
055A 8C40                    C     R0,*R1+         ; MATCH CHANNEL NO.
055C 16FC                    JNE   SLOOP2          ; IF NOT MATCHED, AGAIN
055E C803 0212               MOV   R3,@TRICHL      ; SET TRIGGER CHANNEL
0562 1005                    JMP   NEXT01          ; CONTINUE NEXT STEP
0564 0200 020E       ERROR1  LI    R0,MSG120
0568 06A0 0546               BL    @MSGOUT         ; OUTPUT MSG120
056C 10ED                    JMP   AGN01           ; TRY AGAIN
056E 0200 024F       NEXT01  LI    R0,MSG201
0572 06A0 056A               BL    @MSGOUT         ; OUTPUT MSG201
0576 06A0 054A               BL    @INPUT          ; GET RESPONSE
057A 9820 054E               CB    @BUFFER,@YES    ; NO CHANGE?
057E 000C
0580 161A                    JNE   NEXT04          ; YES, GOTO NEXT04
0582 0204 0038               LI    R4,LABEL9       ; R4 POINTS TO LABEL9
0586 C034            SLOOP3  MOV   *R4+,R0         ; ADVANCE POINTER
0588 1303                    JEQ   NEXT03          ; END OF FILE?
058A 06A0 0574               BL    @MSGOUT         ; OUTPUT ONE MESSAGE
058E 10FB                    JMP   SLOOP3          ; CONTINUE
```

```
0590 06A0 0578  NEXT03  BL   @INPUT             ; GET RESPONSE
0594 C020 057C          MOV  @BUFFER,R0         ; STORE RESPONSE TO R0
0598 0980               SRL  R0,8               ; RIGHT JUSTIFIED
059A 0201 0062          LI   R1,TABLE2          ; GET TABLE2
059E C0F1       SLOOP4  MOV  *R1+,R3            ; KEEP ADDRESS
05A0 1305               JEQ  ERROR2             ; IF ERROR GOTO ERROR2
05A2 8C40               C    R0,*R1+            ; MATCH THRESHOLD LEVEL?
05A4 16FC               JNE  SLOOP4             ; NO, CONTINUE
05A6 C803 0206          MOV  R3,@THRESH         ; SETUP NEW THRESHOLD
05AA 1005               JMP  NEXT04             ; CONTINUE NEXT STEP
05AC 0200 020E  ERROR2  LI   R0,MSG120
05B0 06A0 058C          BL   @MSGOUT            ; OUTPUT MSG120
05B4 10ED               JMP  NEXT03             ; TRY AGAIN
05B6 0200 026F  NEXT04  LI   R0,MSG202
05BA 06A0 05B2          BL   @MSGOUT            ; OUTPUT MSG202
05BE 06A0 0592          BL   @INPUT             ; GET RESPONSE
05C2 9820 0596          CB   @BUFFER,@YES       ; NO CHANGE?
05C6 000C
05C8 161D               JNE  SNEXT2             ; YES, CONT SNEXT2
05CA 0204 002E          LI   R4,LABEL8          ; R4 POINTS TO LABEL8
05CE C034       SLOOP5  MOV  *R4+,R0            ; ADVANCE POINTER
05D0 1303               JEQ  NEXT05             ; END OF FILE?
05D2 06A0 05BC          BL   @MSGOUT            ; OUTPUT ONE MESSAGE
05D6 10FB               JMP  SLOOP5             ; CONTINUE
05D8 06A0 05C0  NEXT05  BL   @INPUT             ; GET RESPONSE
05DC C020 05C4          MOV  @BUFFER,R0         ; STORE RESPONSE TO R0
05E0 0980               SRL  R0,8               ; RIGHT JUSTIFY
05E2 0201 0070          LI   R1,TABLE3          ; R1 POINTS TO TABLE3
05E6 C0F1       SLOOP6  MOV  *R1+,R3            ; GET DELAY INDEX
05E8 1308               JEQ  ERROR3             ; IF ERROR GOTO ERROR3
05EA C131               MOV  *R1+,R4            ; GET DELAY INDEX
05EC 8C40               C    R0,*R1+            ; MATCH SAMPLING RATE
05EE 16FB               JNE  SLOOP6             ; IF NOT MATCHED, AGAIN
05F0 C803 0214          MOV  R3,@DELAY          ; SETUP DELAY INDEX
05F4 C804 020C          MOV  R4,@SMRATE         ; SETUP SAMPLING RATE
05F8 1005               JMP  SNEXT2             ; CONTINUE NEXT STEP
05FA 0200 020E  ERROR3  LI   R0,MSG120
05FE 06A0 05D4          BL   @MSGOUT            ; OUTPUT MSG120
0602 10EA               JMP  NEXT05             ; TRY AGAIN
0604 0200 0163  SNEXT2  LI   R0,MSG82
0608 06A0 0600          BL   @MSGOUT            ; OUTPUT MSG82
                *
                * ACQUIRE DATA
                *
060C C820 0010          MOV  @START,@STEPTR     ; RUN NO. 1 STARTING PTR
0610 020A
0612 C820 000A          MOV  @NONE,@SETUP
0616 E7F4
0618 04C8               CLR  R8
061A C028 0020  LOOP1   MOV  @LABEL7(R8),R0
061E 06A0 060A          BL   @MSGOUT            ; OUTPUT A MESSAGE
0622 C820 020A  LP10    MOV  @STEPTR,@RUNPTR    ; SETUP RUN POINTER
0626 02C8
0628 04E0 020E          CLR  @NUMSAM            ; NO SAMPLES = 0
062C C820 0212          MOV  @TRICHL,@MUXADR    ; GET TRIGGER CHANNEL
0630 E7F6
0632 04E0 0202  LOOP2   CLR  @COUNTL            ; CLEAR THRESHOLD COUNTER
0636 8820 E7FE  LOOP3   C    @ADCDAT,@THRESH    ; HIGHER THAN THRESHOLD?
```

PAGE-11 GAIT DATA ACQUISITION - PART 2 (VER 27/04/82)

```
 063A 0206
 063C 11FA                     JLT    LOOP2               ; NO, TEST AGAIN
 063E 05A0 0202                INC    @COUNTL             ; UPDATE COUNTER
 0642 8820 0202                C      @COUNTL,@LIMIT1     ; HIGHER THAN LIMIT?
 0646 00C4
 0648 1301                     JEQ    NEXT1               ; YES, TEST IS FINISHED
 064A 10F5                     JMP    LOOP3               ; NO, CONTINUE
 064C 04E0 0202      NEXT1     CLR    @COUNTL             ; RESET THRESHOLD COUNTER
 0650 C020 0208      LOOP4     MOV    @RUNPTR,R0          ; SET RUN POINTER
 0654 0201 0001                LI     R1,1                ; SET CHANNEL POINTER
 0658 C801 E7F8      LOOP5     MOV    R1,@MUXADR          ; SET CHANNEL ADDRESS
 065C C420 E7FE                MOV    @ADCDAT,*R0         ; GET ONE SAMPLE
 0660 0220 0200                AI     R0,>200             ; ADVANCE BUFFER POINTER
 0664 0581                     INC    R1                  ; UPDATE CHANNEL POINTER
 0666 0281 0007                CI     R1,7                ; 7 CHANNEL?
 066A 12F6                     JLE    LOOP5               ; IF NOT, CONTINUE
 066C 05A0 020E                INC    @NUMSAM             ; INCREMENT NO OF SAMPLES
 0670 8820 020E                C      @NUMSAM,@WANTED     ; BUFFER FULL?
 0674 00CC
 0676 1517                     JGT    ERROR               ; YES, GOTO ERROR
 0678 C820 0212                MOV    @TRICHL,@MUXADR     ; SETUP TRIGGER CHANNEL
 067C E7F8
 067E 8820 E7FE                C      @ADCDAT,@THRESH     ; LOWER THAN THRESHOLD
 0682 0206
 0684 1507                     JGT    NEXT2               ; NO, CONTINUE
 0686 05A0 0202                INC    @COUNTL             ; YES, UPDATE COUNTER
 068A 8820 0202                C      @COUNTL,@LIMIT2     ; GREATER THAN LIMIT?
 068E 0006
 0690 1310                     JEQ    NEXT4               ; YES, RUN FINISHED
 0692 1002                     JMP    NEXT3               ; NO, CONTINUE
 0694 04E0 0202      NEXT2     CLR    @COUNTL             ; RESET COUNTER
 0698 C020 0214      NEXT3     MOV    @DELAY,R0           ; ADJUST TO 500 HZ
 069C 0600           LOOP6     DEC    R0                  ; DECREMENT COUNTER
 069E 16FE                     JNE    LOOP6               ; CONT UNTIL COUNTER EXHAUSTED

 06A0 05E0 0208                INCT   @RUNPTR             ; UPDATE RUN POINTER
 06A4 10D5                     JMP    LOOP4               ; CONTINUE
 06A6 0200 01BC      ERROR     LI     R0,MSG110           ; OVERFLOW MESSAGE
 06AA 06A0 0620                BL     @MSGOUT
 06AE 0460 04DE                B      @SAM01              ; START AGAIN
 06B2 8820 020E      NEXT4     C      @NUMSAM,@WIDTH      ; VALID SAMPLE WIDTH?
 06B6 0008
 06B8 1501                     JGT    LP20                ; YES, CONTINUE
 06BA 10B3                     JMP    LP10                ; ELSE, REDO STEP
 06BC 0200 01EB      LP20      LI     R0,MSG111           ; FINISHED MESSAGE
 06C0 06A0 0436                BL     @MSGLP2
 06C4 C020 020A                MOV    @STEPTR,R0          ; GET STEP POINTER
 06C8 0220 0E00                AI     R0,>0E00            ; SET POINTER FOR NUMSAM
 06CC CC20 020E                MOV    @NUMSAM,*R0+        ; STORE NUMSAM
 06D0 C420 020C                MOV    @SMRATE,*R0
 06D4 A820 000E                A      @OFFSET,@STEPTR     ; UPDATE STEP OFFSET
 06D8 020A
 06DA 05C8                     INCT   R8                  ; UPDATE RUN NO.
 06DC 0288 000C                CI     R8,12               ; LAST ONE?
 06E0 119C                     JLT    LOOP1               ; IF NOT, CONTINUE
 06E2 0200 01F4                LI     R0,MSG112
 06E6 06A0 06AC                BL     @MSGOUT             ; OUTPUT MSG112
S06EA 0380                     RTWP                       ; RETURN TO INVOKING PROG
 06EC 00                       END    PART2               ; END OF PART2
```

PAGE-12 GAIT DATA ACQUISITION - PART 2 (VER 27/04/82)

E7FE	LA	ADCDAT	0548	LP	AGN01	05DE	RP	BUFFER	*E7FA	LA	CONVER
*0204	LA	COUNTC	0202	LA	COUNTL	04C0	RP	CR	04DA	RP	DATAWP
0214	LA	DELAY	04CE	RP	DTH	0452	RP	DTH01	04D0	RP	DTH02
04C8	RP	DTH03	048E	RP	DTH07	06A6	LP	ERROR	0564	LP	ERROR1
05AC	LP	ERROR2	05FA	LP	ERROR3	*0002	LP	HALF	05DA	RP	INPUT
0456	RP	LABEL1	04AE	RP	LABEL4	0012	LP	LABEL5	0020	LP	LABEL7
002E	LP	LABEL8	0038	LP	LABEL9	0004	LP	LIMIT1	0006	LP	LIMIT2
*039A	DP	LIST	04E2	LP	LOOP01	061A	LP	LOOP1	0632	LP	LOOP2
0636	LP	LOOP3	0650	LP	LOOP4	0658	LP	LOOP5	069C	LP	LOOP6
0622	LP	LP10	06BC	LP	LP20	03CA	LP	LSTFN1	043A	LP	LSTFN2
03B2	LP	LSTLP1	03EC	LP	LSTLP2	0408	LP	LSTLP3	0422	LP	LSTLP4
038C	LP	MOD6	0398	LP	MODFIN	0444	RP	MSG10	0180	LP	MSG101
018A	LP	MSG102	0194	LP	MSG103	019E	LP	MSG104	01A8	LP	MSG105
01B2	LP	MSG106	039E	RP	MSG11	01BC	LP	MSG110	01EB	LP	MSG111
01F4	LP	MSG112	020E	LP	MSG120	022F	LP	MSG200	024F	LP	MSG201
026F	LP	MSG202	028D	LP	MSG203	02B6	LP	MSG204	02E0	LP	MSG205
0310	LP	MSG206	032B	LP	MSG207	0348	LP	MSG208	035E	LP	MSG209
0376	LP	MSG210	03DE	RP	MSG30	03E6	RP	MSG31	0012	RP	MSG40
04A6	RP	MSG50	0096	LP	MSG60	00B6	LP	MSG62	00D8	LP	MSG63
0100	LP	MSG64	011E	LP	MSG65	0150	LP	MSG73	0163	LP	MSG82
06C2	RP	MSGLP2	06E8	RP	MSGOUT	E7F8	LA	MUXADR	056E	LP	NEXT01
0590	LP	NEXT03	05B6	LP	NEXT04	05D8	LP	NEXT05	064C	LP	NEXT1
0694	LP	NEXT2	0698	LP	NEXT3	06B2	LP	NEXT4	000A	LP	NONE
0210	LA	NUMCHL	020E	LA	NUMSAM	000E	LP	OFFSET	044C	RP	OPEN
*3EEE	LA	PART2	0000	LA	R0	0001	LA	R1	000A	LA	R10
000B	LA	R11	*000C	LA	R12	*000D	LA	R13	*000E	LA	R14
*000F	LA	R15	*0002	LA	R2	0003	LA	R3	0004	LA	R4
*0005	LA	R5	0006	LA	R6	*0007	LA	R7	0008	LA	R8
*0009	LA	R9	0406	RP	RNUM	0208	LA	RUNPTR	04DE	LP	SAM01
0516	LP	SAMPL1	*04DA	DP	SAMPLE	E7F4	LA	SETUP	*052C	LP	SLOOP1
0556	LP	SLOOP2	0586	LP	SLOOP3	059E	LP	SLOOP4	05CE	LP	SLOOP5
05E6	LP	SLOOP6	020C	LA	SMRATE	0520	LP	SNEXT1	0604	LP	SNEXT2
0010	LP	START	*E7FC	LA	STATUS	020A	LA	STEPTR	0044	LP	TABLE1
0062	LP	TABLE2	0070	LP	TABLE3	0206	LA	THRESH	0212	LA	TRICHL
0474	LP	UNEXT1	04CC	LP	UNEXT2	*0440	DP	UPDATE	047C	LP	UPFIN1
04D4	LP	UPFIN2	0458	LP	UPLP1	0490	LP	UPLP2	04B0	LP	UPLP3
000C	LP	WANTED	0008	LP	WIDTH	0000	LP	YES			

PAGE-01 DP44 (VER 04/06/82)

```
0100 REM DISC PLOT PROGRAM          (VER 08/04/82)
0110 REM PROGRAM BY WAYNE BRODLAND
0120 REM INPUTS FILE NAME, UP TO 7 CHANNEL NUMBERS & THEIR GAINS,
0130 REM OUTPUTS UP TO SIX STEPS OF THE FILE
0140 REM ONE GRAPH PER CHANNEL, ON ONE SHEET OF PAPER.
0150 DIM N$(20),C(10),NR(10)
0160 DIM PX(100),XP%(300),YP%(300)
0170 DIM I%(255),G(10),S(10),DD%(7,255)
0180 DIM A%(7,3)
0190 PI= 3.14159
0200 FOR I=0 TO 93
0210 READ PX(I)
0220 NEXT I
0230 PRINT
0240 PRINT
0250 PRINT
0260 PRINT
0270 PRINT
0280 REM INPUT PLOT PARAMETERS FROM KEYBOARD
0290 INPUT"FILE NAME",N$
0300 CLOSE 1
0310 OPEN 1,N$,0,512,48
0320 INPUT"PLOT EVERY N",SK
0330 FOR I=1 TO 7
0340 C(I)=I
0350 S(I)=I
0360 NEXT I
0370 NG=7
0380 NC=7
0390 NS=6
0400 INPUT"DEFAULTS (0/1)",ID
0410 IF ID<10 THEN GOTO 440
0420 ID=ID-10
0430 INPUT"NS, 6 STEPS",NS,S(1),S(2),S(3),S(4),S(5),S(6)
0440 IF ID=1 THEN GOTO 540
0450 IF ID=2 THEN GOTO 680
0460 INPUT"6 STEPS    ",S(1),S(2),S(3),S(4),S(5),S(6)
0470 INPUT"7 CHANNELS",C(1),C(2),C(3),C(4),C(5),C(6),C(7)
0480 INPUT"7 GAINS    ",G(1),G(2),G(3),G(4),G(5),C(6),G(7)
0490 FOR I=1 TO 7
0500 IF S(I)=0 THEN NS=NS-1
0510 IF C(I)=0 THEN NC=NC-1
0520 IF G(I)=0 THEN NG=NG-1
0530 NEXT I
0540 FOR I=1 TO 7
0550 A%(I,1)=C(I)
0560 A%(I,2)=0
0570 A%(I,3)=0
0580 NEXT I
0590 IF ID=0 THEN GOTO 800
0600 G(1)=1006
0610 G(2)=1006
0620 G(3)=1006
0630 G(4)=6
0640 G(5)=6
0650 G(6)=3
0660 G(7)=50
0670 GOTO 800
```

PAGE-02 DP44 (VER 04/06/82)

```
0680 NG=3
0690 NC=3
0700 NS=6
0710 A%(1,1)=1
0720 A%(1,2)=2
0730 A%(1,3)=3
0740 A%(2,1)=4
0750 A%(2,2)=5
0760 A%(3,1)=6
0770 G(1)=1004
0780 G(2)=4
0790 G(3)=4
0800 NT=0
0810 PRINT
0820 REM PRINT SUMMARY OF RUN INFORMATION
0830 PRINT"RUN          NO OF POINTS"
0840 PRINT
0850 FOR RA=1 TO 6
0860 MP=(RA-1)*8+7
0870 RDFILE 1,MP,I%
0880 NR(RA)=I%(0)
0890 NT=NT+NR(RA)
0900 PRINT RA,NR(RA)
0910 IF NR(RA)>255 THEN NR(RA)=255
0920 NEXT RA
0930 PRINT
0940 PRINT"TOTAL",NT
0950 PRINT
0960 PRINT
0970 PRINT
0980 M1=400
0990 M2=600
1000 M3=1500
1010 M4=1000
1020 NO=(NT+60)/SK+1
1030 DX=(10000-M1-M2)* 0.45/NG
1040 FOR CA=1 TO NG
1050 REM CALCULATE & PLOT AXES
1060 I=59
1070 GOSUB 1860
1080 OF=M1+(10000-M1-M2)/NG*(CA- 0.5)
1090 OV=OF
1100 IF ABS(G(CA))>998 THEN OF=OF+DX
1110 XP%(0)=-(OV-DX)
1120 YP%(0)=M3
1130 XP%(1)=OV+DX
1140 YP%(1)=M3
1150 XP%(2)=-OF
1160 YP%(2)=10000-M4
1170 XP%(3)=OF
1180 YP%(3)=M3
1190 XP%(4)=0
1200 YP%(4)=-1
1210 GOSUB 1670
1220 NEXT CA
1230 I=21
1240 GOSUB 1860
1250 REM READ AND PLOT DATA
```

```
1260 FOR I=I TO 255
1270 DD%(C,I)=0
1280 NEXT I
1290 IA=0
1300 YG= 5.5
1310 FOR KR=1 TO NS
1320 RA=S(KR)
1330 REM READ DATA
1340 FOR CA=1 TO 7
1350 MP=(RA-1)*8+CA-1
1360 RDFILE 1,MP,I%
1370 FOR I=0 TO 255
1380 DD%(CA,I)=I%(I)
1390 NEXT I
1400 NEXT CA
1410 REM PLOT DATA
1420 FOR CA=1 TO NG
1430 OF=M1+(10000-M1-M2)/NG*(CA- 0.5)
1440 IF ABS(G(CA))>998 THEN OF=OF+DX
1450 XG=G(CA)/NG
1460 IF ABS(G(CA))>998 THEN XG=(G(CA)-SGN(G(CA))*1000)/NG
1470 K=0
1480 FOR I=0 TO NR(RA)/SK-1
1490 XX=0
1500 FOR J=1 TO 3
1510 XX=XX+SGN(A%(CA,J))*DD%(ABS(A%(CA,J)),I*SK)
1520 NEXT J
1530 XP%(I)=OF-XX*XG
1540 YP%(I)=M3+(I+IA)*YG*SK
1550 K=K+1
1560 NEXT I
1570 XP%(0)=-XP%(0)
1580 XP%(K)=0
1590 YP%(K)=-1
1600 GOSUB 1670
1610 NEXT CA
1620 IA=IA+(NR(RA)+40)/SK
1630 NEXT KR
1640 I=21
1650 GOSUB 1860
1660 GOTO 290
1670 REM ARRAY PLOT ROUTINE
1680 T=0
1690 X=ABS(XP%(T))
1700 Y=YP%(T)
1710 IF YP%(T)<0 THEN GOTO 1850
1720 XA=INT(X/100)
1730 XB=INT(X-100*INT(X/100))
1740 YA=INT(Y/100)
1750 YB=INT(Y-100*INT(Y/100))
1760 PX(31)=&4000+XA+6*INT(XA/10)
1770 PX(32)=XB+6*INT(XB/10)
1780 PX(33)=YA+6*INT(YA/10)
1790 PX(34)=YB+6*INT(YB/10)
1800 I=31
1810 IF XP%(T)<0 THEN I=30
1820 GOSUB 1860
1830 T=T+1
```

PAGE-04 DP44 (VER 04/06/82)

```
1840 GOTO 1690
1850 RETURN
1860 REM PLOT OUTPUT ROUTINE
1870 IF PX(I)=&FFFF THEN GOTO 1920
1880 IF LOC(&E7EE)<&FF80 THEN GOTO 1880
1890 LOC(&E7EE)=PX(I)
1900 I=I+1
1910 GOTO 1870
1920 RETURN
1930 REM AXES
1940 DATA &6000,&4010,&0000,&0000,&0000
1950 DATA &7000,&4010,&0000,&0099,&0099
1960 DATA &6000,&4099,&0099,&0050,&0000
1970 DATA &7000,&4010,&0000,&0050,&0000
1980 DATA &FFFF
1990 REM RESET
2000 DATA &6000
2010 DATA &4099,&0099,&0099,&0099
2020 DATA &FFFF
2030 REM CONTINUOUS LINE
2040 DATA &0000,&0000,&0000,&6000
2050 DATA &0000,&0000,&0000,&0000
2060 DATA &7000
2070 DATA &FFFF
2080 REM DOT AND SQUARE
2090 DATA &6000
2100 DATA &4000,&0030,&0000,&0630
2110 DATA &7000
2120 DATA &4000,&0060,&0000,&0400
2130 DATA &4000,&0000,&0000,&0560
2140 DATA &4000,&0060,&0000,&0600
2150 DATA &4000,&0000,&0000,&0460
2160 DATA &6000
2170 DATA &FFFF
2180 REM MULTIPLE AXES
2190 DATA &6000,&4010,&0000,&0083,&0000
2200 DATA &6000,&4000,&0000,&0015,&0400
2210 DATA &7000,&4000,&0000,&0030,&0500
2220 DATA &6000,&4089,&0099,&0015,&0400
2230 DATA &7000,&4089,&0099,&0000,&0600
2240 DATA &6000,&FFFF
2250 REM SKIP TO NEXT AXES
2260 DATA &6000,&4000,&0000,&0033,&0500
2270 DATA &FFFF
2280 END
```

```
PAGE-01 EDIT3            (VER 04/06/82)

0100 REM DISC STANDARDIZATION AND EDITING PROGRAM
0110 REM BY WAYNE BRODLAND
0120 REM ALLOWS CHANNEL SWAPPING,DC OFFSET,
0130 REM DATA RECONSTRUCTION & APPENDING STEP INFORMATION
0140 DIM TE(8),C(8),SI(6,5)
0150 DIM TV(10)
0160 DIM N$(20),DD%(8,255),I%(255)
0170 DIM M$(20)
0180 DIM G(10)
0190 NS=6
0200 CLOSE 1
0210 INPUT"READ FILE NAME ",N$
0220 OPEN 1,N$,0,512,48
0230 CLOSE 2
0240 INPUT"WRITE FILE NAME",M$
0250 OPEN 2,M$,0,512,48
0260 INPUT"NEW CHANNEL ORDER (7)",C(1),C(2),C(3),C(4),C(5),C(6),C(7)
0270 INPUT"CHANNEL MULT FACTORS",G(1),G(2),G(3),G(4),G(5),G(6),G(7)
0280 PRINT"INPUT STEP INFORMATION FOR ";NS;" STEPS"
0290 FOR RA=1 TO NS
0300 INPUT SI(RA,1),SI(RA,2),SI(RA,3),SI(RA,4),SI(RA,5)
0310 NEXT RA
0320 REM LOOP TO DO EACH STEP
0330 FOR RA=1 TO NS
0340 REM INPUT STEP DATA
0350 FOR CA=1 TO 8
0360 MP=(RA-1)*8+CA-1
0370 RDFILE 1,MP,I%
0380 FOR I=0 TO 255
0390 DD%(CA,I)=I%(I)
0400 NEXT I
0410 NEXT CA
0420 REM SWAP CHANNELS
0430 FOR I=0 TO 255
0440 FOR J=1 TO 7
0450 TE(C(J))=DD%(J,I)
0460 NEXT J
0470 FOR J=1 TO 7
0480 DD%(J,I)=TE(J)
0490 NEXT J
0500 NEXT I
0510 REM SET D.C. OFFSET OF FIRST 6 CHANNELS
0520 NR=DD%(8,0)
0530 S=3
0540 FOR J=1 TO 6
0550 DC=0
0560 FOR I=NR-S TO NR-1
0570 DC=DC+DD%(J,I)
0580 NEXT I
0590 DC=DC/S+ 0.5
0600 IF J=4 THEN DC=0
0610 FOR I=0 TO 255
0620 DD%(J,I)=DD%(J,I)-DC
0630 NEXT I
0640 DD%(8,J+2)=DC
0650 NEXT J
0660 REM RECONSTRUCT MISSING DATA
0670 P1=DD%(1,0)+DD%(2,0)+DD%(3,0)
```

```
PAGE-02 EDIT3            (VER 04/06/82)

0680 V=20
0690 P2=DD%(1,V)+DD%(2,V)+DD%(3,V)
0700 M=INT(V*P1/(P2-P1)+ 0.5)
0710 PRINT
0720 PRINT
0730 PRINT"STEP ";RA
0740 PRINT"DATA POINTS RECONSTRUCTED: ";M
0750 FOR J=1 TO 7
0760 FOR I=255-M TO 0 STEP-1
0770 DD%(J,I+M)=DD%(J,I)
0780 NEXT I
0790 NEXT J
0800 REM FILL IN FIRST 6 CHANNELS
0810 FOR J=1 TO 6
0820 IF J=4 THEN GOTO 860
0830 FOR I=0 TO M-1
0840 DD%(J,I)=DD%(J,M)*I/M
0850 NEXT I
0860 NEXT J
0870 DD%(8,2)=NR
0880 DD%(8,0)=NR+M
0890 REM ADJUST GAINS
0900 FOR I=1 TO 7
0910 FOR J=0 TO 255
0920 DD%(I,J)=DD%(I,J)*G(I)
0930 NEXT J
0940 NEXT I
0950 REM TRANSFER STEP INFORMATION
0960 P=11
0970 FOR J=1 TO 5
0980 DD%(8,P+J-1)=SI(RA,J)
0990 NEXT J
1000 REM CALCULATE HEEL STRIKES
1010 XL=0
1020 XX=0
1030 TP=1
1040 FOR I=1 TO 5
1050 TV(I)=0
1060 NEXT I
1070 TV(1)=1
1080 SC=0
1090 FF=DD%(8,2)-1
1100 SD=-DD%(7,FF)-DD%(7,FF-1)+DD%(7,FF-2)+DD%(7,FF-3)
1110 FF=FF-1
1120 IF SD<10 THEN GOTO 1100
1130 FOR I=1 TO FF
1140 TT=DD%(1,I)+DD%(2,I)+DD%(3,I)
1150 IF TT<600 THEN GOTO 1250
1160 XX=(4*DD%(2,I)+8*DD%(3,I))/TT
1170 K=I
1180 IF ABS(XX-XL)< 0.05 THEN GOTO 1230
1190 IF SC<5 THEN GOTO 1250
1200 SC=0
1210 TP=TP+1
1220 TV(TP)=I-1
1230 SC=SC+1
1240 GOTO 1260
1250 SC=0
```

PAGE-03 EDIT3 (VER 04/06/82)

```
1260 XL=XX
1270 NEXT I
1280 TP=TP+1
1290 TV(TP)=FF
1300 FOR I=1 TO 5
1310 DD%(8,20+I)=TV(1)
1320 NEXT I
1330 PRINT"FOOT PATTERN",SI(RA,1),SI(RA,2),SI(RA,3),SI(RA,4),SI(RA,5)
1340 PRINT"HEEL STRIKES          ",TV(1),TV(2),TV(3),TV(4)
1350 PRINT"DIFFERENCES                ",TV(2)-TV(1),TV(3)-TV(2),TV(4)-TV(3)
1360 IF RA<>6 THEN GOTO 1390
1370 INPUT"RL RATIO",RL
1380 DD%(8,16)=1000*RL
1390 REM STORE BACK TO DISC
1400 FOR CA=1 TO 8
1410 FOR I=0 TO 255
1420 I%(I)=DD%(CA,I)
1430 NEXT I
1440 MP=(RA-1)*8+CA-1
1450 WRFILE 2,MP,I%
1460 NEXT CA
1470 NEXT RA
1480 END
```

```
0100 REM RECONSTRUCTION PROGRAM   (VER 03/05/82)
0110 REM PROGRAM BY WAYNE BRODLAND
0120 REM CONSTRUCTS MULTIPLE STEPS FROM SINGLE STEPS
0130 REM CALCULATES MOTION OF X CENTER OF FORCE
0140 DIM N$(20),C(10),NR(10)
0150 DIM PX(60),XP%(270),YP%(270)
0160 DIM I%(255),G(10),S(10)
0170 DIM DD%(8,255),P(10)
0180 DIM V%(5,270),H(10)
0190 DIM R$(20),L$(20),C$(20),B(10),O(10)
0200 PI= 3.14159
0210 FOR I=0 TO 60
0220 READ PX(I)
0230 NEXT I
0240 PRINT
0250 PRINT
0260 REM INPUT PLOT PARAMETERS FROM KEYBOARD
0270 SK=1
0280 INPUT"OPTION(2/3)",NN
0290 INPUT"PLOT BASE & RANGE",BP,RP
0300 PRINT"FILE NAME, STEP, BASE & GAIN FOR:"
0310 INPUT"TYPE 1      ",R$,S(1),B(1),H(1)
0320 INPUT"TYPE 2      ",L$,S(2),B(2),H(2)
0330 INPUT"COMBINATION",C$,S(3),B(3),H(3)
0340 IF L$="X"THEN L$=R$
0350 IF C$="X"THEN C$=L$
0360 REM INPUT STEP CONSTRUCTION PARAMETERS
0370 INPUT"SUM OFFSET",O(5)
0380 FOR J=1 TO 3
0390 INPUT"TYPE, OFFSET & X LOCATION",T(J),O(J),P(J)
0400 NEXT J
0410 REM INPUT DATA FROM DISC
0420 FOR J=1 TO 3
0430 IF J=1 THEN N$=R$
0440 IF J=2 THEN N$=L$
0450 IF J=3 THEN N$=C$
0460 CLOSE 1
0470 OPEN 1,N$,0,512,48
0480 FOR CH=1 TO 3
0490 MP=(S(J)-1)*8+CH-1
0500 RDFILE 1,MP,I%
0510 FOR I=0 TO 255
0520 IF CH> 1.5 THEN DD%(3*J-3+CH-1,I)=I%(I)
0530 DD%(3*J-3,I)=DD%(3*J-3,I)+I%(I)
0540 NEXT I
0550 NEXT CH
0560 MP=(S(J)-1)*8+8-1
0570 RDFILE 1,MP,I%
0580 NR(J)=I%(0)
0590 IF NR(J)>255 THEN NR(J)=255
0600 NEXT J
0610 REM TIME & GAIN SCALE, AND TRANSFER TO V%
```

PAGE-02 CON7 (VER 07/06/82)

```
0680 V%(J-1,I)=V%(J-1,I)*ABS(H(J))
0690 NEXT I
0700 NEXT J
0710 G(1)=1004:G(2)=1004:G(3)=1004
0720 NG=3:NC=7:NS=1
0730 REM CALCULATE COMPOSITE STEF
0740 FOR J=1 TO 3
0750 IF T(J)=0 THEN GOTO 800
0760 FOR I=0 TO BP/B(J)*NR(T(J))+2
0770 IF I+O(J)>RP THEN GOTO 790
0780 V%(3,I+O(J))=V%(3,I+O(J))+V%(T(J)-1,I)*SGN(H(T(J)))
0790 NEXT I
0800 NEXT J
0810 REM CALCULATE CENTER OF FORCE
0820 FOR I=0 TO RP
0830 K=I*B(3)/BP
0840 I1=INT(K)
0850 R=K-I1
0860 V%(4,I)=0
0870 X1=DD%(7,I1+1)*4+DD%(8,I1+1)*8
0880 X2=DD%(7,I1)*4+DD%(8,I1)*8
0890 X3=DD%(6,I1+1)*R+DD%(6,I1)*(1-R)
0900 IF X3<500 THEN GOTO 920
0910 V%(4,I)=((1-R)*X2+R*X1)*250/X3
0920 AA=0
0930 V%(5,I)=0
0940 FOR Z=1 TO 3
0950 IF I-O(Z)<0 THEN GOTO 990
0960 IF T(Z)=0 THEN GOTO 990
0970 V%(5,I)=V%(5,I)+V%(T(Z)-1,I-O(Z))*P(Z)
0980 AA=AA+V%(T(Z)-1,I-O(Z))
0990 NEXT Z
1000 IF AA<=500 THEN V%(5,I)=0
1010 IF AA>500 THEN V%(5,I)= 250.*V%(5,I)/AA
1020 NEXT I
1030 REM PLOT AXES
1040 M1=400:M2=600
1050 M3=1500:M4=1000
1060 DX=(10000-M1-M2)* 0.45/NG
1070 FOR CA=1 TO NN
1080 I=59
1090 GOSUB 1870
1100 OF=M1+(10000-M1-M2)/3*(CA- 0.5)
1110 OV=OF
1120 IF ABS(G(CA))>998 THEN OF=OF+DX
1130 XP%(0)=-(OV-DX)
1140 YP%(0)=M3
1150 XP%(1)=OV+DX
1160 YP%(1)=M3
1170 XP%(2)=-OF
1180 YP%(2)=10000-M4
1190 XP%(3)=OF
1200 YP%(3)=M3
1210 XP%(4)=0
1220 YP%(4)=-1
1230 GOSUB 1680
1240 NEXT CA
1250 I=21
```

```
1260 GOSUB 1870
1270 REM PLOT DATA
1280 T(4)=0
1290 O(6)=0:O(7)=0:O(8)=0
1300 YG=10
1310 FOR CA=1 TO 2+2*NN
1320 CB=INT(CA/2+ 0.6)
1330 IF CA>2 THEN CB=CB-1
1340 OF=M1+(10000-M1-M2)/3*(CB- 0.5)
1350 IF ABS(G(CB))>998 THEN OF=OF+DX
1360 XG=G(CB)/NG
1370 IF ABS(G(CB))>998 THEN XG=(G(CB)-SGN(G(CB))*1000)/NG
1380 IF CA>4 THEN GOTO 1400
1390 IF T(CA)=0 THEN GOTO 1640
1400 K=0
1410 IA=O(CA)
1420 Z=0
1430 FOR I=0 TO RP-IA
1440 IF CA>4 THEN GOTO 1500
1450 IF V%(T(CA)-1,I)<>0 THEN GOTO 1480
1460 Z=Z+1
1470 GOTO 1590
1480 XP%(I-Z)=OF-V%(T(CA)-1,I)*XG
1490 GOTO 1540
1500 IF V%(CA-3,I)<>0 THEN GOTO 1530
1510 Z=Z+1
1520 GOTO 1590
1530 XP%(I-Z)=OF-V%(CA-3,I)*XG
1540 IF INT(CA/2)*2<>CA THEN GOTO 1570
1550 IF I-Z=0 THEN GOTO 1570
1560 IF INT((I-Z)/2)*2=I-Z THEN XP%(I-Z)=-XP%(I-Z)
1570 YP%(I-Z)=M3+(I+IA)*YG*SK
1580 K=K+1
1590 NEXT I
1600 XP%(0)=-XP%(0)
1610 XP%(K)=0
1620 YP%(K)=-1
1630 GOSUB 1680
1640 NEXT CA
1650 I=21
1660 GOSUB 1870
1670 STOP
1680 REM ARRAY PLOT ROUTINE
1690 T=0
1700 X=ABS(XP%(T))
1710 Y=YP%(T)
1720 IF YP%(T)<0 THEN GOTO 1860
1730 XA=INT(X/100)
1740 XB=INT(X-100*INT(X/100))
1750 YA=INT(Y/100)
1760 YB=INT(Y-100*INT(Y/100))
1770 PX(31)=&4000+XA+6*INT(XA/10)
1780 PX(32)=XB+6*INT(XB/10)
```

PAGE-04 CON7 (VER 07/06/82)

```
1840 T=T+1
1850 GOTO 1700
1860 RETURN
1870 REM PLOT OUTPUT ROUTINE
1880 IF PX(I)=&FFFF THEN GOTO 1930
1890 IF LOC(&E7EE)<&FF80 THEN GOTO 1890
1900 LOC(&E7EE)=PX(I)
1910 I=I+1
1920 GOTO 1880
1930 RETURN
1940 REM AXES
1950 DATA &6000,&4010,&0000,&0000,&0000
1960 DATA &7000,&4010,&0000,&0099,&0099
1970 DATA &6000,&4099,&0099,&0050,&0000
1980 DATA &7000,&4010,&0000,&0050,&0000
1990 DATA &FFFF
2000 REM RESET
2010 DATA &6000
2020 DATA &4099,&0099,&0099,&0099
2030 DATA &FFFF
2040 REM CONTINUOUS LINE
2050 DATA &0000,&0000,&0000,&6000
2060 DATA &0000,&0000,&0000,&0000
2070 DATA &7000
2080 DATA &FFFF
2090 REM DOT AND SQUARE
2100 DATA &6000
2110 DATA &4000,&0030,&0000,&0630
2120 DATA &7000
2130 DATA &4000,&0060,&0000,&0400
2140 DATA &4000,&0000,&0000,&0560
2150 DATA &4000,&0060,&0000,&0600
2160 DATA &4000,&0000,&0000,&0460
2170 DATA &6000
2180 DATA &FFFF
2190 END
```