# THE EXTRACTION OF SINGLE LIMB FORCE DATA FROM COMPOSITE RECORDS ON A LARGE FORCE BOARD

by

Ross Manning

A thesis
presented to the University of Manitoba
in partial fulfillment of the
requirements for the degree of
Master of Science

Winnipeg, Manitoba

# THE EXTRACTION OF SINGLE LIMB FORCE DATA FROM COMPOSITE RECORDS ON A LARGE FORCE BOARD

BY

ROSS MANNING

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

This research has focused on data analysis techniques for a large triangular force plate which was designed to accommodate multiple-step gait patterns. Based on the vertical force records alone, two new algorithms have been developed to identify the time at which a heel strike occurs. A new technique has also been developed for identifying the occurrence time of the toe-off of the trailing foot.

As an application of the identification of these temporal events, two algorithms were developed to extract, from periods of dual support, the individual force information on each limb for a single, multiple-step traverse of the force board.

## ACKNOWLEDGMENTS

I began my Master's Program with the specific intent of preparing myself for a Biomedical Engineering career. And now, as I look back on the last two years, I realize that I have come a long way towards this goal. I can also see that I have had a lot of help along the way, and for this I have many people to thank.

I wish to thank Dr. A. B. Thornton-Trump, my adviser, for his leadership and for his support at many levels. I wish also to gratefully acknowledge the financial support offered by the Natural Sciences and Engineering Research Council.

Many thanks to Dr. Balakrishnan for his previous work and advice on the force board. Special gratitude is extended to Mr. W. Brodland for sharing with me his wisdom and experience, and for helping me to maintain a focus.

The interdisciplinary nature of the Biomedical Lab has created a pool of electrical and mechanical resource people interacting within a spirit of sharing and of caring. I wish to thank these friends and colleagues for much of the knowledge I have acquired as a result of our discussions.

And most of all I would like to thank my loving wife Wendy who has encouraged and supported our family for the last two years. (Its your turn now Wendy)

## DEDICATIONS

I dedicate this thesis to my beautiful wife Wendy, and to my beautiful son Michael. You are the sunshine of my life.

TABLE OF CONTENTS

## LIST OF FIGURES

LIST OF FIGURES (cont'd)

## LIST OF TABLES



## APPENDICES

## NOMENCLATURE

A,B,C....vertical force readings at corners A,B,C of force board

A/D......analog to digital converter

CPX,CPY..center of vertical pressure in x and y directions

CPX(hs)..center of pressure at time of heel strike

CPX(t) ..center of pressure at time "t"

CPX(to)..center of pressure at time of toe-off

D,E .....locations of center of pressure per Figure 19

FL ......vertical force on left foot

FR ......vertical force on right foot

h........perpendicular height of force board

K........spring constant

L........base length of force board

"LRL"....a "left, right, left" sequence of footfalls

M,N .....limits of CPX location per Figure 16

"PECK" ..a plot of CPX against CPY

Ra1,Ra2..resistance values of active strain gauges

Rd1,Rd2..resistance values of dummy strain gauges

RVCP.....resultant velocity of the center of pressure

SDVSUM...second difference of total vertical force

T .......total vertical force (FL + FR)

T(hs)....time at heel-strike

T(to)....time at toe-off

VA.......velocity angle of the center of pressure

Vab,Vcb..voltage across given pts on strain bridge circuit

# CHAPTER 1

# INTRODUCTION

## 1.1 Gait Studies and Force Boards in Perspective

Human walking has been studied extensively by many investigators from various disciplines including anatomy, kinesiology, neurophysiology and biomechanics. Recorded studies of human gait date back as far as 1682 [6]. The tools available to such early researchers were limited, and many subsequent studies of human locomotion were based on visual observations alone. With the invention of photography in 1839, a new research tool was made available, and, three decades later, improvements in the state of the art made it possible to photograph "fast" motion. Pioneering work done by Muybridge [5] around 1883 marked the beginning of a new era in the study of human gait. By 1872, researchers [1,2] had also begun to develop mechanisms to measure foot contact forces. A variety of ingenious force measuring devices were subsequently developed.

As a result of the efforts of their predecessors, modern researchers of human gait have several accurate and sophisticated data gathering tools at their disposal. Kinematic data, relating the subject's various limb positions with time, has been recorded at high rates using charged couple device equipment (C.C.D.) [52]. The evolution of digital data acquisition, storage and processing equipment, most notably the personal computer, allows for more complex analyses on larger data sets. As discussed in more detail in section 2.2, advancements have been made in the design of force boards, and there has been an increasing interest in their clinical application. Force board systems are in fact now commercially available [57]. Unfortunately, these force boards are relatively small, typically 18" square, and can only accommodate a single foot. This creates a tendency for the subject to "aim" or alter his or her otherwise natural gait in order to hit the force plate.

The University of Manitoba force plate is unique by virtue of its size (see Figure 1). This large, triangular shape offers many advantages. The "aiming" tendency is greatly reduced, and since the board can accommodate up to three footfalls for a normal subject, multiple-step gait data can be collected. A preliminary study on Legg Perthes braces clearly demonstrated the advantage of the large platform as the subjects showed a "broad compass gait" [56].

## 1.2 Problem Statement

As shown in Figure 2, one gait cycle consists of two basic phases for each limb. The stance, or support phase, begins with the heel strike of a given limb, and ends with the toe-off of the same limb. This toe-off marks the start of the swing, or non-support phase, which ends with the following heel strike. For a relatively short period of time, the stance phases for each foot overlap. During this period both limbs are weight-bearing and, while the net force components of the ground reaction are known, the individual limb forces are not.

Accompanying the many advantages and features of a larger force platform, are a set of particular problems and design trade- offs. Although it has been shown that 99.7% of the force signal power is below the 8th harmonic [19], it is obvious that the designer of a force board should aim for as high a natural resonant frequency as possible, this being a trade-off with size. Characteristic of large force boards is an greater mass and lower structural rigidity, both of which tend to lower the natural frequency. Clearly an optimizing technique would be appropriate here to predict the best combinations of size, mass, and rigidity. The first step in such a study would be the examination of the mechanical properties of the present force board.

Several studies have been made concerning the location and movement of the center of vertical pressure on the sole of the foot [10,15,18,26,35,41-43]. These studies are discussed in more detail in section 2.2. Owing either to limitations of the equipment, or the nature of the study, limited research has been done regarding the progression of the center of pressure from one foot to the next in the walking cycle. By virtue of its size, the University of Manitoba force board permits such a study. A graphical, three dimensional representation of the center of pressure against time could well prove to be a useful diagnostic tool in a clinical setting.

In many gait related studies, both force data and cinematographic data is collected, most often in an attempt to make some Newtonian connection between the forces and kinematics of body segment position. Typically the latter information is obtained by analyzing, on a frame by frame basis, the location of targets attached to the anatomical locations defining that segment. In many cases a "hand digitizing" technique is used to input the coordinates of the body markers to the computer. The expense of this equipment, the constraints and problems associated with the synchronizing of visual and force information, and the errors introduced in position identification are a few of the disadvantages of such a system. In his M.Sc. thesis, Brodland [54] demonstrated that at least one temporal landmark of human gait, that being the instance of an on-board heel strike as illustrated in Figure 2, could be identified based on the vertical force record alone. While his algorithm is highly accurate, it is based on the location of the center of pressure in one direction only, this adding the constraint that the subject must walk, more or less, in a straight line. The time of toe-off, a much more subtle temporal landmark from a force perspective , was not identified by Brodland's algorithm.

A clear application of the heel strike and toe-off times is the identification of the periods of dual support. During dual support, the forces generated by a single foot are lost in the composite record. Brodland [54] proposed "reconstruction procedures" for obtaining the force record of any particular foot pattern (for example a left-right foot sequence). The requirement of the Brodland algorithm is that data from two or more traverses be combined, this data having been taken at different times.

## 1.3 Objectives of This Study

The objectives of this study are:

(1) To examine the vibrational properties of the force board with a view towards optimizing the design of the suspension system.

(2) To develop a meaningful representation of the dynamic location of the center of vertical pressure for a multiple-step traverse of the force board.

(3) To examine and develop new algorithms to identify the occurrence times of both heel strikes and toe-offs based on the vertical force record alone, these algorithms being independent of the direction of travel of the subject.

(4) To develop algorithms to extract both left and right individual force records from a single, multiple-step traverse of the force board.

CHAPTER 2

# LITERATURE REVIEW

## 2.1 Introduction

The historical development of the study of human walking since the pre-renaissance era has been thoroughly reviewed by others [37]. Of the numerous gait-related studies that have been done, many have involved platforms or devices to measure ground-reaction forces. Detailed and colorful historical accounts of the technological evolution of these devices have been given by Gola [17], Cunningham [53], Balakrishnan [55], and others.

The review of related literature given in this chapter is not presented in a historical perspective, this having been done by many others, but rather in light of its contribution to the outcome of this research. Specifically, this literature search and review was done with the following purposes in mind:

(1) to gain a basic understanding of the fundamental gait parameters and the methodology used in studying gait.

(2) to study and compare force board data collection systems with regards to their construction and performance properties.

(3) to gain information regarding the appropriate selection of data sampling rate and filter cutoff frequency.

(4) to examine various other methods used in identifying heel strike and toe-off times.

(5) to become aware of other studies relating to center of

pressure movement and to see whether or not previous methods had been developed to extract single limb support information from composite force records ( none were found ).

## 2.2 Force Boards

While many studies have been done using platforms or devices to measure ground reaction forces, most have focused on the processing of the data collected, and detailed information regarding the devices themselves has been difficult to obtain. Cunningham [53] describes two such devices; a pylon device inserted into an artificial limb, and a 15" by 20" force board consisting of an aluminum plate supported by tubular aluminum columns at each of the four corners. With a rosette arrangement of strain gauges on each of the support columns, the analog outputs provided a measure of force in the vertical, medio-lateral, and posterior-anterior directions as well as the torques about the vertical axes. The natural resonant frequency in the vertical direction was reported to be very high (no values given), with a value of 105 hz. in the horizontal direction, and a rotational frequency of 140 hz. about the vertical axes. Damping in the horizontal direction was added by providing an oil film between the bottom of the plate and a second rigid horizontal plate.

Another force board described in some detail was that of Gola's [17]. He used a small triangular platform with two "chain-link" supports at each corner. With a total of six supports, the "solution" of forces was statically determinant, meeting with one of his design constraints. Since the supports were angled relative to the standard vertical/horizontal coordinate system, a transformation matrix was necessary to resolve forces into their vertical and horizontal components. The board was claimed to have a natural resonant frequency of 190 hz. (direction unspecified). In terms of support geometry, the Gola force board is most like the University of Manitoba device which was also designed with the constraint of static determinancy. As is discussed in more detail in sec.5.1.2, this requirement greatly limits the suspension rigidity.

With the exception of the force board designed by Yamashita and Katoh [9], all the reported force platforms were small in comparison to the University of Manitoba force plate and were intended for measuring the ground reaction forces on a single footfall. The force plate developed by Harris [33], for example, consisted of a 72.6 cm. square plate suspended by cantilever transducers at each corner. Commercially made force platforms such as the "Vicon" [57] and the more common "Kistler" force platform have been used by many researchers [15,18,21,36,37]. The Kistler Type 9261 platform provides eight analog outputs; three directions of force, moments about three axes, and the center of pressure in two directions with respect to an inertial reference axes at the board's center. A natural frequency of 200 hz. is reported (direction unspecified).

To overcome the size limitations, some researchers, such as Chao et al [37] have used more than one force plate. In this study they used a custom-made transparent plate with seven quartz transducers and located a second Kistler Type 9281-A11 force platform one stride length from the first. In terms of size alone, the Yamashita/Katoh [9] force board is most similar to the University of Manitoba's device. It is a right isosceles triangle in shape, the hypotenuse being 222 cm. (87") in length compared to 96" on the University of Manitoba board. Yamashita's board was designed and used for two footfall gait patterns. Unfortunately, the device is capable of vertical force measurement only. No information was given regarding its dynamic properties.

## 2.3 Studies Relating to Center of Pressure

Several studies have involved the monitoring of the foot-pressure distribution and/or the location of the center of vertical force using a variety of schemes [9,10,15,18,26,27,32,33, 35,41-43,48,54]. In a study done by Scranton and McMaster [10], a camera was set up under a plexiglass walkway covered with liquid crystal sheets. These yielded isobaric contour representations of the local pressure distribution on the feet of normal and flat-footed subjects. In a study of runners by Cavanagh [18], the position of a footfall was marked by powdered chalk left on an adhesive covered surface. The center of pressure as indicated by the vertical force distribution, was located relative to the foot center line as indicated by the chalk impressions. Hiroshi [35] used "Prescale" a brand-name pressure detecting sheet (visual) in his study of foot-sole pressure.

Many of the studies involving center of pressure were not on gait [41,43,33]. For example, in two of his studies, Murray [41,43] monitored the location of the center of pressure for subjects in the standing, squatting, and sitting positions. Harris [33] studied the swaying position of the center of pressure for normal and cerebral-palsied subjects standing on a platform. Of the remaining studies on gait, most have focused on the center of pressure for a single limb [10,15,18,27,36,37,43, 48]. Using three models of the "inverted pendulum" model, Siegler [32] observed the location of the body's center of gravity in relation to the center of pressure on a single foot.

Of great interest was the manner in which several researchers have analyzed and represented center of pressure related data. Cavanagh [15] discussed several methods of smoothing the center of pressure plots to reduce the effect of artifact-related scatter. In a separate study, Cavanagh and Lafortune [18] used computer graphics to represent, in a three dimensional sense, the location of a single vertical force vector on the sole of the foot. Grundy [42] plotted the vertical force against the location of the center of pressure in the sagittal plane. In a somewhat similar manner, Bocardi [48] produced a vectorial representation of the resultant ground reaction force with the tail of the vector located at the momentary center of pressure in the sagittal plane. Harris [33] displayed the x and y coordinates of center of pressure on a density histogram. Using computer graphics, a number of isometric columns were drawn on a grid with the height of the columns representing the time (number of sample points) and the position on the grid corresponding to the center of pressure location.

Of all the literature reviewed, the study done by Yamashita [9], was most similar to this study, not only in terms of the equipment as mentioned earlier, but also due to the fact that he analyzed the dynamic location of the center of pressure for a two-step gait pattern. The Fourier coefficients were determined for both the posterior-anterior and the medio-lateral components of the center of pressure. In spite of a lower accuracy in locating the center of pressure ( 5%-10% versus 0.5% on the University of Manitoba board), the curves generated in his study compare well with the equivalents from this study. While not commented on by Yamashita, it is also interesting to note that the "corners" associated with the heel strike and toe-off times are evident in his plots as shown in Figure 3.

# CHAPTER 3

## DATA ACQUISITION SYSTEM AND EXPERIMENTAL SET UP

### 3.1 Introduction

The intent of this chapter is to provide a brief description of the equipment and software used in the data collection process, and to describe some of the tests done to determine the mechanical properties of the force board. It should be made clear at this point that most of the hardware, and some of the software used, had been developed prior to the author's involvement in this research. A more detailed description of the equipment and a historical account of its evolution is given in the author's "User's Guide to the Force Plate" [60].

### 3.2 System Overview

The force board system, shown schematically in Figure 4, measures and records the vertical, medio-lateral, and posterior-anterior ground reaction forces, and determines when each foot contacts or is lifted off the force board and approach areas. Eight channels of data are digitized by the A/D board and fed into the computer. With reference to Figure 5, three of these channels are used to measure vertical force (one transducer at each corner), two for medio-lateral force (one each at corners A and C), and one for posterior-anterior force (located at corner A). The remaining two channels were used to determine the times of contact and removal of the feet from the plate (one channel per foot). The force board and approach areas were covered with light sheet metal to which was connected a nine volt battery, making these areas "hot" relative to the system ground. Metal ducting tape was attached to the sole of each of the subject's feet and completed the circuit when in contact with the metal skins. Using this "0/1" approach, the times of heel strikes and toe-offs were measured independently for each foot. The times of these events, measured in this manner, were the reference standards which were used to gauge the accuracy of the corresponding algorithm-identified values.

### 3.3 Force Board Suspension System

A clever and unique suspension system was developed by Dr. Balakrishnan to support the plate and to generate force-proportional analog signals with sensitivity in three mutually perpendicular axes. Five supporting elements with attached strain gauges were used to support the corners of the plate. To facilitate explanation, two schematics of the support system are shown in Figure 5.

With reference to Figure 5(a), each corner of the plate is suspended from the frame above by a short length of wire rope. The vertical support transducer, attached to the bottom of the wire rope, and to the plate, is sensitive to axial loading. The primary advantages of this pendulum-like support is that these vertical supports offer minimal restraint to horizontal movement and are therefore, only sensitive to the vertical force. Cross-talk, or the effect of one transducer on the next, is minimized by this mechanical arrangement.

Horizontal restraint and force measurement is explained as follows: with reference to Figure 5(b), horizontal support transducers are located at corners A and C. Corner A offers restraint in both the medio-lateral (y) and posterior-anterior (x) directions, while corner C offers restraint in the medio-lateral direction only. This mechanical arrangement minimizes cross-talk, both between the vertical transducers at corners A, B, C and the horizontal transducers at A, and C and between the medio-lateral and posterior-anterior transducers. To illustrate this, consider a small deformation "d" as shown in Figure 5(b). This deformation produces an axial strain in wire rope member Ax, and a lateral movement in members Ay and Cy. Using the Castigliano Theorem [59], it can be shown that nearly all of the posterior-anterior loading that produces this deformation would be taken by member Ax. The force in Ax produces a bending of the horizontal support element shown in Figure 6.

The arrangement of strain gauges further reduces crosstalk. For the horizontal suspension element shown in Figure 6, the strain gauges on opposing faces are grouped in pairs in the Wheatstone bridge circuit. For example, the applied force in Ax shown causes an elongation of face 2 and a shortening of face 4, to which the circuit is sensitive. To further illustrate this the Kirchoff formulas are given below (refer to Figure 6):

(1) $V_{ac} = V_{ab} - V_{cb}$

subst. $V_{ab} = V_{bd}\left[\frac{R_{a1}}{R_{a1} + R_{d2}}\right]$ and $V_{cb} = V_{bd}\left[\frac{R_{a2}}{R_{a2} + R_{d1}}\right]$

yields (2) $V_{ac} = V_{bd}\left[\left[\frac{R_{a1}}{R_{a1} + R_{d2}}\right] - \left[\frac{R_{a2}}{R_{a2} + R_{d1}}\right]\right]$

The value of the bracketed term above remains relatively unchanged when the strain-induced resistance changes in RA1 and RA2 are in the same sense, as would occur due to axial loading, or due to bending about an axis perpendicular to the paired faces. Both of these are, of course, due to cross-talk.

In a similar fashion, the strain gauges of the vertical suspension element shown in Figure 7 are arranged so as to be sensitive to axial strain only on the paired faces. This relationship is verified by the Kirchoff formulas:

$$(1) \quad V_{ac} = V_{ab} - V_{cb}$$

$$\text{subst. } V_{ab} = V_{bd} \left[ \frac{R_{a1}}{R_{a1} + R_{d2}} \right] \text{ and } V_{cb} = V_{bd} \left[ \frac{R_{d1}}{R_{a2} + R_{d1}} \right]$$

$$\text{yields } (2) \quad V_{ac} = V_{bd} \left[ \left[ \frac{R_{a1}}{R_{a1} + R_{d2}} \right] - \left[ \frac{R_{d1}}{R_{a2} + R_{d1}} \right] \right]$$

## 3.4 Signal Conditioning Hardware

As shown schematically in Figure 4 and in the photograph Figure 8, there are three units in the signal conditioning hardware. Signal processing of the six force channels consists of initial amplification and filtering ( 10 hz. lowpass ) by the strain-bridge amplifiers, further amplification by the conditioning amplifiers, and final analog to digital conversion by the Techmar A/D board. The shoe switches were connected directly to the A/D, bypassing the other two units.

## 3.5 Software

Four main programs, linked together in the manner shown in

Figure 9, were used to collect and process data. Listings of all programs are given in Appendix 1. The data acquisition program "GAITRM", written in Assembler language, read the eight channels from the A/D at a user selected rate of up to 1000 hz. This data was stored initially in RAM, and following the completion of a "run" or six traverses of the force board, the data was written to a floppy disk. Time skewing, or the elapsed time between the reading of successive channels, was very low. Provision was made in "GAITRM" to read the outputs of each channel with no load applied to the force board. This allowed for subsequent corrections for the D.C. float. The triggering of "GAITRM" was based on the physical loading of the board. The digital sum of the three vertical force channels represented the start and stop triggers. A start was initiated when, for five consecutive data points, the change in this digital sum was positive and greater than a threshold value. Using this edge approach versus a voltage level approach, the problem of a false start due to D.C. float was eliminated. The relatively gradual unloading of the plate made it difficult to use this edge approach to stop the program from reading data. Instead, a voltage level approach was used and a stop was indicated when 15 consecutive readings were taken which were lower than a threshold level, this level having been adjusted for the D.C float.

The intent of the calibration program was to relate the digitized voltage levels to the corresponding forces that produced them. Following a calibration procedure described in more detail in the User's Guide [60], the calibration factors for all transducers were determined and stored on a floppy disk.

The data analyses programs read in the raw gait data from the data disk, corrected for D.C. offset, and applied the calibration factors which were also read in from the floppy disk, to give true force data. Using true force data, the centers of pressure and various other parameters were calculated as further explained in Chapter 4.

Preparatory to the actual data run, a resolution test was done using the program "RESOLVE" to ensure that all of the vertical channels were reading correctly as gauged by the accuracy in determining the center of pressure.

### 3.6 Force Board Performance

#### 3.6.1 Vibrational Properties

For the purposes of studying its vibrational properties, the force board can be modeled as shown in Figure 10. Applying this model to each direction of support, the effective overall spring constant at a given corner is considered to result from the series combination of the constants of a number of stiffer springs connected in series. The component springs represent a particular subassembly of the overall support stucture, and its effect on the rigidity of the support can be determined by removing it from the system. Since the force(s) generated at the subject's foot or feet is variable in magnitude, direction, and in point(s) of application, the governing differential equations describing the oscillatory motion of the force board system is complex and beyond the scope of this research. Clearly the resonant natural frequencies in the vertical, medio-lateral, posterior-anterior, and rotational senses are variable and dependent on the foot-floor reaction force and its location. This notwithstanding, a series of vibration tests were performed on the force board with two specific objectives in mind:

(1) For the purposes of data collection, to establish the base level resonant frequencies. This would allow for an appropriate selection of the cut-off frequency for the low-pass filter and predict vibrational related artifacts.

(2) To determine where in the structure design improvements were warranted, specifically to identify those areas where an increase in rigidity would be most cost effective.

Two series of tests were performed on different dates. In both cases the vibrations were induced by striking the board lightly at its center point with a closed fist. The vibrations were monitored in two ways; using a hand held "Davey" vibrometer (strip recorder), and digitally using the data acquisition program to read the output of the strain gauges at 500 hz. Since the two methods were in good agreement, the results listed in Table 1 are an average of the two.

In the first series of tests, the force board was left unloaded. The vertical resonant frequency was measured at 30 hz. for the normally suspended force board. In test 1(b), the wire rope connection was released and each of the corners were rested on the tubular crossbars as shown in Figure 11. Supported in this manner, the vertical resonant frequency increased to 38 hz. To examine the stiffness of the board itself, (i.e. independent of its supports), the three corners were rigidly supported from the floor using concrete blocks (test 1(c)). In this case the vertical resonant frequency was measured at 65 hz.

For the second series of tests, a 100 pound weight was placed at the center of the normally suspended force board. The results of these tests are given in Table 1.

3.6.2 Resolution of Center of Pressure Tests

Tests were done on a regular basis to determine the resolution and accuracy with which the location of the center of pressure could be predicted. These tests were of particular concern since this parameter was used extensively in data analysis software.

Three marks were made at random locations on the force board, and their measured coordinates determined. Using a sharp object ( a center punch), hand pressure was applied at one location and then transferred to the second and third locations. Using the program "RESOLVE", the location of the computer predicted center of pressure was compared to the actual measured location. The results of a typical resolution test are shown in Figure 12 and Table 2.

CHAPTER 4

# THEORY AND DEVELOPMENT OF ALGORITHMS

## 4.1 Introduction

The purpose of this chapter is to describe the experimental approach used, and its underlying philosophy, the theory and the history of evolution of the five algorithms that were developed. A mention of the history of this developmental process is made since it very much affected the ultimate direction and outcome of this study.

From the beginning of this research, the primary objective had always been to develop two algorithms to extract single limb force information from a composite record, specifically that record being any single, multiple-step traverse of the force board. Both of the extraction algorithms envisioned (described in more detail in section 4.6) required the times of heel strikes and toe-offs. The first stage of algorithm development was seen as identifying these temporal events.

Brief consideration was given to using an ultrasonic proximity sensor or some type of laser/mirror equipment to detect all foot contact transitions. However, in keeping with an intention to limit the cost and complexity of the system, these ideas were quickly abandoned. With reference to Figure 2, Brodland [54] had already demonstrated that both "on-board" and "off-board" heel strikes could be identified, the former using his center of pressure based algorithm, and the latter by contact sensitive areas. It was decided to develop an algorithm along the same line. In retrospect, what was seen at the time as a small first stage of development, was a gross underestimation of the task ahead.

Confident that the Brodland algorithm could be successfully developed, it was decided that other parameters would be explored in the hope of identifying new indicators of heel strike times. Some promising such indicators observed early in this process of development led author on a tangent, albeit a successful one, away from the original objective.

While a method of identifying heel strikes had been decided upon, a method of determining toe-off times was still needed. A review of Brodland's apparatus showed promise. He had used "force-sensitive" areas to identify the "off-board" heel strikes. These consisted of spring-loaded, hinged boards placed on either side of the force plate. The movement of these boards was detected using sensitive micro-switches. It was decided that a similar arrangement could indicate toe-offs in addition to the heel strikes. These however, would be "off-board" toe-offs. The "on-board" toe-offs would be found by assuming a temporal repetition of the gait cycle. For example, assuming that the time from a right heel strike to a left toe-off would be the same one complete cycle later, then the "on-board" toe-off time could be determined by adding the period of double support to the time of the heel strike. With reference to Figure 2, the following equations apply:

$$T(to) = T(hs) + \text{double support period}$$

where T(to)= time of toe-off, T(hs)=time of heel strike

## 4.2 Base Data Collection

The first set of data collected served several important functions. It provided a data base upon which all of the algorithms were developed, as well as establishing the threshold values for each. In addition, it led to improvements in the hardware, protocol and software related to data acquisition.

Using eight normal subjects, described in table 3, gait data was collected over a period of several days. Preparatory to a data run, metal ducting tape was placed on the sole of each foot and electrical connection made as described in section 3.2, and as shown in Figure 1. For each "run" or a set of six traverses of the force plate, the subjects were instructed to walk normally and not to "aim", or otherwise alter his or her gait in order to hit the force plate squarely. For each traverse the foot-fall pattern was monitored and recorded (e.g. the code LRL indicates a left-right-left sequence of steps). In the event that a foot landed both on the force plate and the walkway, the data for that traverse was flagged and later discarded. For each run, two traverses each of single step, two step, and three step sequences were taken. The various gait patterns were achieved by having the subject walk along an unmarked "lane". For example, to obtain a single step traverse, the subject would walk close to the apex of the triangle, and towards the middle of the board for a two step pattern.

Data was collected at 60 hz. sampling rate using an earlier version of "GAITRM" than the one described in section 3.5. This earlier version used a voltage level versus the edge approach to trigger the start and stop of data reading. This manner of triggering did not compensate for the D.C. float and caused a good deal of grief before it was finally corrected.

Near the end of this phase of data collection it was observed that the ducting tape had been breaking at about the mid-sole location. The implication of this was that, although the shoe-identified heel strikes were accurate, the toe-off times were not valid (a toe-off time was in fact a "heel-off" time). This did not represent a problem at the time since the method of finding toe-off times, as discussed earlier, had already been decided upon. When an alternate toe-off algorithm was later developed, it did mean however that this data was unusuable for this purpose.

## 4.3 Editing of Data

As discussed in section 3.5, the editing of data consisted of the conversion of raw data (voltage values) into meaningful force data. As illustrated in Figure 9, this was done by reading in the raw data, correcting for D.C. offset, and then reading in and applying the calibration factors. At the time of collection of this data, this was done using the program "EDITROSS" (listed in Appendix 2). The edited data (force) was stored on a disk for subsequent analysis.

## 4.4 Data Analysis and Display Programs

### 4.4.1 Introduction

In order to find new indicators of heel strike times, it was necessary to derive several parameters, and to display them in such a way that any relationship between a given parameter and the time of heel contact would be apparent. After the data had been edited using "EDITROSS", the force data was read in by the analysis program and several parameters related to the center of pressure were determined, these being: the x and y locations of the center of pressure (CPX,CPY respectively), the velocities of the center of pressure in both directions (VCPX,VCPY), the resultant velocity of the center of pressure (RVCP) and the corresponding angle of this resultant velocity (VA). With reference to Figure 5(a), the equations for determining the first two of these are given below, while further explanation of the others is reserved for later sections.

$$(1)\quad CPX = \frac{(B/2 + C).L}{A + B + C}$$

$$(2)\quad CPY = \frac{B.h}{A + B + C} = \frac{B.(L/2)}{A + B + C}$$

A printout of selected data allowed for the identification of heel strike times based on the shoe-switch information, these being later superimposed on all plots as described in the following sections. The printouts were also used to establish the threshold values of parameters used by the heel strike algorithms.

4.4.2 Plotting Program

The intention of this program was to graphically display all of the parameters discussed in section 4.4.1 with the provision for a superpositioning of the heel-strike times given by the shoe-switches. The program "PLOTRM" displayed a main menu from which 3 types of plots could be selected:

(1) A plot of all three forces; vertical, medio-lateral, and posterior-anterior (see Figure 13)

(2) A plot of all center of pressure-related parameters and the vertical force (see Figure 14)

(3) A plot of the dynamic location (x and y coordinates) of the center of pressure (see Figure 15)

The plot of the center of pressure related parameters confirmed Brodland's findings and also led to the development of the other algorithms discussed in the following section, and to the development of the toe-off algorithm at a later point as explained in section 5.4. The third plot called "PECK" , a sample of which is shown in Figure 15, was an attempt to relate three parameters (x and y coordinates of center of pressure, and time). One peck mark was made at the coordinates of each center of pressure against an outline of the force board. The temporal influence is revealed by the density of the cluster of dots.

## 4.5 Heel Strike Identification Algorithms

### 4.5.1 Background

Since much of this research has depended on the pioneering work of Brodland [54], it is appropriate at this point to review the theory and basis of his heel strike algorithm.

For normal locomotion, there is a continuous net forward displacement of the center of vertical force. The rate of this is not uniform with time throughout the gait cycle. In the periods of single limb support, the center of pressure must obviously be within the confines of the foot contact area, or as shown in Figure 16, between the limits M and N in the sagittal plane. During this time the other, unloaded limb is being swung forward from a lagging to a leading position relative to the loaded foot, and when the heel of this swung limb first contacts the ground, a rapid forward progression of the center of pressure is initiated. The onset of this rapid progression (i.e. its velocity) in the direction of locomotion causes a "corner" in the CPX vs. time plot as evident in Figure 14, and identifies the heel strike time using the Brodland algorithm.

Before leaving discussion of this algorithm, there are a few shortcomings that must be pointed out. The algorithm assumes that the stride distance is large in relation to the foot length, and ignores, for the most part, localized movements of the center of pressure on each foot. For example, in the case of an "overlapping " short stride, shown schematically in Figure 17(a), the secondary heel strike would not produce a significant increase in the forward progression of the center of pressure. In the extreme case it may in fact cause a temporary backwards movement of the same. This limitation renders the algorithm unsuitable for analyzing a short shuffling type of gait which is often characteristic of the elderly.

A second problem with the algorithm is that it assumes that the majority of limb motion occurs in the sagittal plane. While Brodland had calculated the medio-lateral position of the center of pressure, it was not incorporated in his algorithm. To illustrate the significance of this point, consider the gait pattern shown from above in Figure 17(b). While the heel strike of step 2 may be easily identified, the time of the cane-strike cannot be identified using the Brodland algorithm.

### 4.5.2 Algorithm 1 (Resultant Velocity of the Center of Pressure)

In response to the shortcomings listed in the previous sections, it was decided to somehow develop a method of identifying heel strikes based on the net movement of the center of pressure. Brief consideration was given to representing the center of pressure in polar rather than Cartesian coordinates. This idea was abandoned however due to the difficulty in establishing an appropriate origin point. Since the overall movement versus the location of the center of pressure is the important parameter, it was decided to use the resultant velocity of the center of pressure (RVCP) as determined by the Pythagorean relation:

$$RVCP = (VCPX^2 + VCPY^2)^{1/2}$$

The component velocities were initially obtained using the first forward difference but this method was later replaced by the Finite Impulse Response (F.I.R.) differentiation technique [61].

### 4.5.3 Algorithm 2 (Velocity Angle of Center of Pressure)

Having calculated the resultant velocity of the center of pressure, it was a simple task to find the resultant velocity angle (VA) as shown below:

$$VA = \tan^{-1}(VCPY/VCPX)$$

This velocity angle, determined at each data point, was plotted in hopes that some "corner" may be evident at the time of heel strike. However, for reasons discussed later, this plot was erratic as shown in Figure 14 and provided no information when plotted in this manner.

4.5.4 Algorithm 3 (Second Difference of Vertical Force SDVSUM)

The third algorithm, found quite by surprise, is based on the total vertical force rather than on the relative individual loads as for the first two methods. From early tests it was noted that a marked increase in the vertical force, or a notch occurs at the time of heel strike as shown in Figure 13. In retrospect, this seems to be an obvious expectation, yet the fact that this had gone unnoticed by previous researchers is curious. For each data point in the vertical force curve, the second finite difference was calculated. A heel strike was identified when this second difference exceeded a certain threshold value which was determined from the data printout.

4.6 Single Limb Force Extraction Algorithms

4.6.1 Introduction

For a typical gait pattern at normal walking speed, the period of dual support is only about 13 % of the total gait cycle time. While the total vertical force is being recorded at this time, it is impossible to determine how this total load is being shared by the two supporting limbs. While there are many advantages to large force boards, some of which were mentioned in section 1.1, this inability to resolve individual limb forces creates definite limitations. A review of the literature failed to reveal any studies regarding the problem of extracting single limb force histories. The algorithms developed and discussed in the following sections are a response to this deficiency. Both of the algorithms require the times of all heel strikes and toe-offs. These can be found using the algorithms previously discussed or obtained in other ways, by cine analysis for example. The development of the two groups of algorithms (i.e. heel strike and force extraction) are independent of one another and have been treated as such in this thesis.

4.6.2 Algorithm A (Temporal Shift)

The underlying assumption of this algorithm is that the gait cycle is repeated one full gait cycle later. For example, all of the forces generated by the left foot must be the same two strides later. One of the limitations of this method is that it is only applicable on a three step sequence, a right-left-right ("RLR") for example.

There are two ways in which the algorithm can be utilized. Considering the force record shown in Figure 18, the complete right record can be obtained by a backwards shifting of the trailing edge of the left record as shown. While this shifted portion is for the dual support phase, it is in fact, single limb force data since the right foot is off the force board at this time. This shifting, of course, requires knowledge of the two equivalent right heel strike times.

Using the second method, the leading edge of the lead foot force record can be time shifted forward as shown in Figure 18, this giving a second right foot in our example. In either case, after the shift has been made, the force information for the other foot can be determined by subtracting the shifted portion from the composite record.

A completely desynthesized record determined in this way is shown in Figure 18(b). One of the "errors" in this method, is that a small difference in the force value may occur at the shift points. The implication and analysis of this is discussed in section 5.3.2.

### 4.6.3 Algorithm B (Fractional Split)

This algorithm is based on the analysis of the position of the center of pressure in relation to the two foot placements during double support. Unlike the temporal shift approach, this algorithm works for both two and three step sequences. To facilitate the explanation of this method, we will first consider the gait of a hypothetical individual having two peg-legs. The individual force on each limb is thus restricted to a point location.

As shown in Figure 19(a), during double support, the location of the center of pressure is somewhere between the two pegs, or at the general point CPX(t) in the sagittal plane. The points CPX(hs) and CPX(to) are the center of pressure locations at heel strike and toe-off times respectively. Considering moments about point CPX(hs), the fraction of loading on the left and right feet can be expressed as:

$$\frac{FR}{T} = \frac{D}{D + E}$$

$$\frac{FR}{T} = \frac{CPX(t) - CPX(hs)}{CPX(to) - CPX(hs)}$$

$$\frac{FL}{T} = 1 - \frac{FR}{T}$$

where R,L are the forces in the right and left foot

and T is the total vertical force (i.e. L + R)

The reality is however, that each foot is of finite size, and consideration must be given to the local progression of the center of pressure on each foot. As shown in Figure 19(b), these local positions of centers of pressure "d1" and "d2" can be referenced to CPX(hs) and CPX(to) respectively. In theory, information concerning this localized progression of the center of pressure on a single foot during the double support phase is availible. For example, for a "LRL" cycle, the center of pressure on the left foot is known from the time of its initial heel strike to the time of the "off-board" toe-off of the right foot, and upon exiting the board, from the time of the "off-board" left heel strike to the time of the "on-board" toe-off of the right foot. Both of these periods are double support phases, but in each case, only one limb is on the force board. While this is fine in theory, the rotational vibration of the board produce errors in this region of the CPX(t) curve as shown by the circled areas in Figure 14.

Barring the collection of better data , the assumption was made that, during this double support phase, each localized center of pressure advances forward at a constant rate. For the trailing foot, this rate was assumed to be the same as it was just before heel strike and for the leading foot the same as just after toe-off, or along the tangents shown in Figure 19(c). This assumption is reasonable based on the findings of Yamashita[9]. Using this refinement to the "peg-leg" model the resulting equations become:

$$\frac{FR}{T} = \frac{D}{D+E} = \frac{CPX(t) - (CPX(hs) + d1)}{(CPX(to) - d2) - (CPX(hs) + d1)}$$

where $d1 = VCPX.t$ (left foot)
and $d2 = VCPX.t$ (right foot)

$$\frac{FL}{T} = 1 - \frac{FR}{T}$$

# CHAPTER 5

## RESULTS AND DISCUSSION

### 5.1 Force Board Performance

#### 5.1.1 Center of Pressure

Using a test procedure described in section 3.6.2, the center of pressure of three measured points was determined. The data given in Table 2, and displayed in the plot shown in Figure 12, indicates an accuracy of about 1/2" or less than 0.5% error in the center of pressure detection. This compares extremely well to the 5%-10% accuracy reported by Yamashita [9] in his studies using a similar sized board. In terms of dynamic response, the plot in Figure 12 indicates that there was little variation of the location of the center of pressure from the straight line connecting the test points. For vertical loads of under about 10 pounds, the center of pressure becomes unreliable. Also, as evidenced by the initial portion of the CPX plot in Figure 14, the center of pressure oscillates following a heel strike, presumably as a result of rotational vibration about the y axis. Unfortunately, as mentioned in section 4.6.3, this vibration renders the information unusable for measuring the localized center of pressure movement on the first foot fall on the force board.

The "PECK" method of dispaying the temporal location of the center of pressure seems to provide a simple, yet meaningful display. A traverse made by a 23 pound, 21 month old boy shows that at the lower limit of weight, the foot patterns and center of pressure movement is well displayed as indicated in Figure 20. At the other weight extreme is the "PECK" plot of a 200 pound subject with a pathological gait as shown in Figure 21. This form of representing center of pressure can also be used in assessing the geometric variables associated with gait, such as stride length, the relative positions of the footfalls, and may also indicate when a foot has landed at the edge of the force board.

5.1.2 Vibrational Properties

While this force board has demonstrated good sensitivity, produces minimal cross-talk, and locates the center of pressure with high accuracy, it is in need of stiffer supports to increase the resonant natural frequency, especially in the horizontal plane. The intent of this section is to suggest design changes as indicated by the vibrational study.

In the vertical sense, the marginally acceptable natural frequency of 30 hz. was increased to only 38 hz. by removing the effect of the vertical support transducer and its supporting wire rope. The most substantial increase in stiffness was achieved by removing the spring equivalent of the transversely loaded tubular cross member shown in Figure 11. This resulted in an increase from 38 hz. from 65 hz., and strongly suggests the need to stiffen this member, either by replacing it with a larger or thicker-walled one, or by providing additional supports to the existing by way of gussets or struts.

The resonant frequencies in the horizontal directions, 23 hz. medio-laterally and 17 hz. in the posterior-anterior direction, are unacceptably low and render horizontal force measurements questionable with the present configuration. As is evident from the horizontal force plots in Figure 13, the signal/noise ratio is indeed very low, and clearly illustrates the need for improvement. (This is in part is why the author's research has focused on vertical forces).

Although the horizontal supports were not tested in the same manner as the vertical supports, there are several projections which can be made regarding design improvements. In general, an improved design can be achieved by following one of two approaches:

(1) retaining the basic support geometry but strengthen selected areas.

(2) redesigning the entire horizontal support configuration (leaving the vertical arrangement as is).

In regards to the first approach choice, the following suggestions are listed in the decreasing order of their projected cost/benefit:

(a) stiffen the cantilever support post "A" a shown in Figure 11 by the addition of gusset plates or struts.

(b) provide a broader base to the support transducers to reduce the "rocking" tendency. ( A modified transducer shown in Figure 32 was made but has yet to be installed)

(c) shorten the cables "D" in Figure 11 or replace them with stiffer ones.

(d) replace or "beef-up" the 3/16" floor plates in the corner units (see Figure 11)

While it is agreed that an acceptable overall stiffness could be achieved using the above or similar changes, it is probably more advisable that a total redesign of the horizontal support system be made since, in addition to achieving higher natural frequencies, a redesign could also eliminate some of the alignment problem's discussed in the User's Guide [60].

In reviewing Balakrishnan's thesis [55], it is clear that his design of the force board was based on the constraint that the support system be statically determinant. Since an analog approach was used at the time, this was necessary in order to simplify the process. However since the development of a digital system by Brodland, this is no longer a requirement. Unlike many Engineering problems which require the determination of reactions due to an applied known force, the situation with the force plate is one of reconstruction. If the reactions at each corner are known, (as indicated by the strain gauge readings) it is a simple task to determine the magnitude, direction, and point of application of the causative force. Removal of the determinancy constraint allows for any number of supports and permits a much more rigid support, the trade-off being a more expensive data acquisition system.

A relatively simple change that may be considered, is the addition of a posterior-anterior support at corner B as shown in Figure 33. With the present configuration, a posterior-anterior force applied at point "p" produces a couple C = d x f with the reactive force in the P/A support at corner 1. This couple produces equal and opposite reactions in the M/L restraints at corners 1 and 3. In theory, this should not produce an erroneous net medio-lateral force indication since this is obtained by algebraically summing the two equal and opposite readings. It does however greatly reduce the signal/noise ratio . In addition this effect initiates a rotational vibration about the vertical axis.

## 5.2 Testing of the Algorithms

### 5.2.1 Introduction

As mentioned in section 4.1, two main blocks of data were collected at different times, the first block being used for developmental purposes, and the second block for the testing of the developed algorithms. Due to the time lapse between the collection of these two data sets, several refinements had been made to the data collection process, these being:

(1) addition of the provision for a 125 hz. sampling rate to the "GAITRM" program. (Previous data was collected at 60 hz.)

(2) modification of the start/stop triggering of "GAITRM" as described in section 3.5

(3) improvements in the calibration procedure which increased the center of pressure accuracy from 5 % to about 0.5%

(4) Correction of the problem of the breakage of the ducting tape on the shoe switches as discussed in section 4.2.

(5) An inclusion in the test protocol of an immediate resolution test using the program "RESOLVE" to verify correct readings prior to any data collection.

(6) improvements to the procedure for the aligning of the force board.

While all of the above changes led to the collection of more accurate data, the improvements to the shoe switches now meant that the toe-off times were reliable. This led to the late development of the toe-off identification algorithm. It is for this reason that it is discussed in this chapter rather than earlier.

### 5.2.2 Acquisition of Test Data

Test data was collected from three subjects, two males and one female, described briefly in Table 3. One of the male subjects had a highly asymmetrical, pathological gait and wore corrective footwear on his left foot (see table 3 for details). The other two subjects walked in a normal fashion.

Each subject was advised to walk normally and to avoid "aiming" for the force board. By monitoring each traverse, unusable data was flagged and not used in the subsequent analysis. For each subject 24 traverses each of the single left and right feet were made, 24 traverses of two-step patterns (12 each of LR and RL), and 24 traverses of three-step patterns (12 each of LRL and RLR). As explained further in section 5.5.1, a large data base of single foot data was collected for purposes of comparing the extracted single-limb force records.

Some problems were encountered in the collection of data from the subject with the abnormal gait. His normal stride was too long for a three footfall pattern on the force board. Also due to an error in the data acquisition program, and due to the fact that the subject tired more readily, less usable data was collected on this subject than for the other two.

## 5.3 Results of Heel Strike Identification Algorithms

An assessment of the success of the Resultant Velocity of the Center of Pressure (RVCP) and the Second Difference of the Vertical Force (SDVSUM) heel strike algorithms was made by comparing the algorithm-identified heel strike times to the "exact" times indicated by the shoe switches. For each heel strike, the positive or negative deviation from the exact times was determined in terms of the number of datapoints at the sampling rate used for data collection. As shown in the histograms in Figures 22 and 23, the results are grouped in three ways: (1) on an overall basis (all 113 heelstrikes included), (2) on the basis of normal versus abnormal gait (92 normal gait heel strikes and 21 abnormal gait heel strikes), and (3) according to the individual feet of the subject exhibiting the abnormal gait (13 heel strikes on his right or "good" foot and 8 heel strikes on his left foot, the one with the built-up shoe).

In general, the accuracy of both methods is extremely good. The RVCP method predicts a heel strike within an overall accuracy of 0.68 of a data point (based on R.M.S). At the sampling rate of 125 hz. used, this represents an "error" of about 27 milliseconds. For the SDVSUM algorithm, the corresponding figures are slightly higher at 0.86 of a datapoint or 34 milliseconds. These figures compare well to the accuracy of 0.5 datapoint reported by Brodland [54], especially in light of the fact that he had analyzed a total of only 12 heel strikes using a single subject.

As indicated on the top graphs in Figures 22 and 23, the data is grouped in a relatively narrow band about the zero point, with some isolated scatter to the right, this representing late heel strike prediction. Viewing the data on an normal/abnormal gait basis, reveals that most of this "late" band occurs in the subject exhibiting abnormal gait. A further breakdown of this subject's data reveals that virtually all of this "late" band occurs on the subject's left foot, the one with the built-up shoe. This suggests that the subject has a tendency to "scuff" his left foot just prior to the application of any significant loading. As shown in Figure 24(a), the leading edge of the vertical force plot for the left foot indicates that this foot accepts load at a low rate in the early stages, this in contrast with the right foot, the leading edge of which is shown in Figure 24(b) below. The delay in load acceptance is also evident in the area indicated in Figure 24(b). This raises an interesting question as to whether or not this "late" prediction of heel strikes is a shortcoming of the algorithm. While the purist might argue that a heel strike occurs as soon as any contact exists between the board and the subject' foot is made, the biomechanist may be more interested in the time at which the foot begins to accept a "reasonable" rate of loading , (this adjective being loosely defined). This in mind, the algorithm-identified heel strikes might prove better than those identified by cine methods when used in making some Newtonian connection between forces and body kinematics.

To test the ability of the algorithms to identify an "overlapping" gait pattern as defined in section 4.5.1 and as illustrated in Figure 17(a), a subject was instructed to walk such that the medio-lateral separation between steps was high and the posterior-anterior separation was low. The vertical force and center of pressure plot ("PECK") of this traverse are shown in Figure 25. Both algorithms identified the heel strike time exactly.

## 5.4. Toe-off Identification Algorithm

As mentioned earlier, the development of the toe-off identification algorithm occured at a relatively late stage in this research. On the plots of the derived parameters, as illustrated in Figure 14, a corner, albeit a more subtle one, was observed at the end of the dual support phase, that is at the toe-off point , on the CPX, and CPY plots. Threshold values of the RVCP parameter, corresponding to this corner point, were obtained from the printouts and used to develop an algorithm to identify the times of toe-off. In total, 65 toe-offs were identified within an average R.M.S deviation of 0.44 of a datapoint, and exhibiting a relatively small spread as shown in Figure 26. To the best of the author's knowledge, this is the first time that toe-off times have been successfully identified based on force records.

## 5.5 Analysis of Force Extraction Algorithms

### 5.5.1 Introduction

In general, the assessment of the two force extraction algorithms was based on comparing one or more parameters of a "master" average one-step record with the corresponding extracted record. This master record was obtained by normalizing each of the 24 single foot records (per foot) to a common time base and finding an average. In addition, the relative spread (standard deviation) was determined for each data point in the master plot, and an envelope of +/- one standard deviation from the average record plotted as shown in Figures 28 and 30. To eliminate possible error due to heel strike and toe-off identifications, the shoe-switch identified values were used.

5.5.2 Temporal Shift Algorithm

The software and approach used to assess the Temporal shift algorithm is shown in Figure 27(a). Only a single parameter, that being the force value at the second, "off-board" heel-strike time (used to mark the shift points), was used to gauge the relative success of this algorithm. The rationale for this is explained as follows: Since this method uses single support force data, unaffected by the time shifting, the small "gap" at the shift point, labelled "X" in Figure 27(a), is the only possible error that might be introduced by the method. The gap is due either to the normal statistical variation of force values, due to the non-repetition of the step one stride later, or else it may be due to some locational bias in terms of force readings on the force board. If, for example, the board tended to give a low reading of force with the subject on the east side of the board, then one would expect that the standard deviation of the distribution of this gap "X" would be greater than root two of the standard deviation of the force values at this point, as obtained from the analysis of the single-step records [62]. For each foot of both subject, the results, shown in Table 4, suggested that, indeed, the "X" gap could be explained statistically in terms of the normal inter-step variabilty, and is not due to an artifact of the force board or this algorithm.

5.5.3 Fractional Split Algorithm

The success of this method was assessed using the software and approach illustrated in Figure 27(b). While the method can be applied to both two and three step gait patterns, it was decided that the middle force of three step patterns be extracted, a right on an "LRL" pattern for example. Using the program "FRACTION", the middle force was extracted from the three step pattern, and normalized to a common time base. Twelve records each of the left and right feet were determined in this manner. Once all of the records had been processed an average extracted record, and the standard deviation envelope for the record was determined. These master extracted records are shown in Figures 29 and 31. The results of the comparison of the master one-step records with the master extracted records are discouraging. Clearly the extracted portions contain a great deal of the vibration-related artifact. But yet another error is present, unrelated to the algorithm. The middle, single-support section of the force record, i.e. from toe-off to the next "off-board" heel strike indicates a marked difference. Since this is in no way related to the fractional split algorithm, it suggests that vertical forces occuring in the center portion of the board do not produce the same readings as an equivalent traverse made close to the apex of the triangle, i.e a single step traverse. Further testing of the algorithm should be made after improvements to the system rigidity are made as discussed earlier.

CHAPTER 6

# CONCLUSIONS

Two new algorithms for heel strike detection have been successfully developed. The RVCP algorithm has predicted heel strike times to within 0.68 of a sample point, and the SDVSUM to within 0.86 of a sample point. Both of these algorithms were successfull for irregular gait patterns.

An algorithm for the detection of toe-off times was also successfully developed and is capable of an accuracy of 0.44 of a sample point in prediction time.

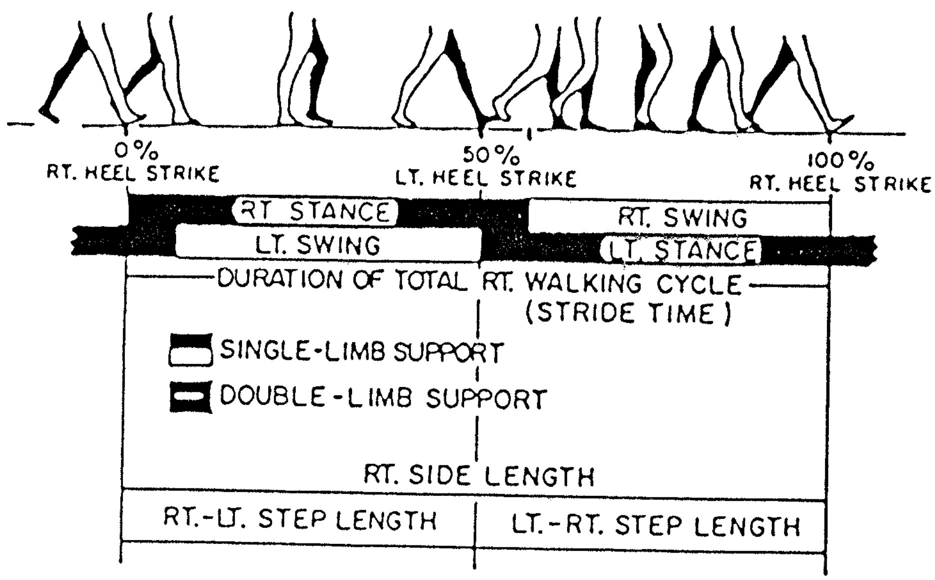
During a multiple-step traverse of the force board, the period of dual support can now be determined successfully based on the force record alone. The implications of this are that a multiple footfall force plate can be used without any patient preparation to determine the single and double support times and assymmetry of gait. Clinics currently require foot switches and cine data for this analysis.

Two algorithms, based on the identification of the dual support phases as determined using the above algorithms, can be used to find force records for a single limb. The Temporal Shift algorithm, applicable on three step patterns only, produces highly accurate extracted records. The results of the Fractional Split algorithm, applicable to any muliti-step pattern were very much affected by artifacts of the force board relating to its lack of rigidity.

The dynamic location of the center of pressure displayed by the "PECK" method provides a meaningful visual tool for gait analysis. It displays the location of the footfalls which is useful in analyzing the distance factors related to gait (e.g. stride length, step width, etc.,). This would also indicate questionable steps which have occurred near the edge of the force board. The "PECK" method also displays the relative rates at which each of the limbs accept loading. A disadvantage of this method is that a true time history is not given. While the eye assumes a continuous advancement of the center of pressure along the curved line, it cannot display a backwards movement, albeit that these reversed movements are assumed to be due to force artifacts.

The rigidity of the suspension system is too low, especially in the horizontal direction. The resulting oscillations are of relatively low frequency and high amplitude and are under-damped. The attendant force artifacts very much limit the reliability of some of the readings and thereby restrict the scope and nature of further studies and suggests the need for design improvements to the force board suspension.

Figure 1 : Photograph of Force Board System



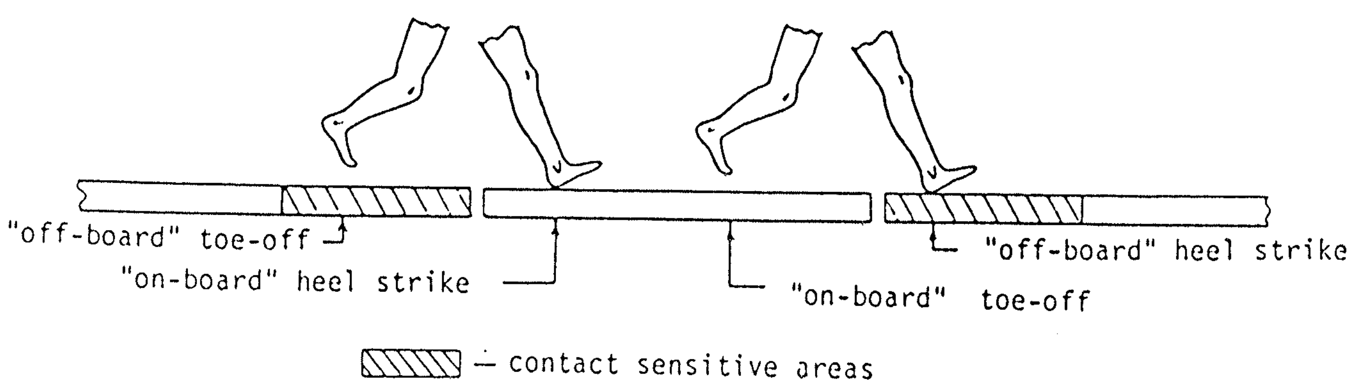
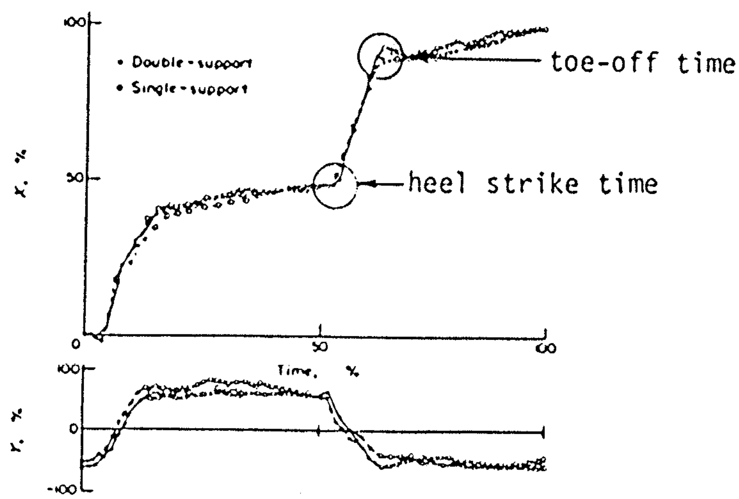
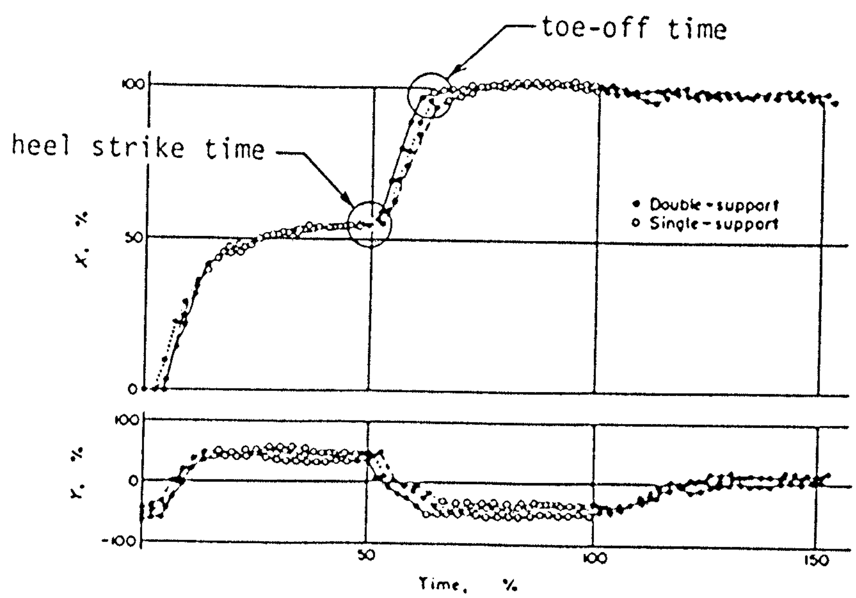




Figure 2: Temporal Events of a Normal Gait Cycle




Normalized moving pattern of point of application in steady walking.




Moving pattern of point of application in stopping.

T-toe-off time H-heel strike time

Figure 3: Center of Pressure Plots from Yamashita and Katoh Study [9]

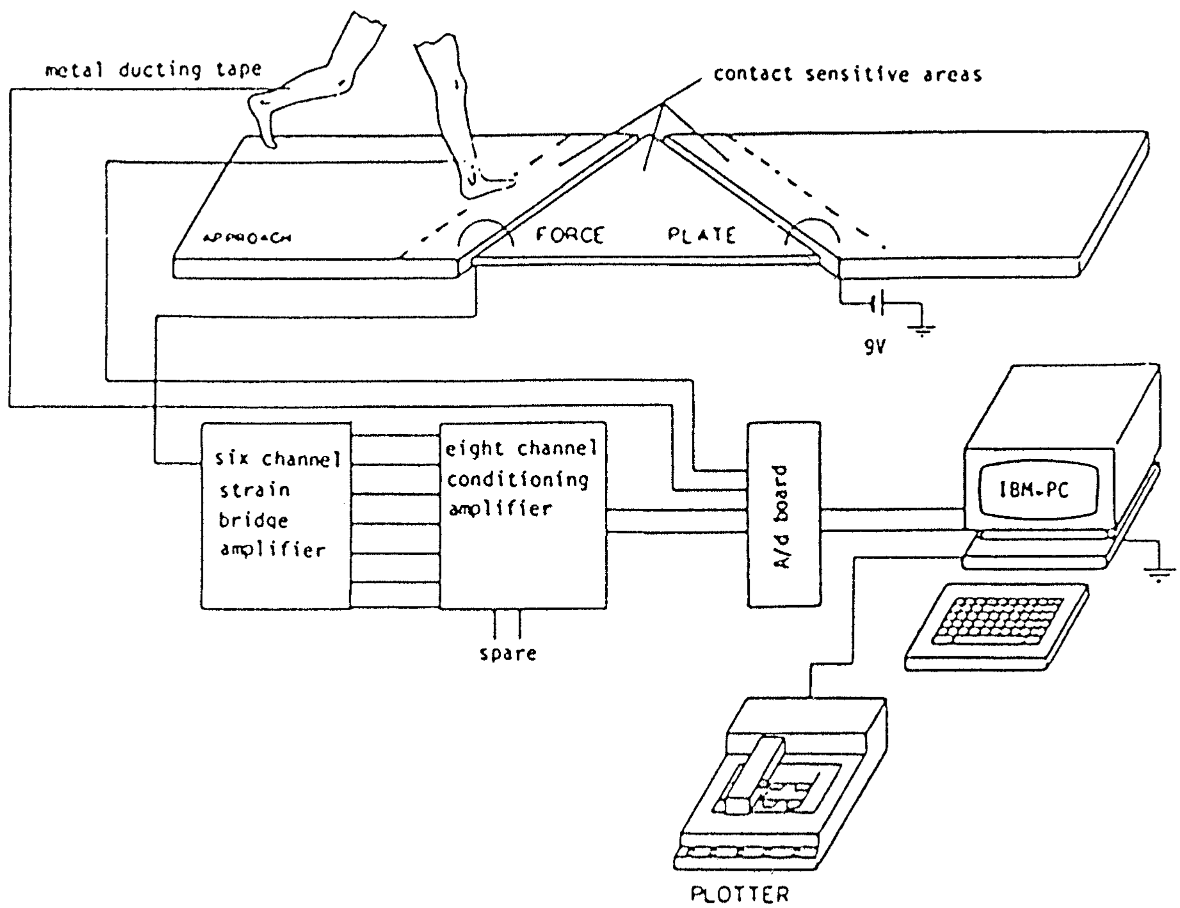



Figure 4: Schematic of Force Board System

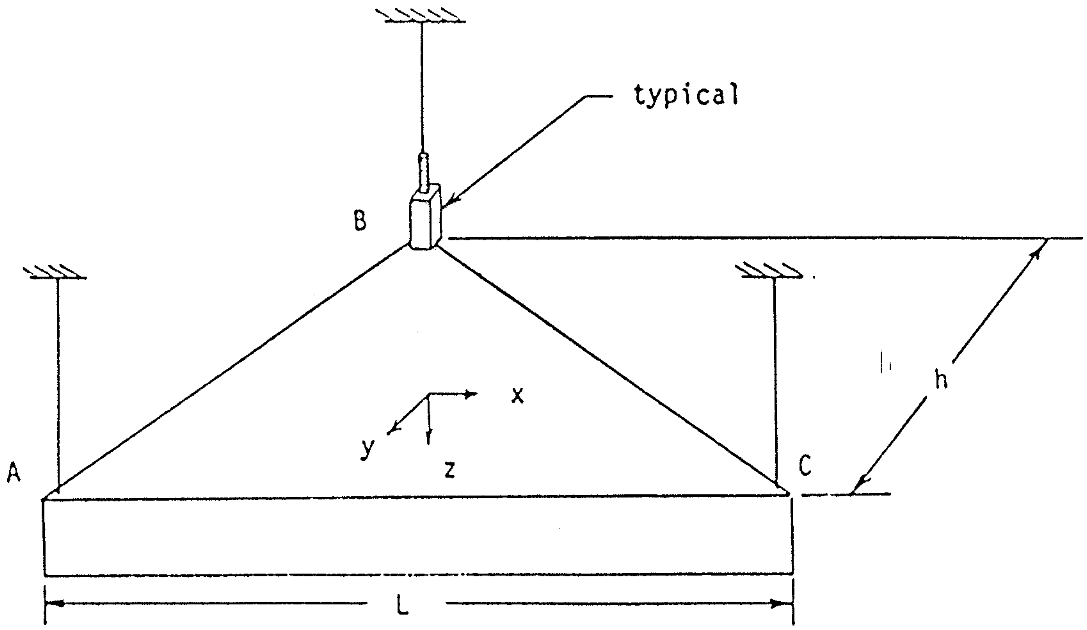



Figure 5(a): Schematic of Vertical Board Support

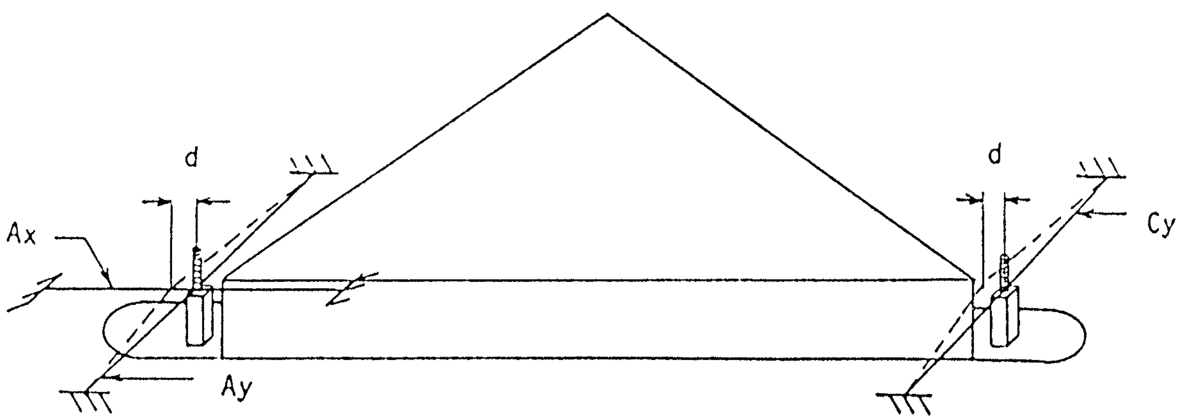



Figure 5(b): Schematic of Horizontal Board Support

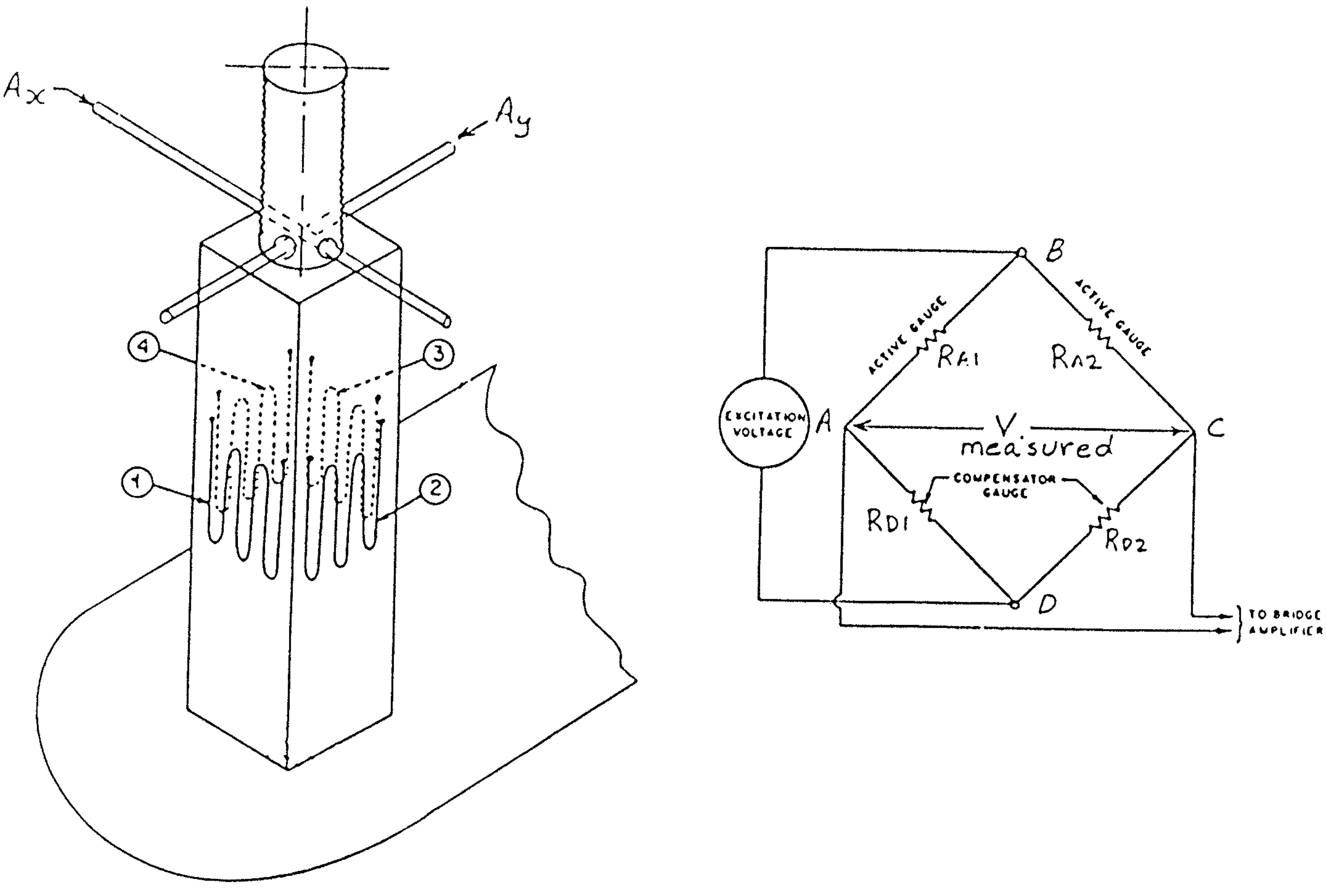



Figure 6: Horizontal Support Transducer

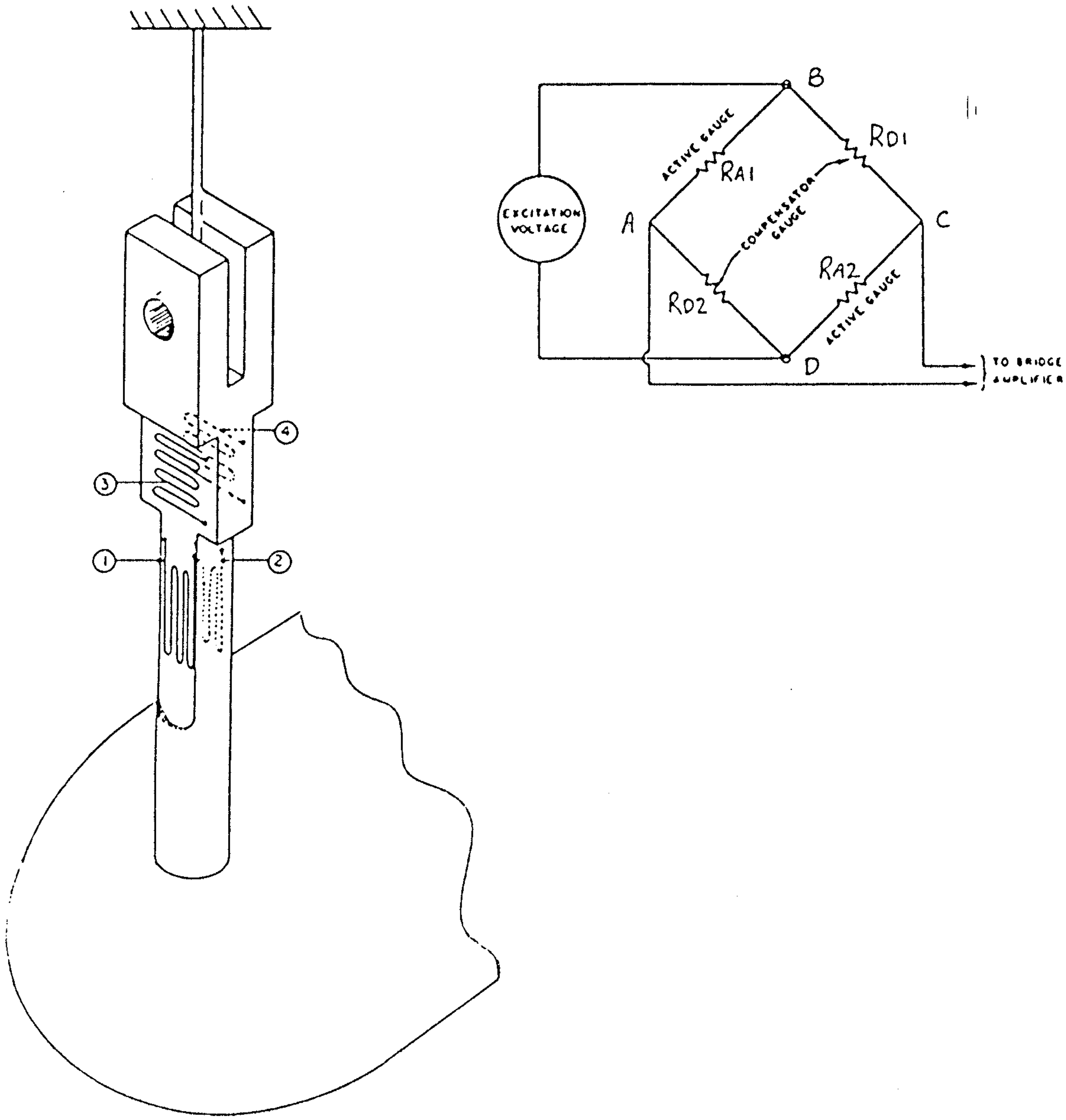



Figure 7: Vertical Support Transducer

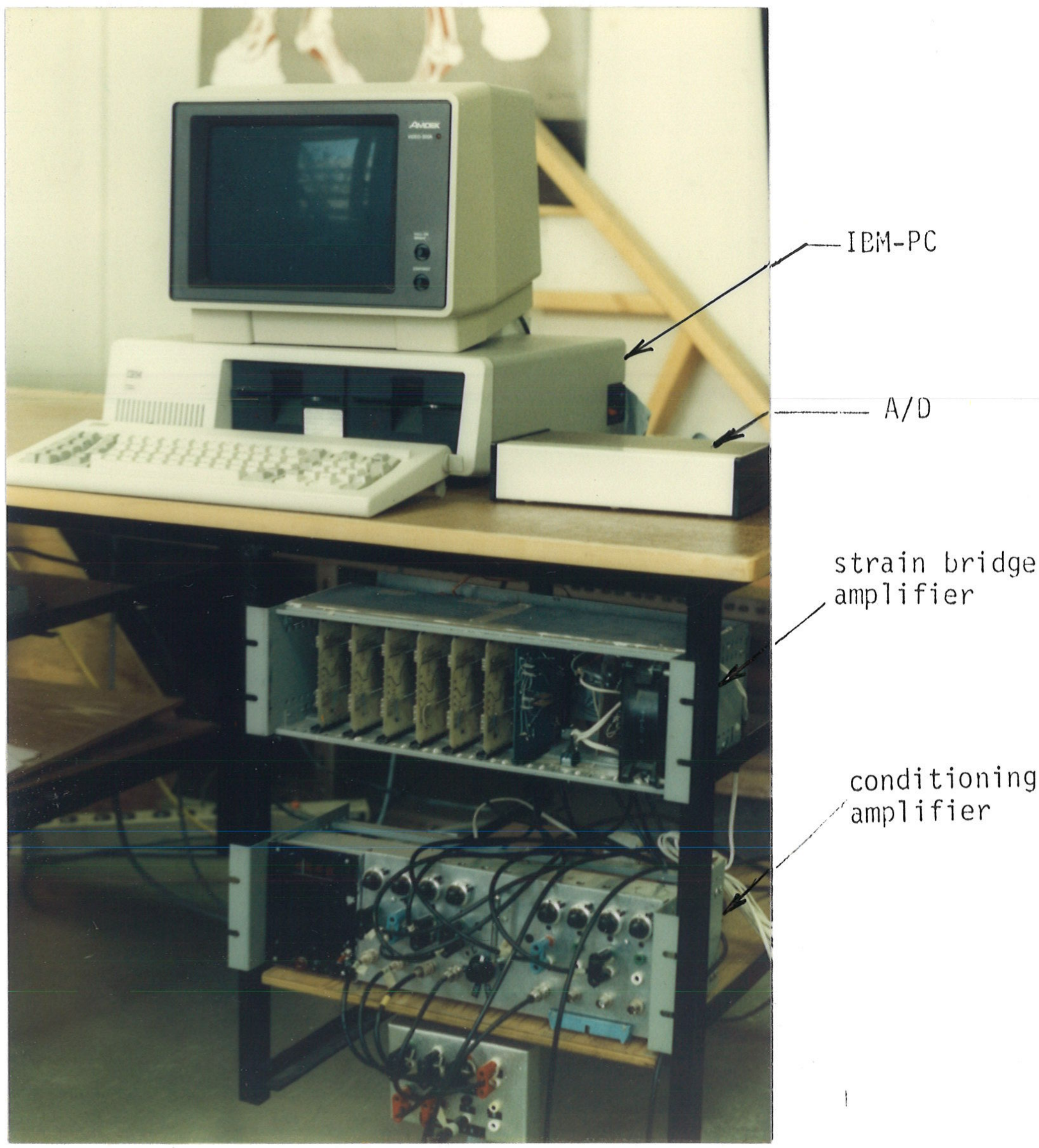



**Figure 8: Photograph of Signal Conditioning Hardware**

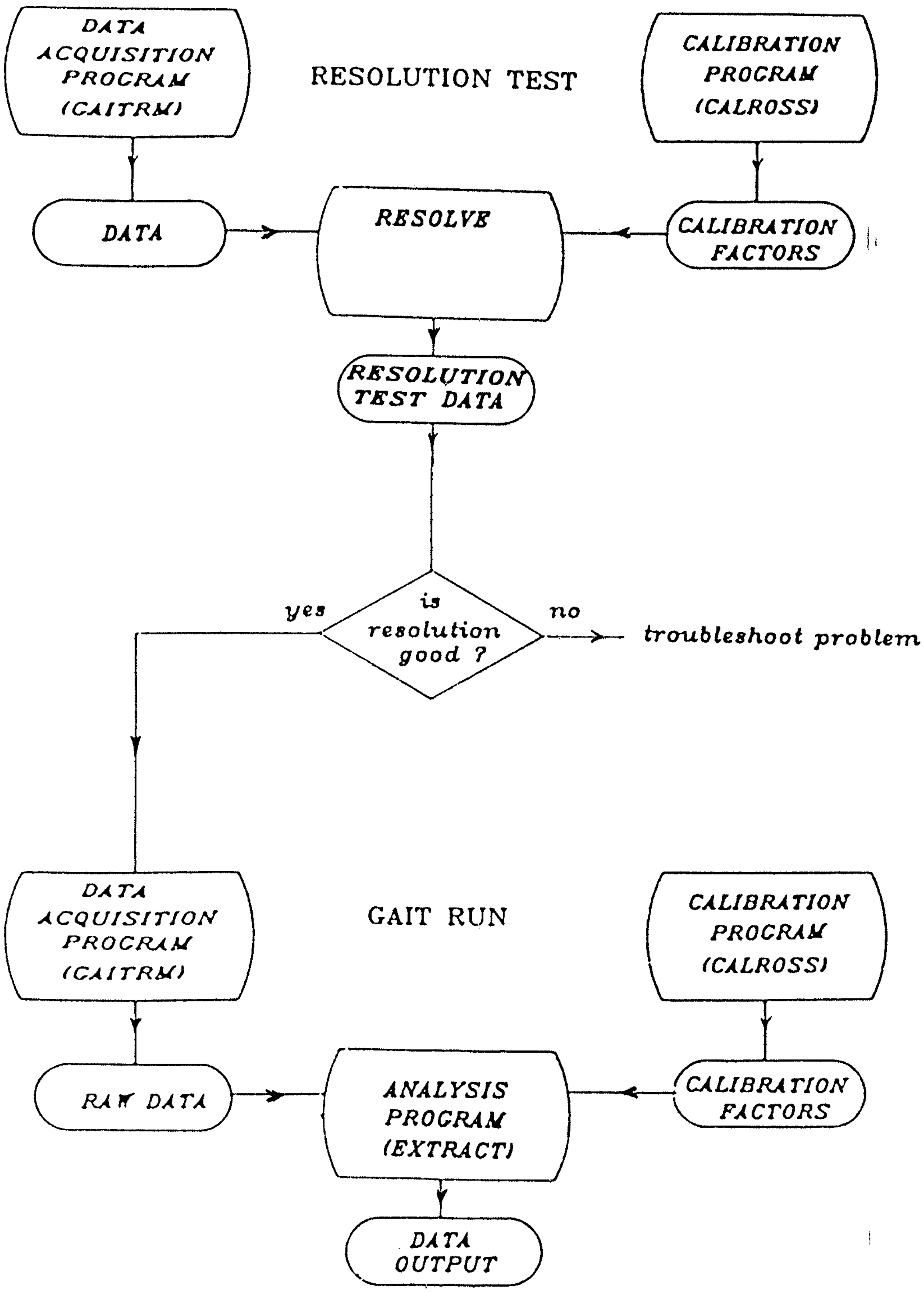
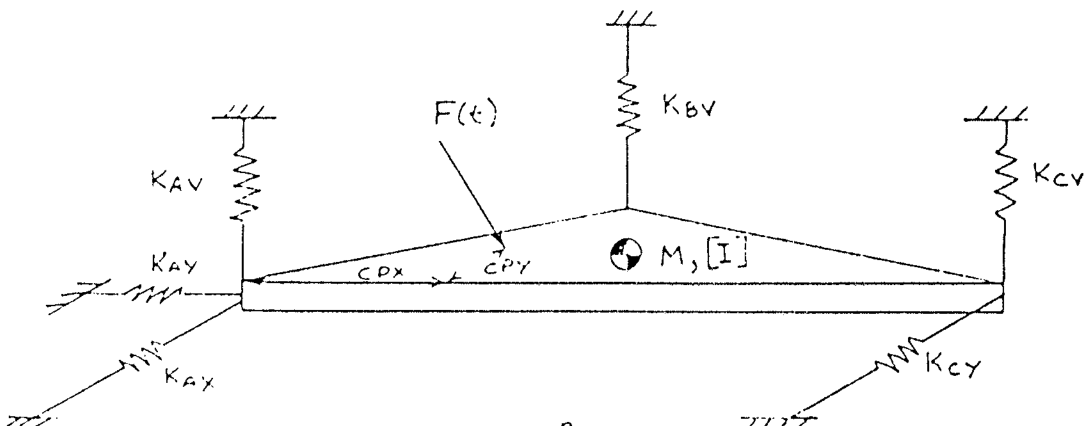



Figure 9: Software Overview Block Diagram




$$\frac{1}{K_{eff.}} = \sum_{i=1}^{n} \frac{1}{K_i}$$

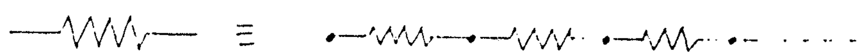


Figure 10: Spring Model of Force Board Suspension

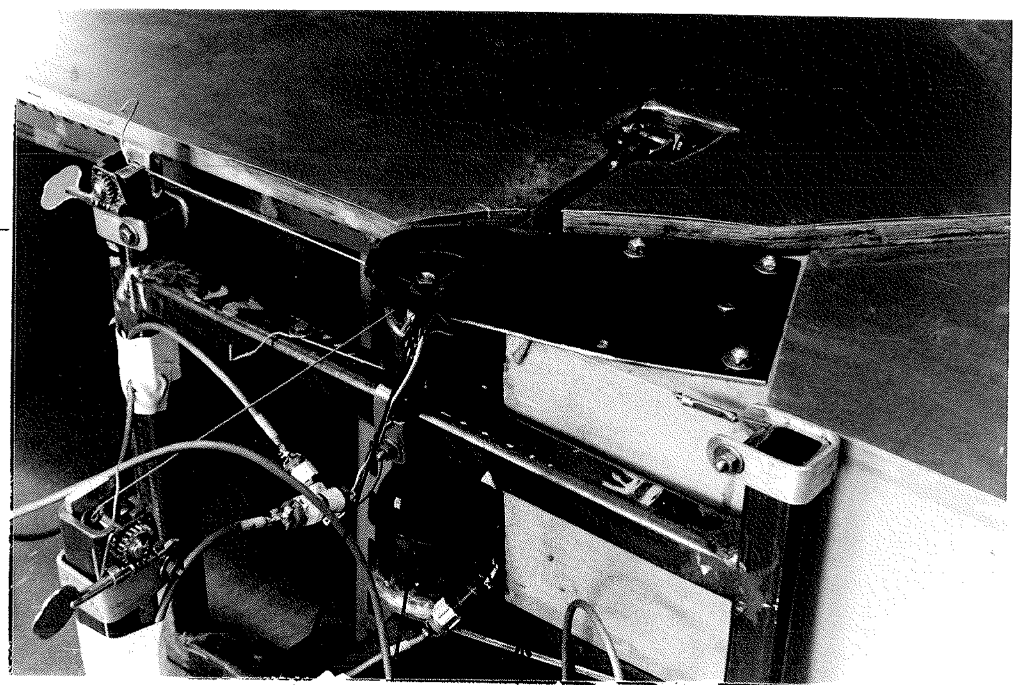



Figure 11: Photograph of Corner Support Structure

FILENAME: rt280885.1

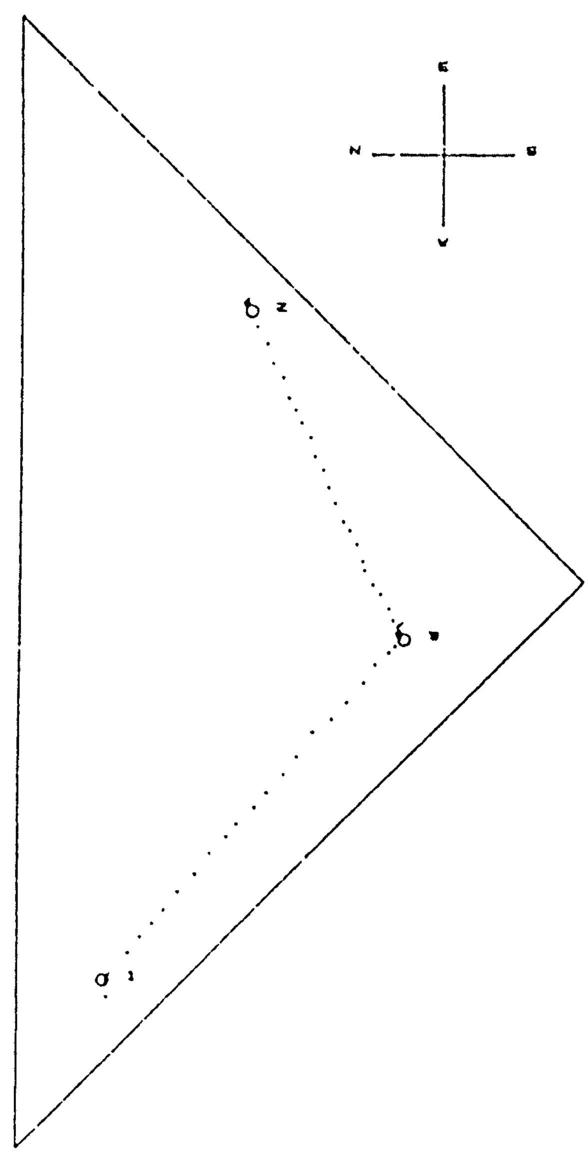


Figure 12: Typical Resolution Test Plot ("PECK")

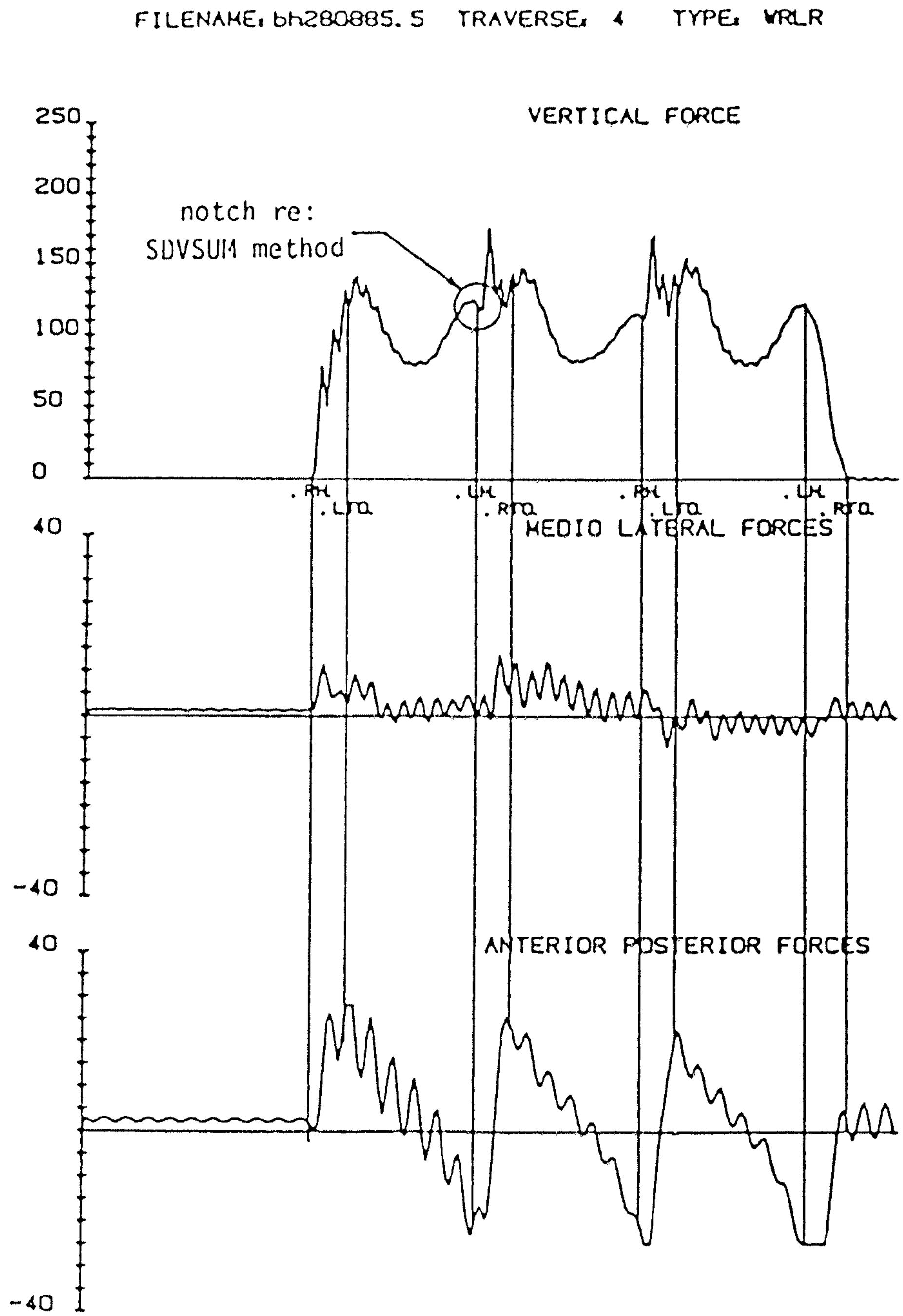



Figure 13: Sample Three-Forces Plot

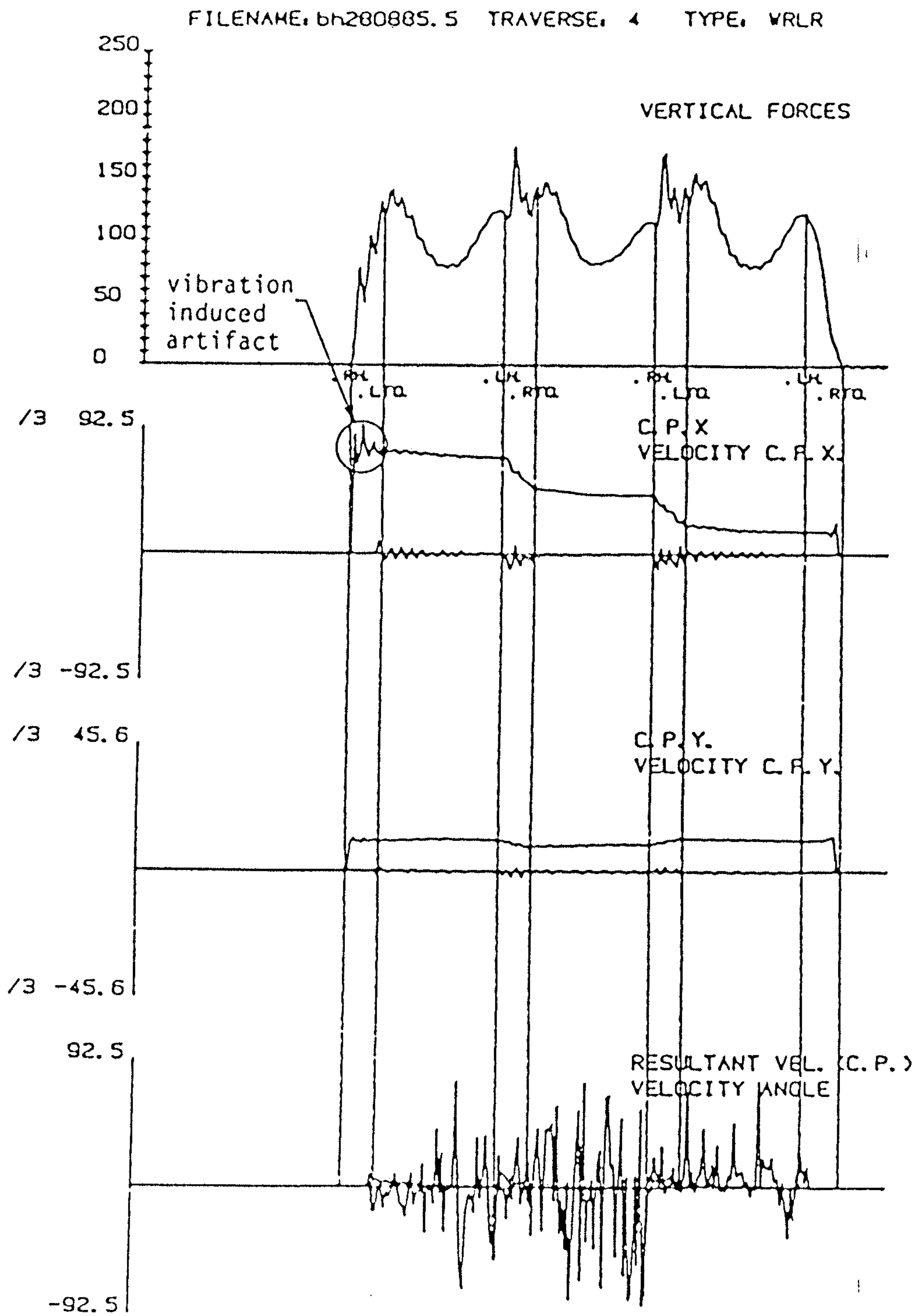



Figure 14: Sample Plot of Derived Parameters

FILENAME: bh280885.5 TRAVERSE: 4 TYPE: WRLR

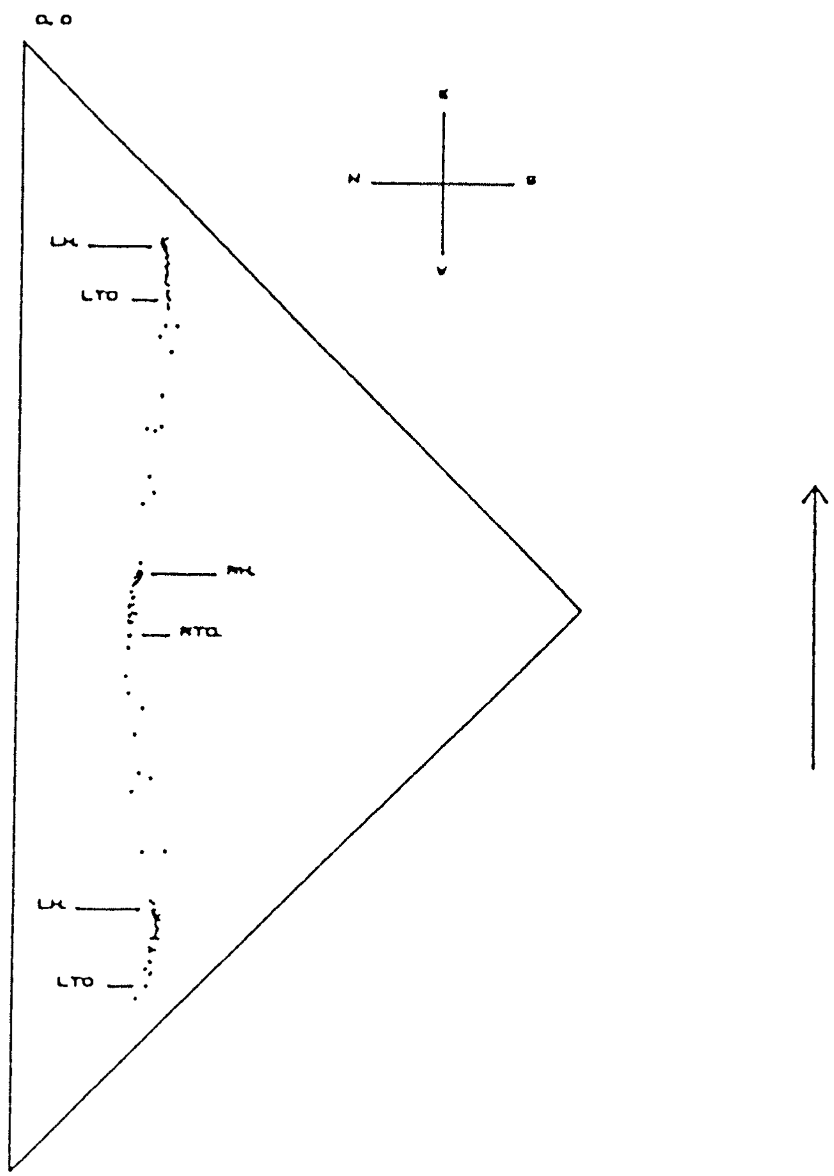


Figure 15: Sample Center of Pressure Plot ("PECK")

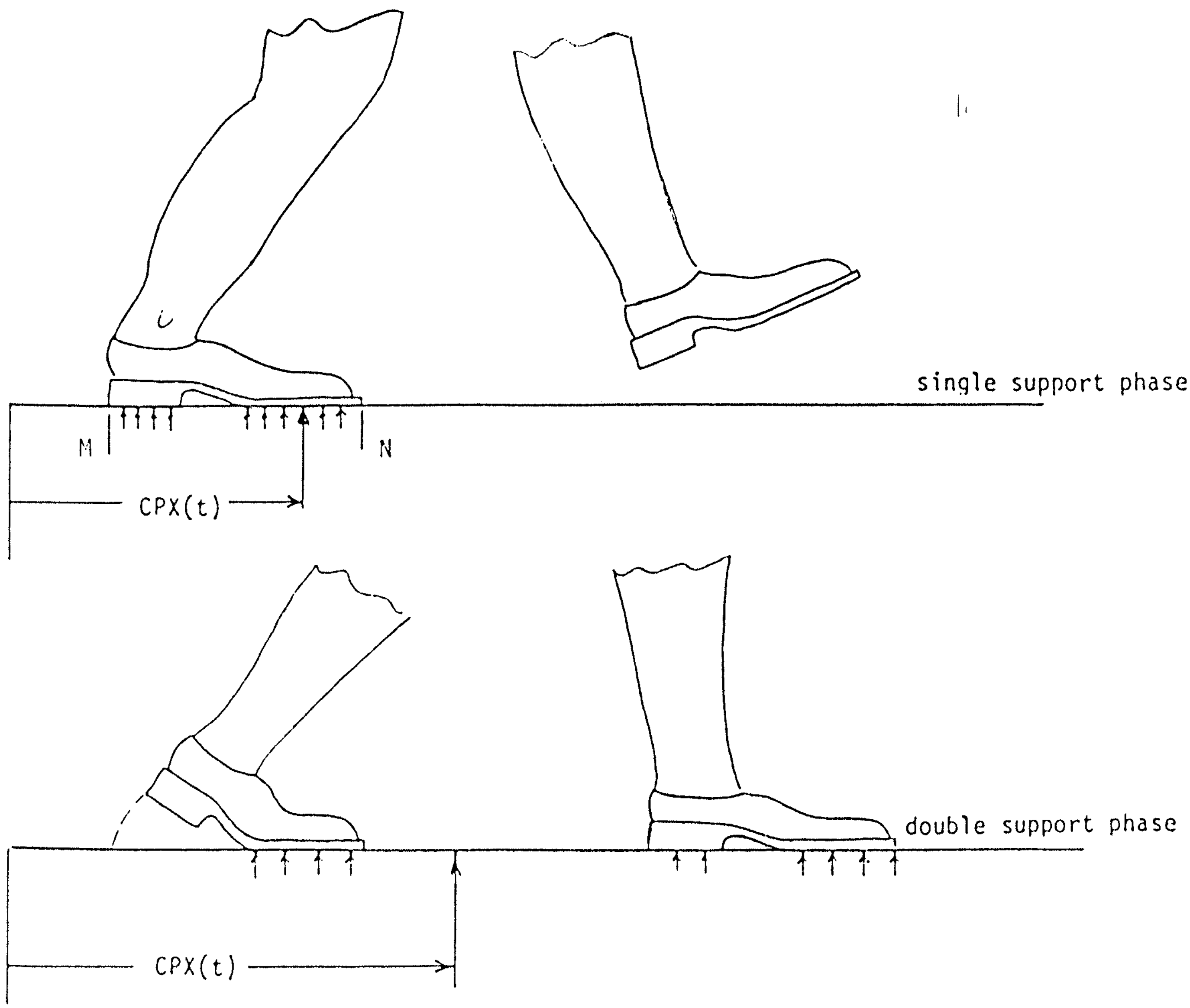
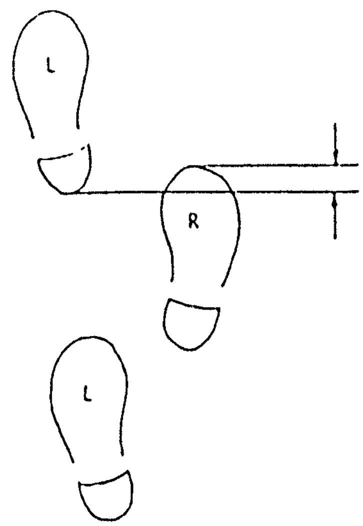
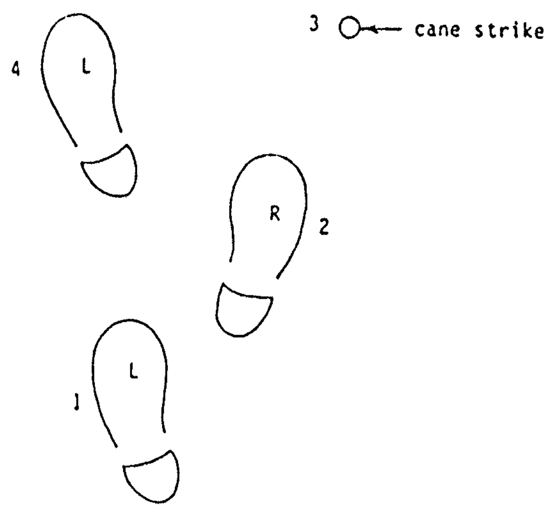



Figure 16: Center of Pressure Relative to Foot Position




(a) overlapping gait




(b) cane-assisted gait

Figure 17: Abnormal Gait Patterns (a,b)

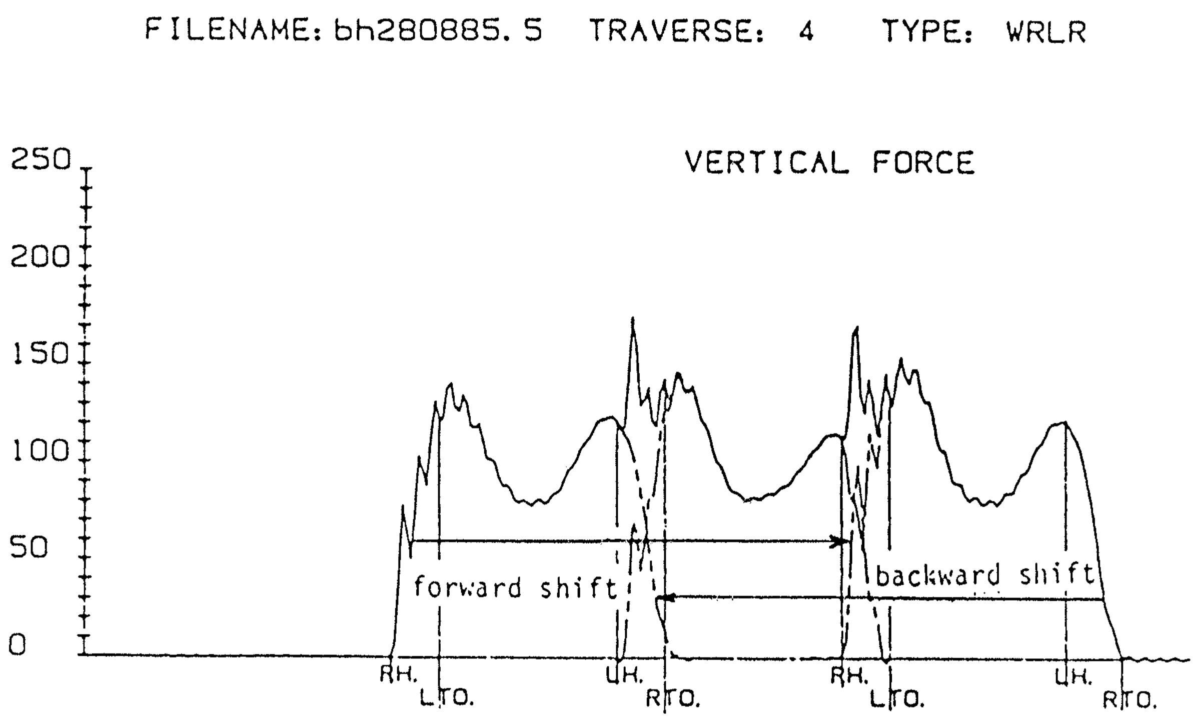



Figure 18: Temporal Shift Single Limb Force Extraction

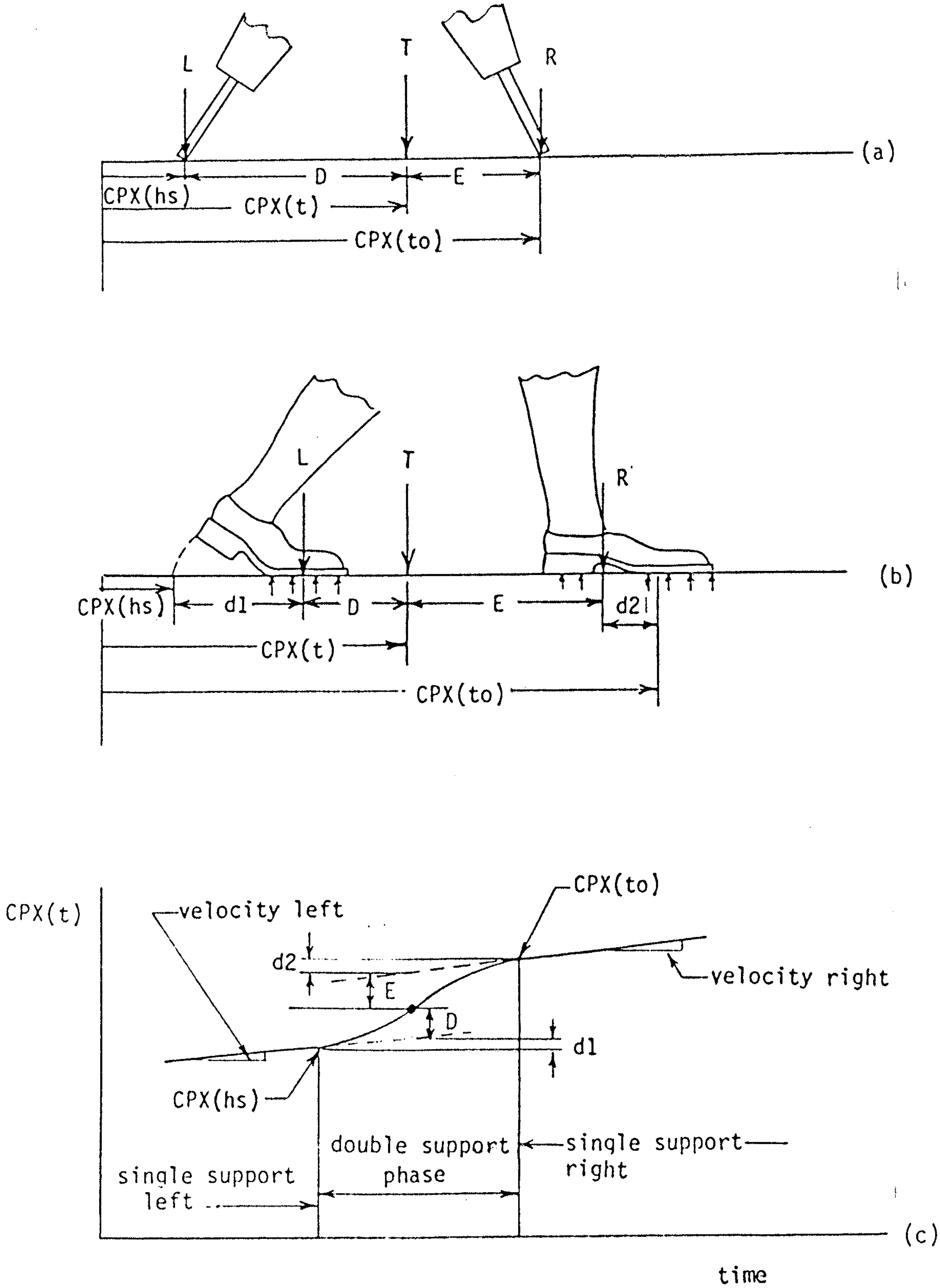



Figure 19: Fractional Split Single Limb Force Extraction

FILENAME: mm260885.1 TRAVERSE: 3 TYPE:

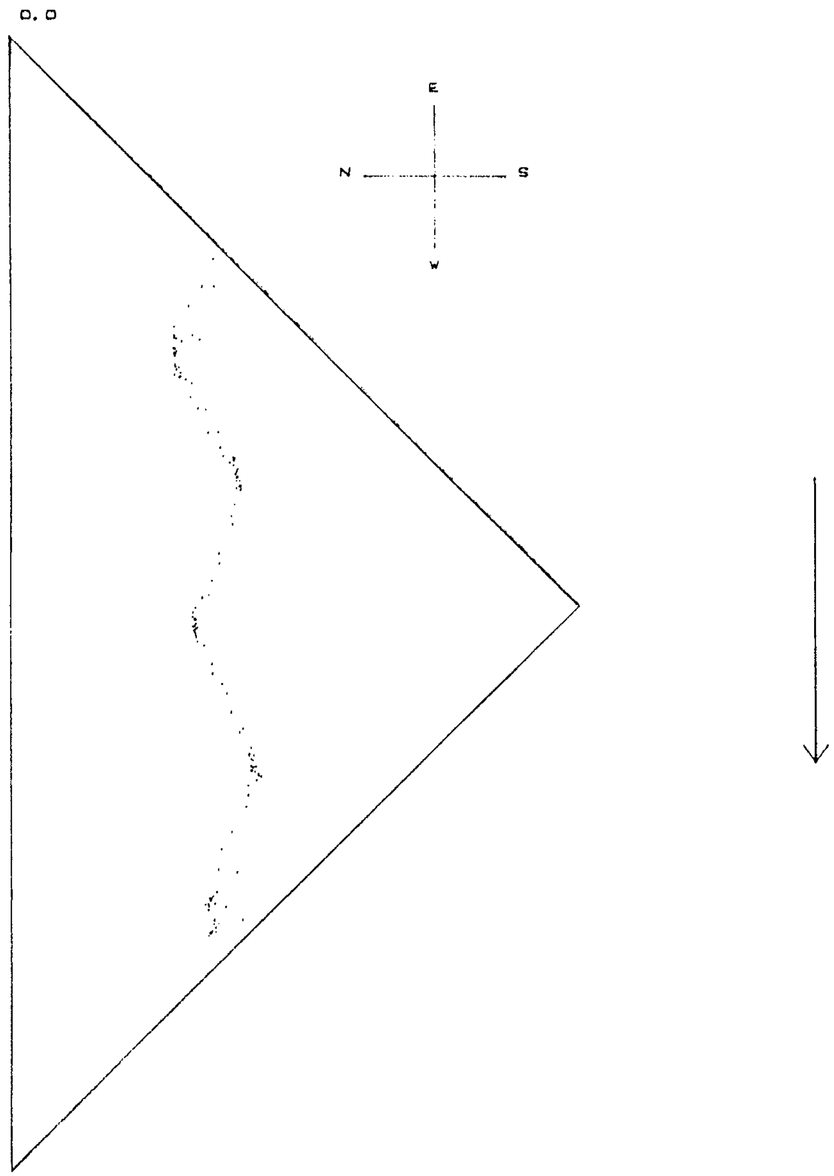



Figure 20: "PECK" Plot for a Twenty-Three Pound Infant

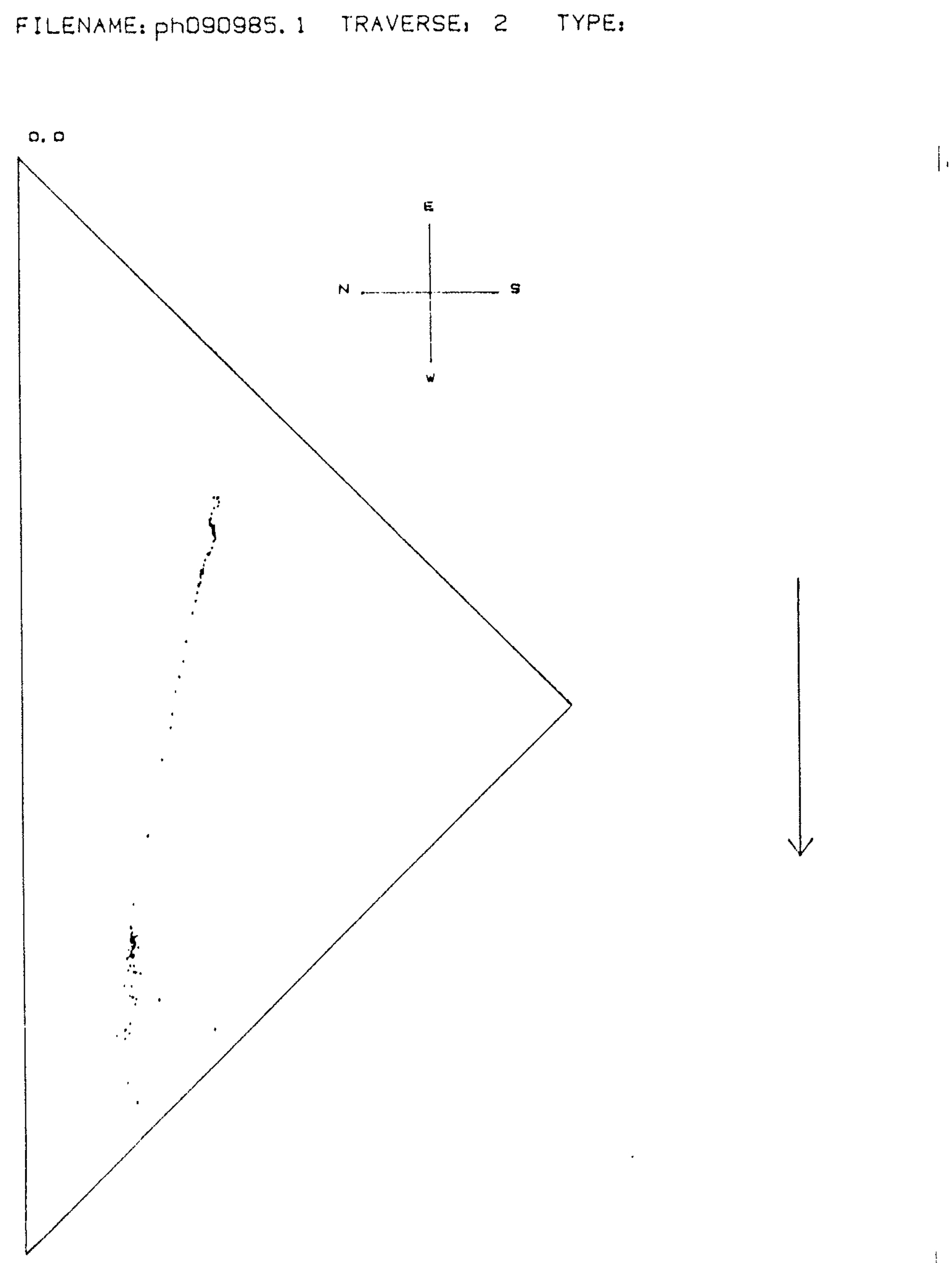



Figure 21: "PECK" for 200 lb Subject With Pathological Gait

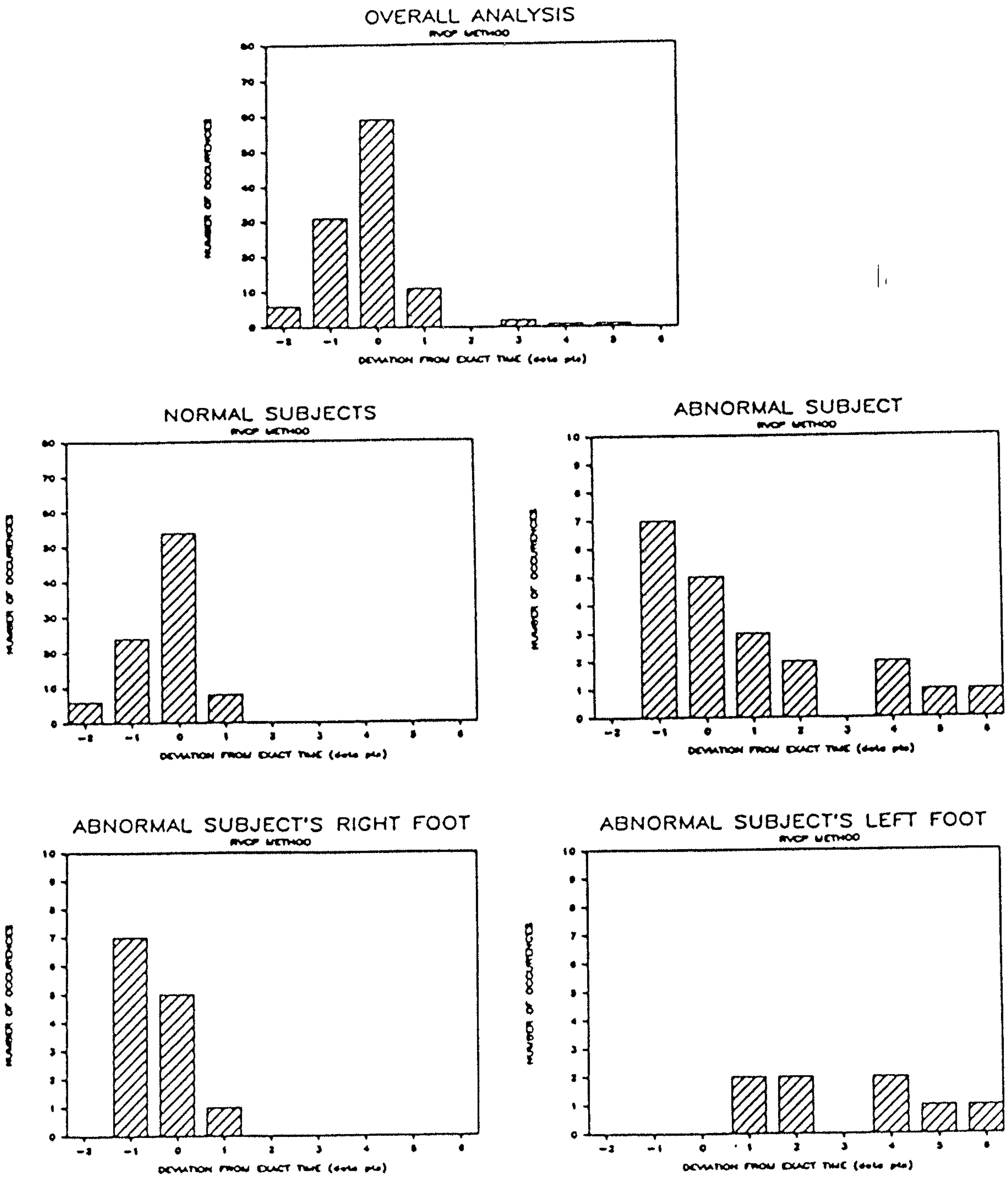



Figure 22: Results of RVCP Heel Strike Algorithm

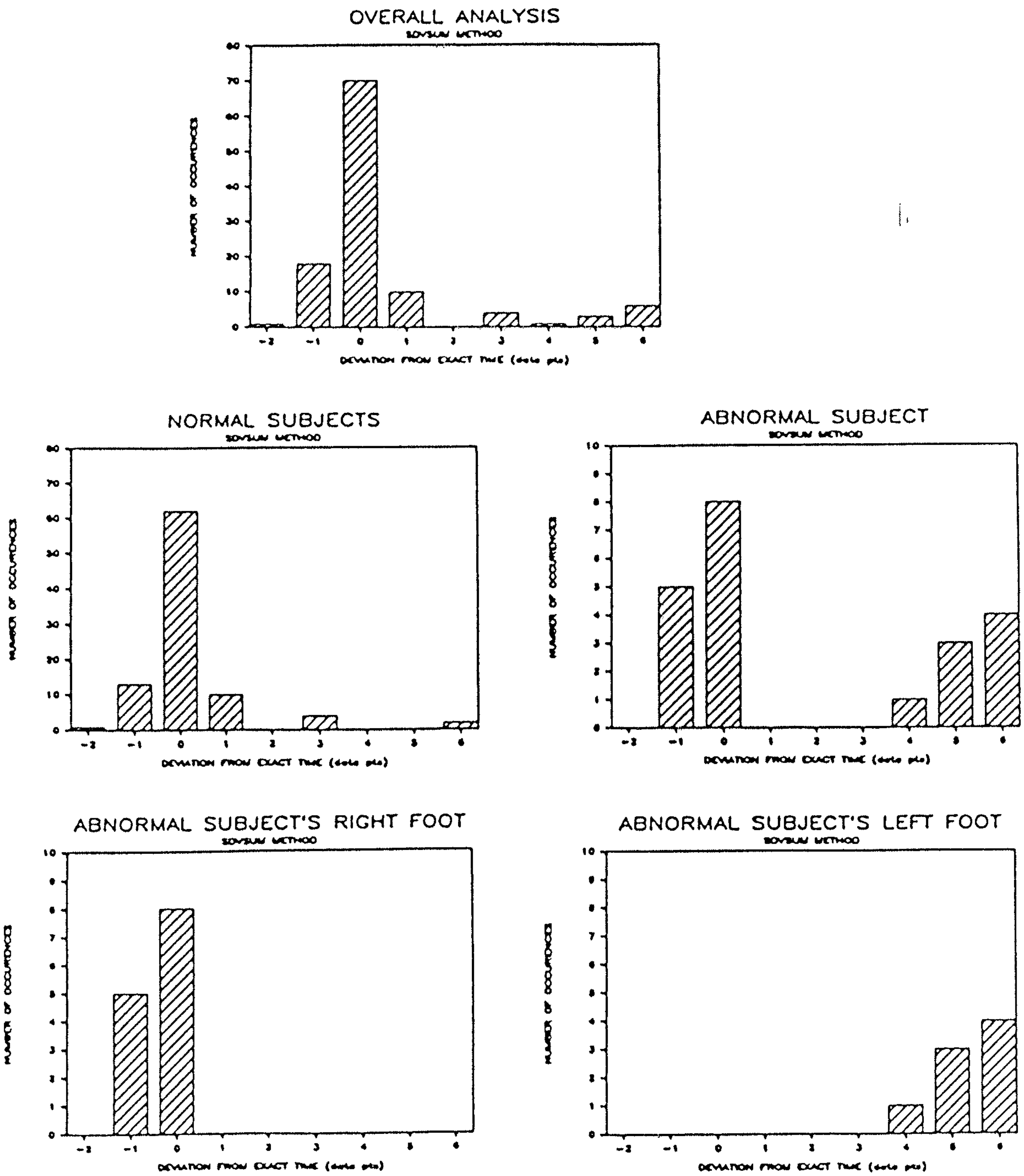
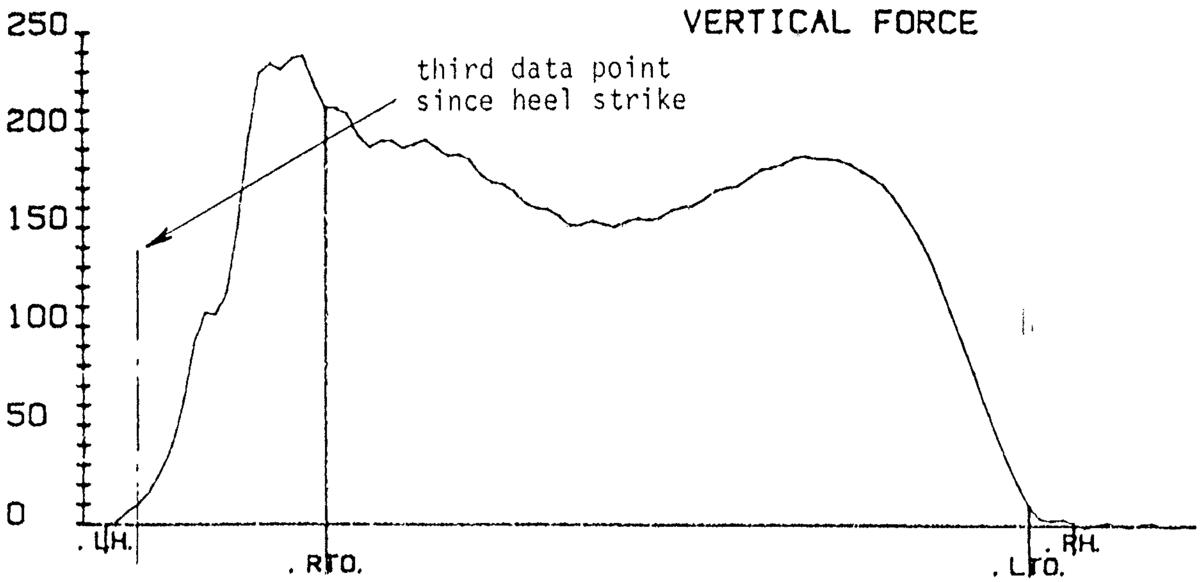
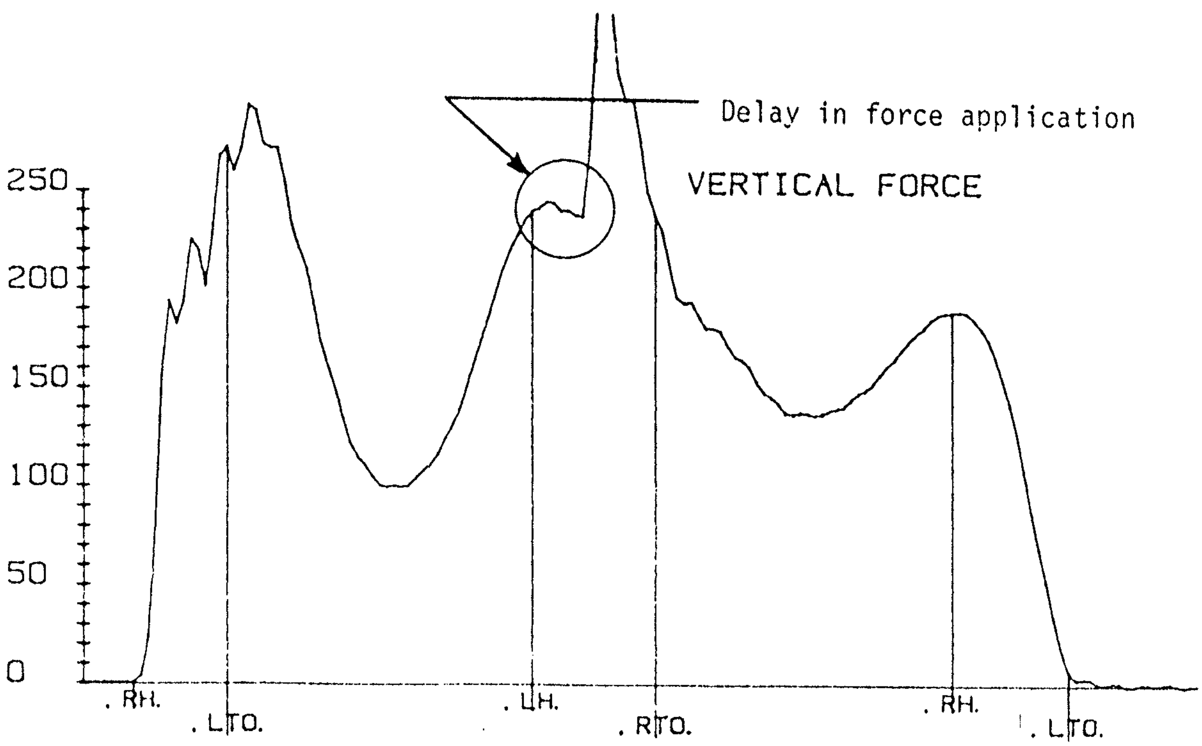



Figure 23: Results of SDVSUM Heel Strike Algorithm




(a) single left foot




(b) right-left sequence

Figure 24: Left Heel Strike of Subject With Abnormal Gait

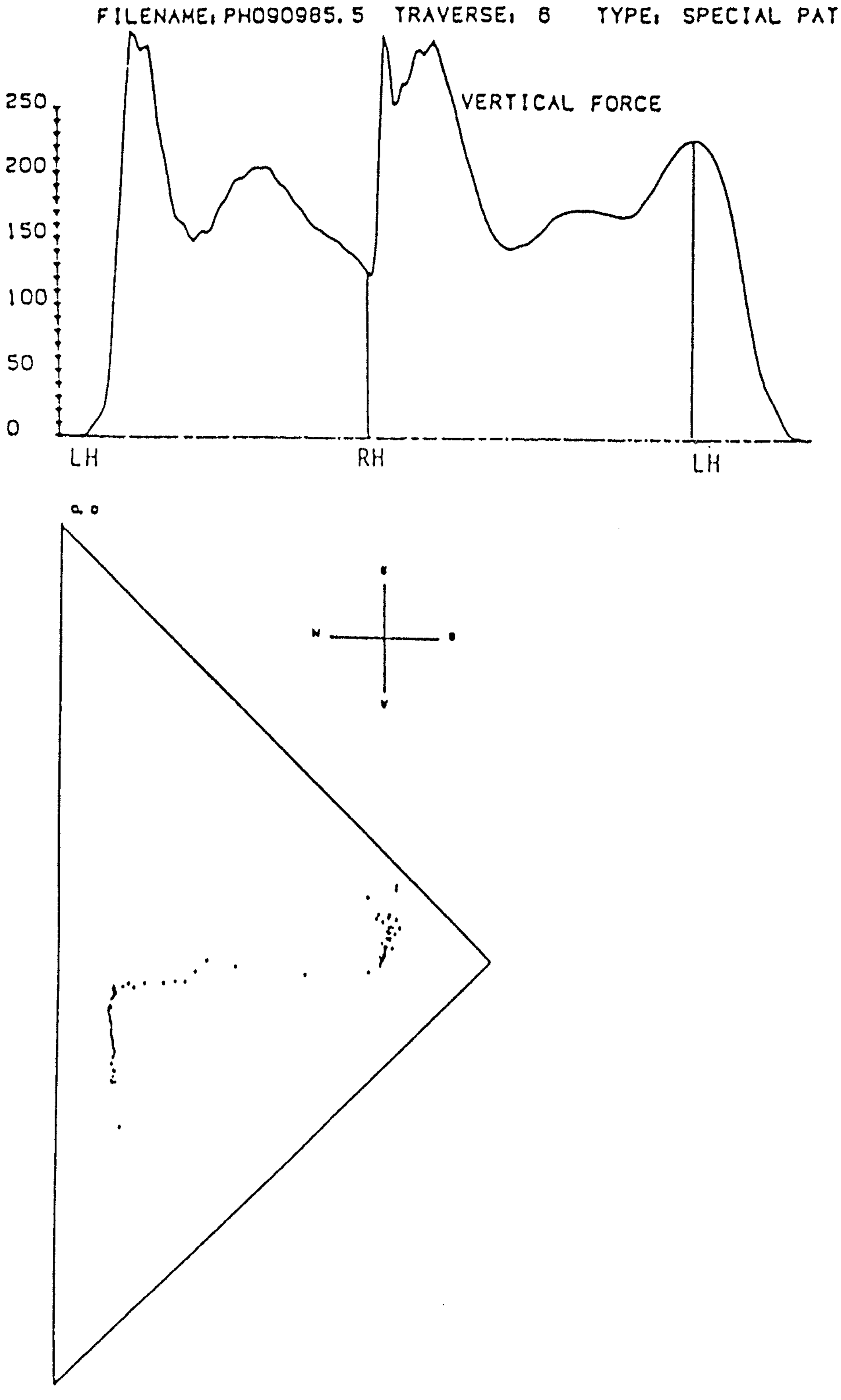



Figure 25: "Overlapping" Gait Pattern

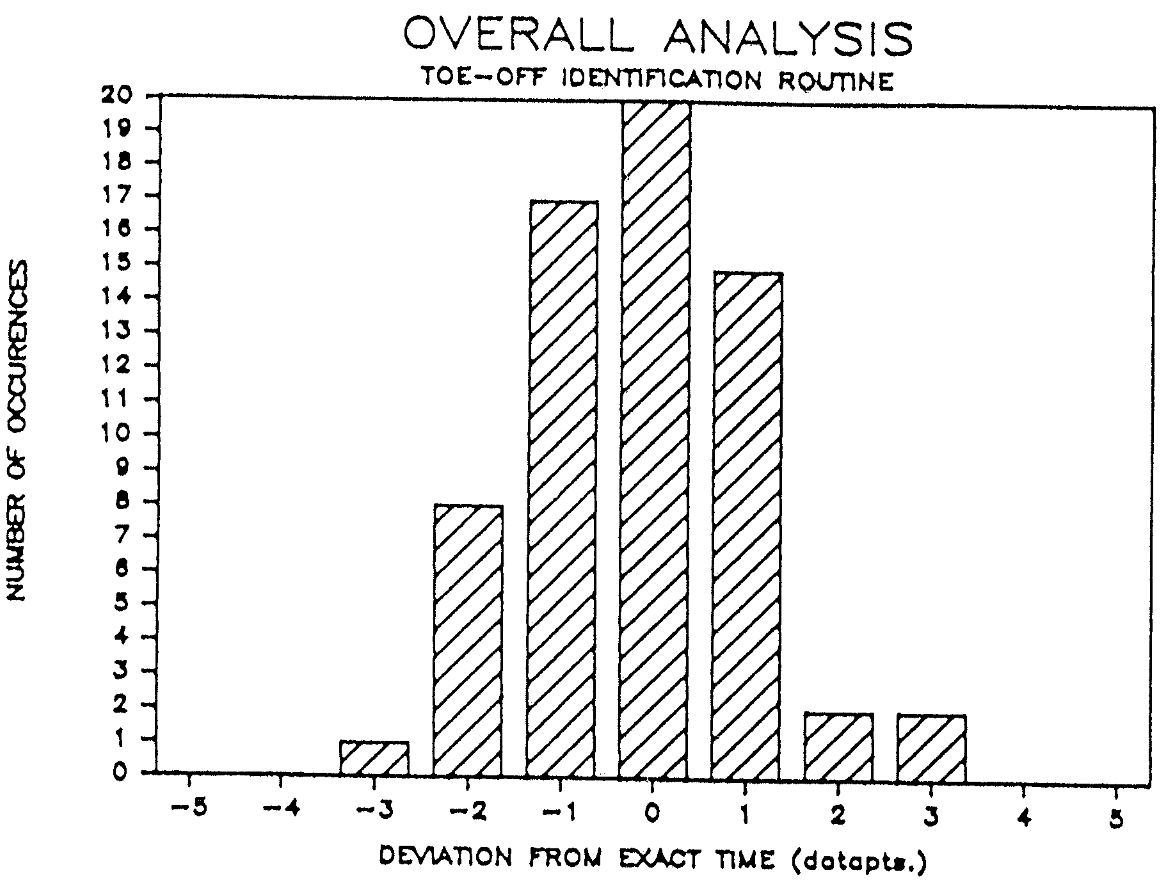
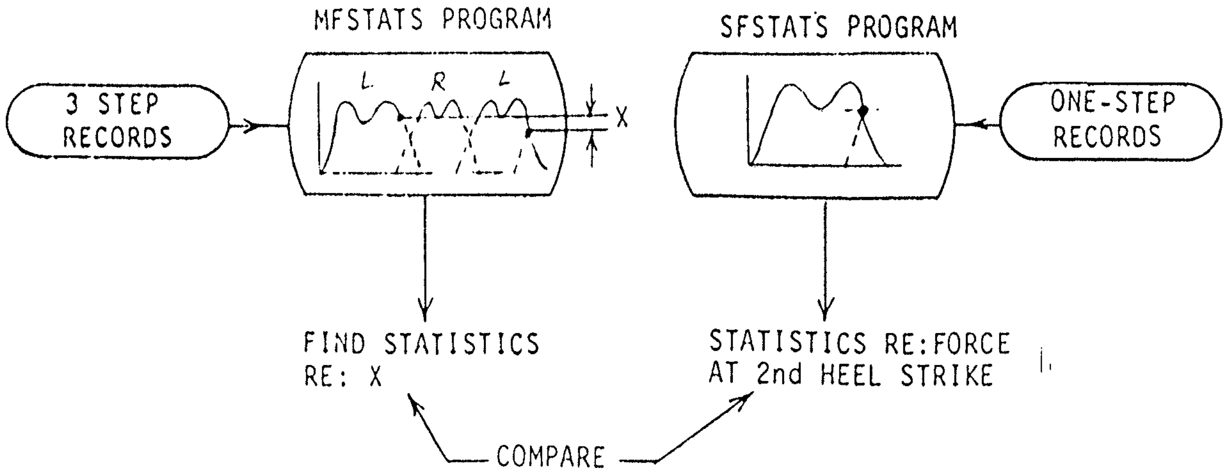
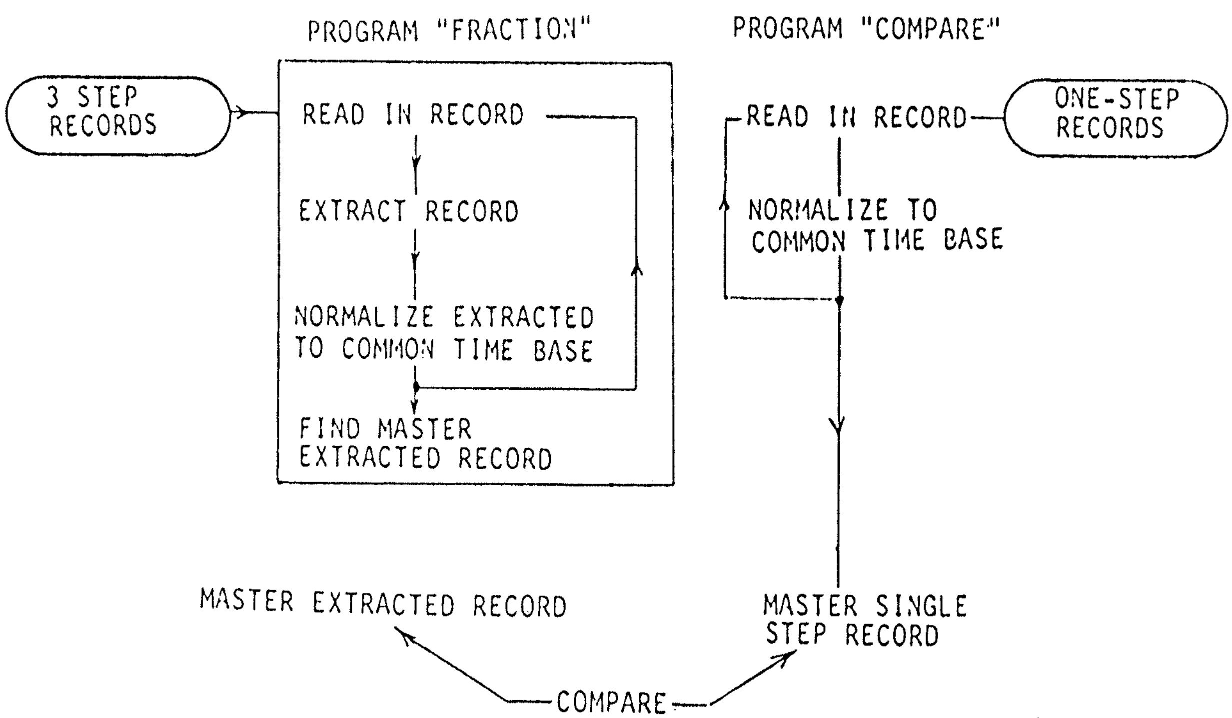



Figure 26: Toe-off Identification Results




(a) Analysis of Temporal Shift Algorithm




(b) Analysis of Fractional Split Algorithm

Figure 27: Software to Assess Force Extraction Algorithms

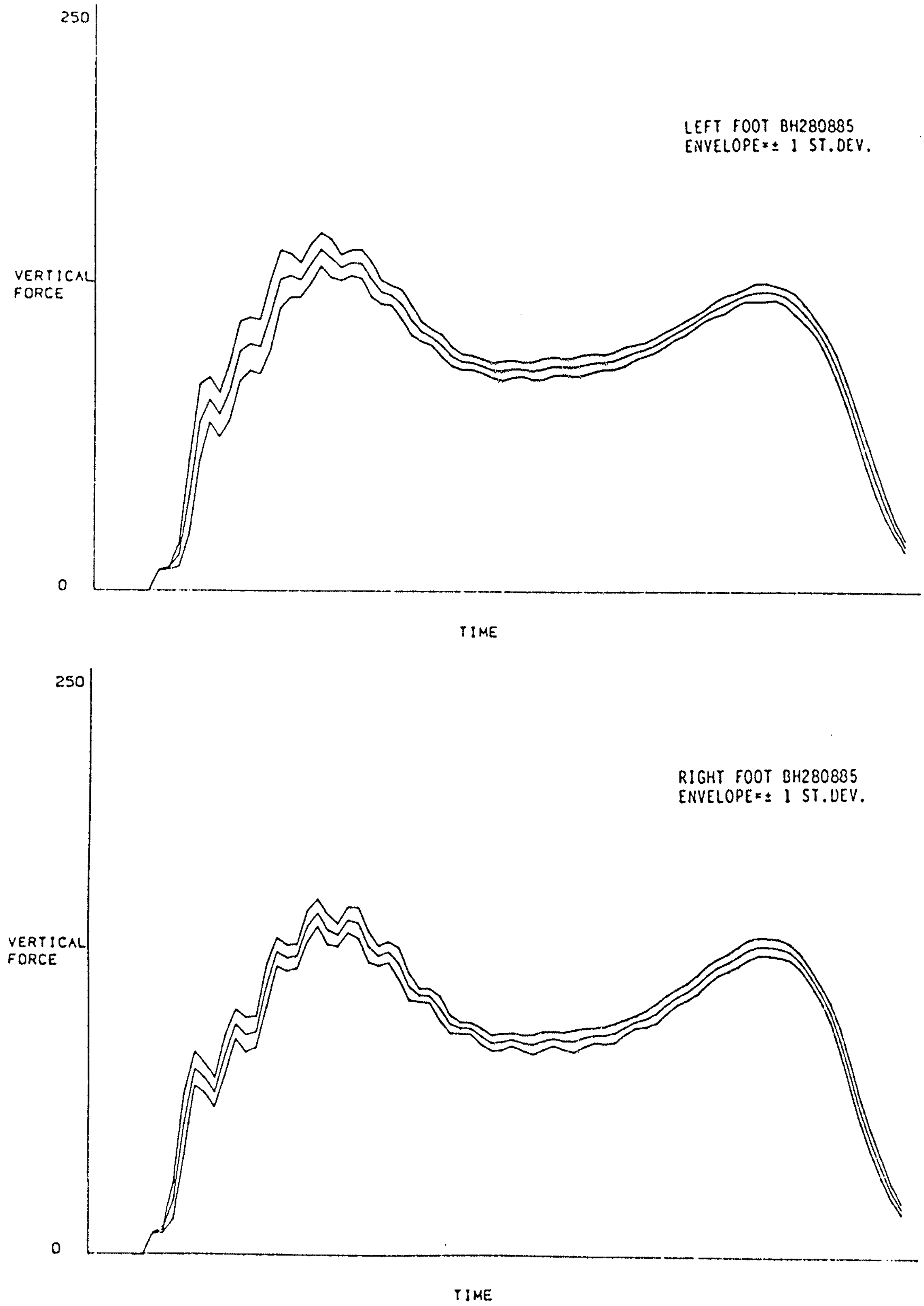
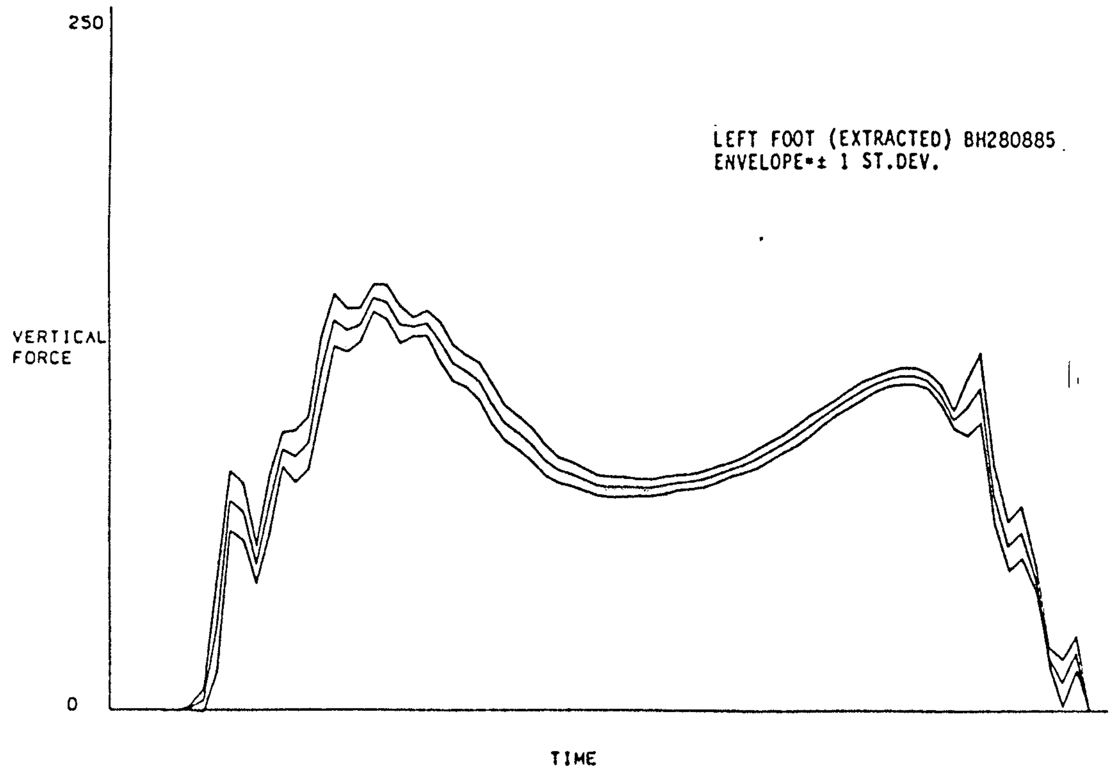



Figure 28: Master One-Step Records for Subject BH



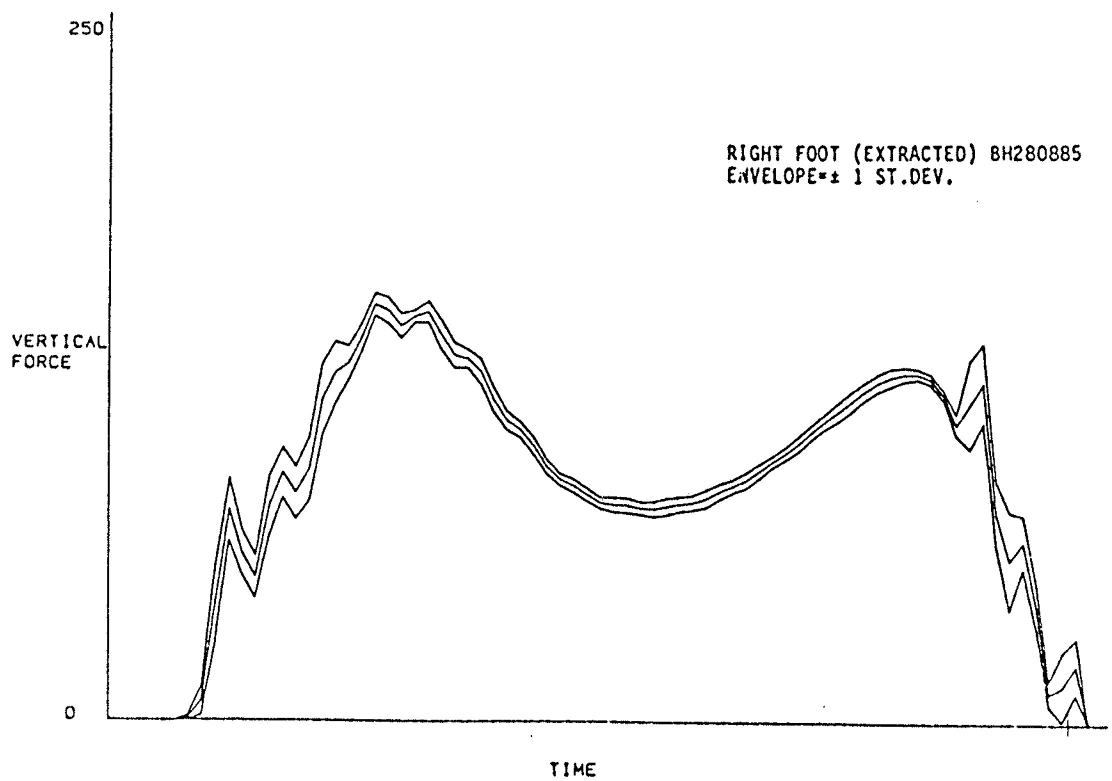




Figure 29: Master Extracted Records for Subject BH

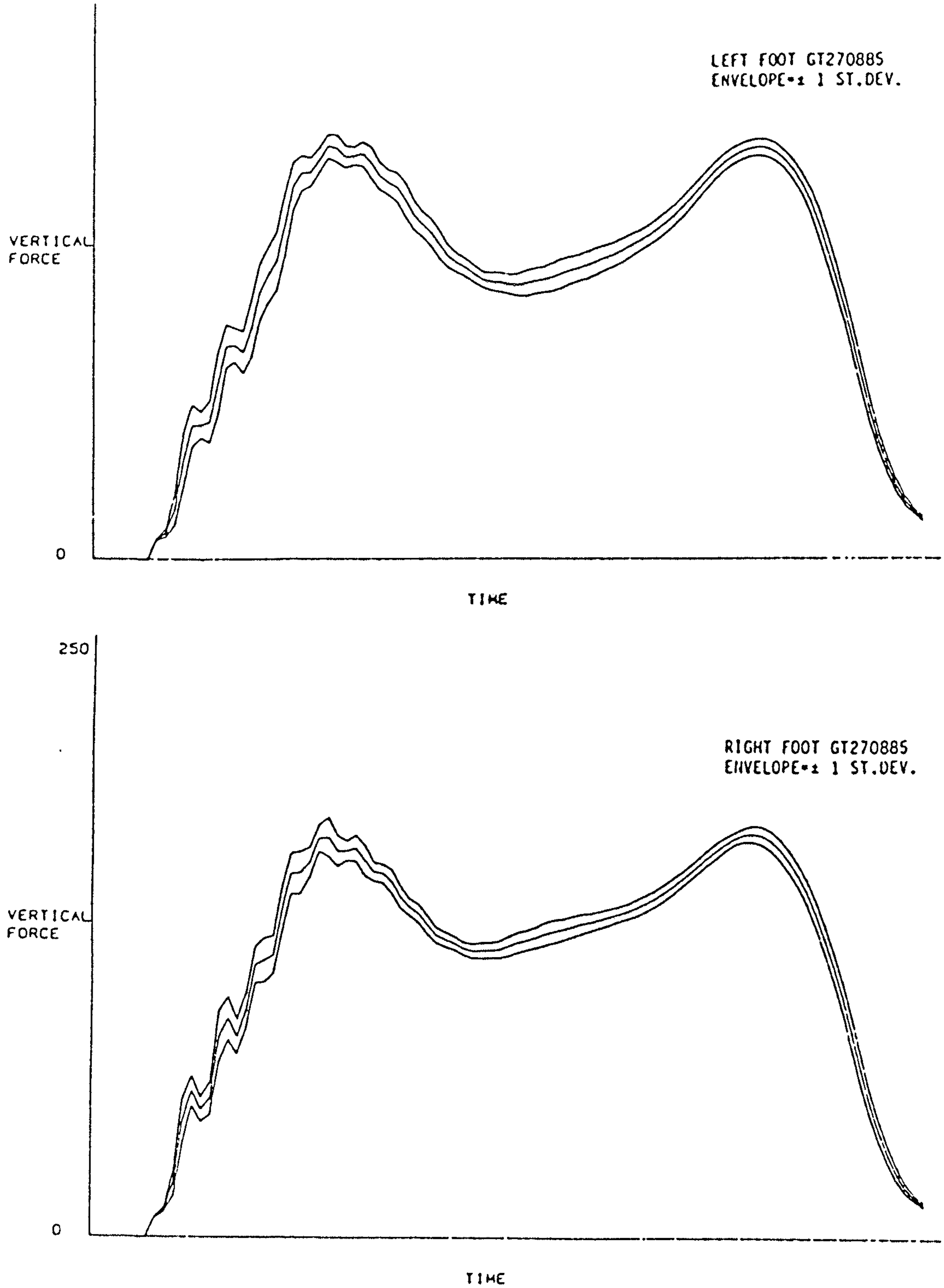
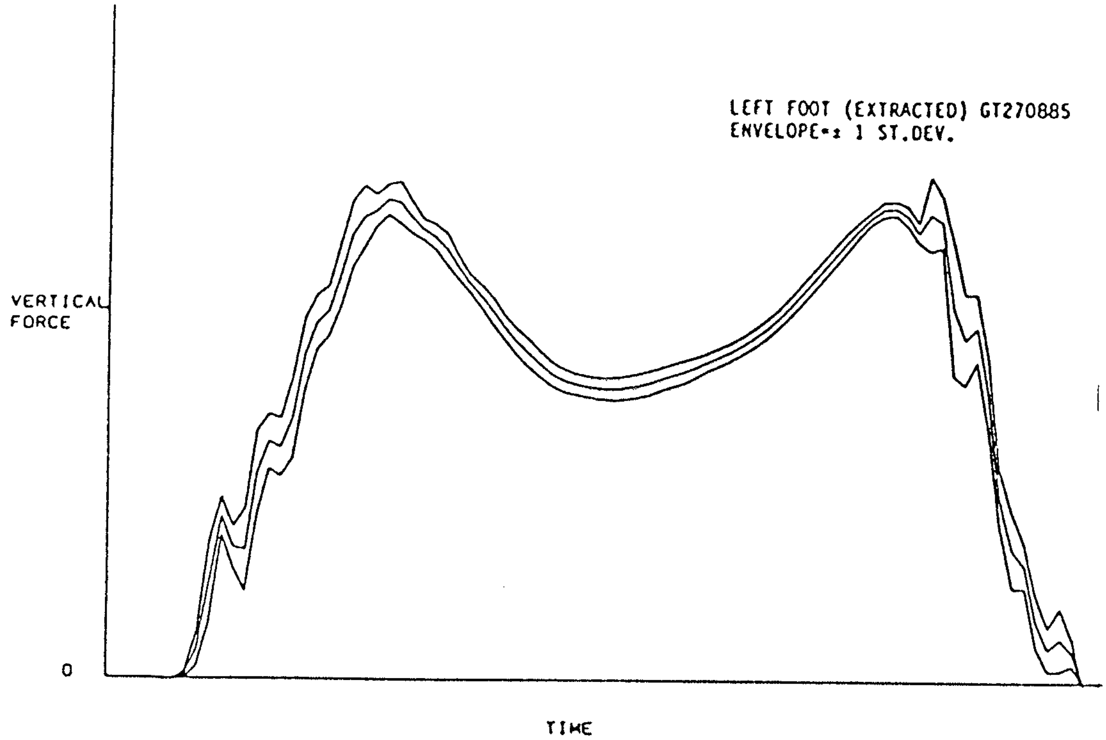



Figure 30: Master One-Step Records for Subject GT



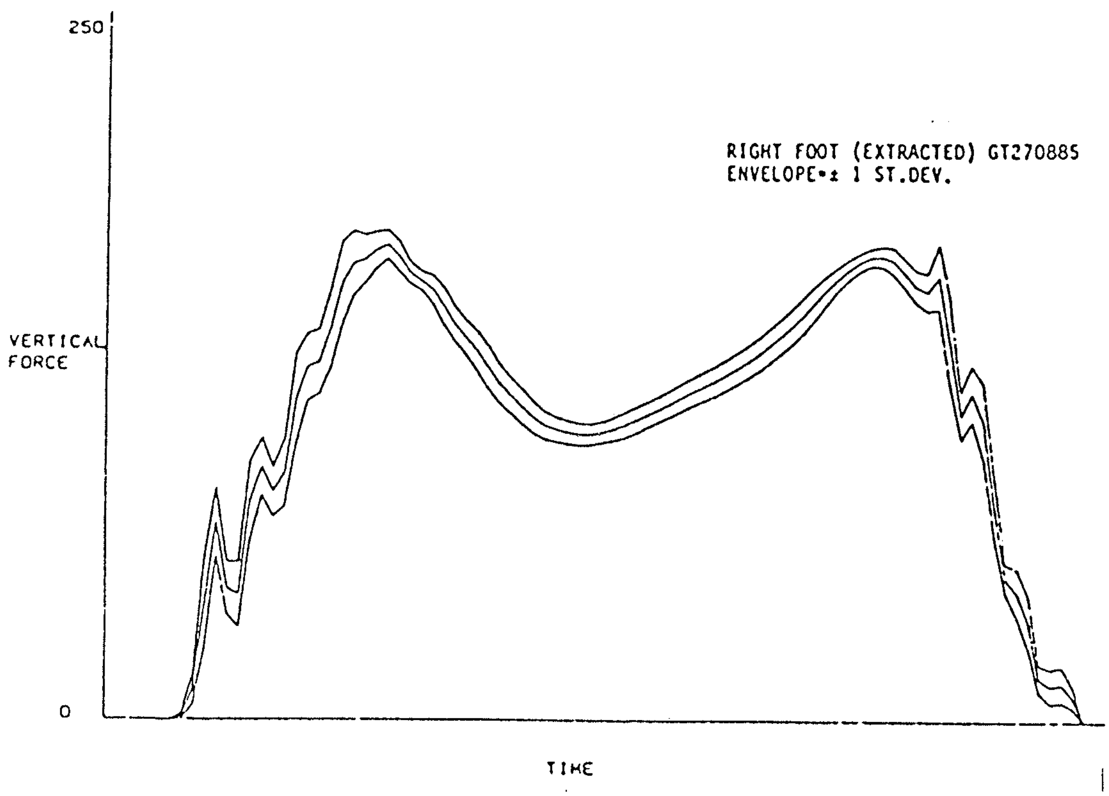




Figure 31: Master Extracted Records for Subject GT

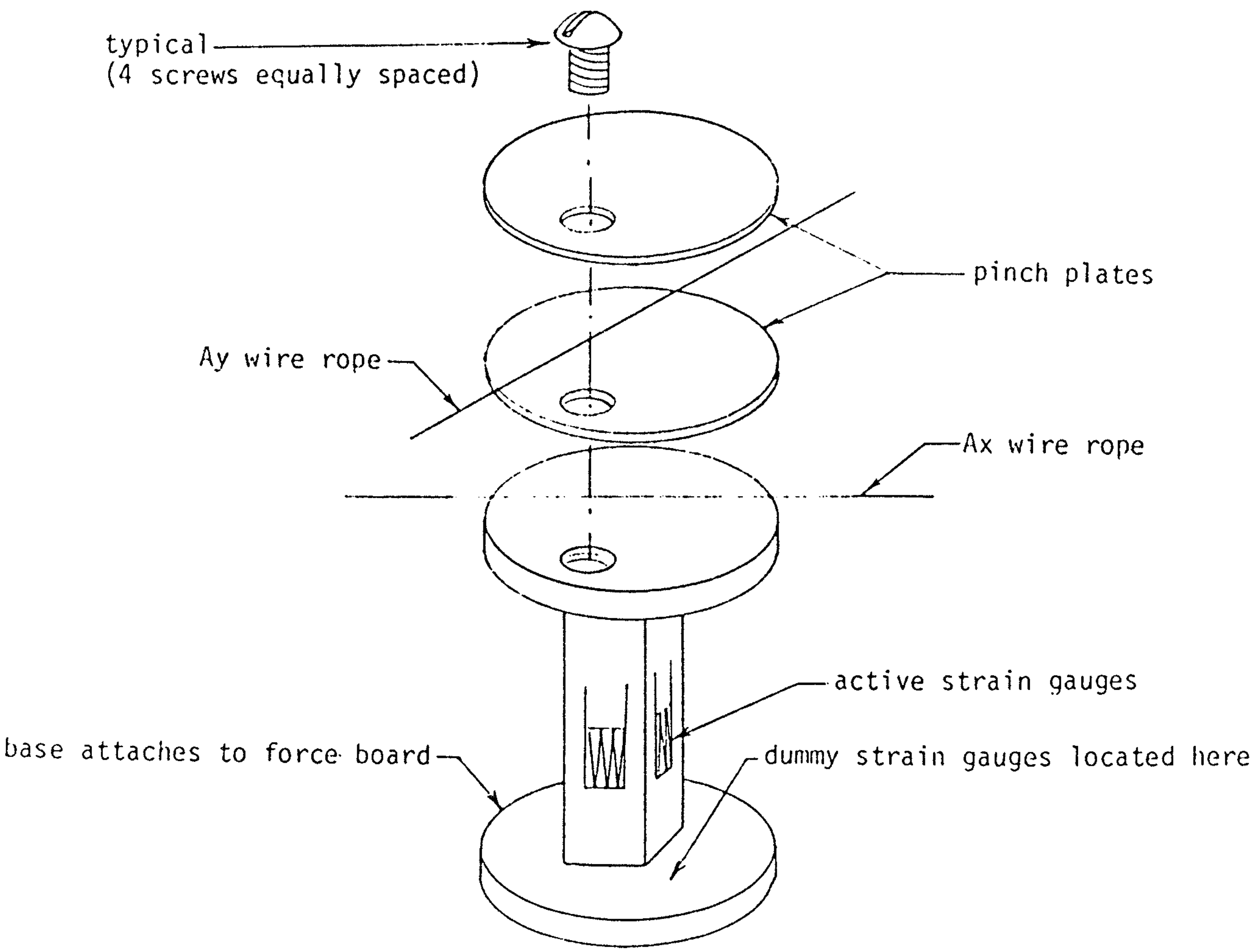



Figure 32: Modified Horizontal Support Transducer

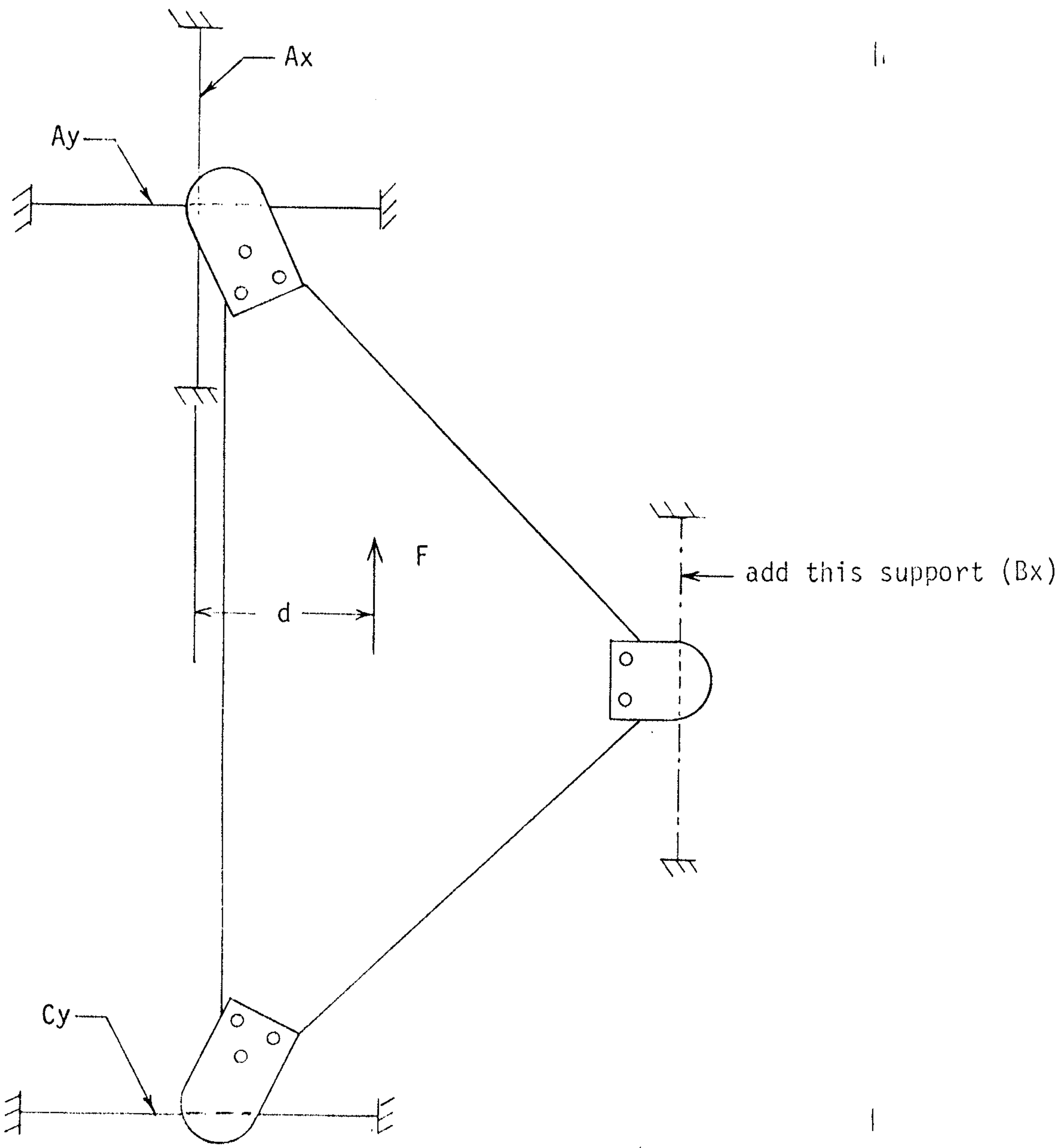



Figure 33: Modified Board Support

Test 1: -March 5/85

-unloaded board

| test | suspension | (natural resonant frequencies (hz.) | | |
|---|---|---|---|---|
| | | vert. | med.-lat. | pos.-ant. |
| 1(a) | normal | 30* | 23* | 17* |
| 1(b) | resting on cross-member | 38 | 23 | 17 |
| 1(c) | on posts to floor | 65 | - | - |

* mechanical and digital measurement of vibration

Test 2: -June 15/85

-100 pounds at board center

| test | suspension | natural resonant frequencies(hz.) | | |
|---|---|---|---|---|
| | | vert. | med.-lat. | pos.-ant. |
| 2 | normal | 25 | 17 | 13 |

Table 1: Results of Vibration Tests

FILENAME: rt280885.1 TRAVERSE NO: 2

| PT | VSUM | CPX | CPY |
|---|---|---|---|
| 1 | 24.06 | 23.58 | 17.98 |
| 2 | 24.66 | 23.42 | 17.95 |
| 3 | 26.00 | 23.59 | 18.01 |
| 4 | 26.92 | 23.73 | 17.96 |
| 5 | 27.73 | 23.59 | 17.98 |
| 6 | 28.33 | 23.45 | 17.96 |
| 7 | 29.25 | 23.57 | 17.91 |
| 8 | 30.48 | 23.63 | 17.86 |
| 9 | 31.49 | 23.52 | 17.92 |
| 10 | 32.62 | 23.64 | 17.93 |
| 11 | 33.74 | 23.47 | 17.94 |
| 12 | 34.97 | 23.66 | 18.03 |
| 13 | 35.89 | 23.76 | 17.99 |
| 14 | 36.70 | 23.65 | 18.01 |
| 15 | 37.72 | 23.56 | 18.06 |
| 16 | 38.65 | 23.53 | 18.15 |
| 17 | 39.56 | 23.38 | 18.12 |
| 18 | 41.08 | 23.26 | 18.18 |
| 19 | 42.82 | 23.61 | 18.04 |
| 20 | 43.85 | 23.75 | 18.08 |
| 21 | 44.46 | 23.65 | 18.06 |
| 22 | 45.38 | 23.63 | 18.14 |
| 23 | 46.40 | 23.65 | 18.07 |
| 24 | 46.50 | 23.50 | 18.14 |
| 25 | 47.60 | 23.18 | 18.14 |
| \| | \| | \| | \| |
| 103 | 70.35 | 23.43 | 18.12 |
| 104 | 70.67 | 23.75 | 18.03 |
| 105 | 70.13 | 23.57 | 17.95 |
| 106 | 68.53 | 23.68 | 18.08 |
| 107 | 66.90 | 23.57 | 18.14 |
| 108 | 65.77 | 23.51 | 18.14 |
| 109 | 65.87 | 23.48 | 18.12 |
| 110 | 67.07 | 23.51 | 17.94 |
| 111 | 67.70 | 23.66 | 18.00 |
| 112 | 67.68 | 23.31 | 18.08 |
| 113 | 68.63 | 23.50 | 18.20 |
| 114 | 69.86 | 23.73 | 18.10 |
| 115 | 70.15 | 23.57 | 18.09 |

| average CPX | measured CPX | average CPY | measured CPY |
|---|---|---|---|
| 23.58 | 24.25 | 18.05 | 18.5 |

Table 2: Numerical Data of a Typical Resolution Test

Base Data

| subject | filename | sex | weight | height | age |
|---|---|---|---|---|---|
| NA | NA240685.1 | M | 165 | 6'1" | 24 |
| KM | KM240685.1 | M | 165 | 5'10" | 26 |
| MM | MM240685.1 | M | 190 | 6' 1" | 25 |
| MN | MN240685.1 | M | 133 | 5'7" | 19 |
| AD | AD240685.1 | M | 195 | 5'6" | 25 |
| RM | several | M | 155 | 5'10" | 37 |
| DY | DY210685.1 | M | 133 | 5'7" | 19 |
| LO | LO210685.1 | F | 110 | 5'2" | 31 |

Test Data

| subject | filename | sex | weight | height | age |
|---|---|---|---|---|---|
| BH | BH280885.1 to .9 | F | 113 | 5'2" | 22 |
| GT | GT270885.1 to .9 | M | 150 | 5'8" | 23 |
| * PH | PH090985.1 to .9 | M | 200 | 5'9" | 25 |

* the subject has some hip deformity, and wears 3" built-up left shoe. An additional 2" build-up would be required to achieve level stance.

Table 3: Description of Subjects

| Multi-step statistics (re:inter-step gap "X") | | | | | Single-step statistics (re: force at 2nd heel strike | | |
|---|---|---|---|---|---|---|---|
| Subject | Type | mean"X" | std.dev."X" | | foot | std.dev. | $\sqrt{2}$*std.dev. |
| BH | RLR | 2.175 | 4.802 | | right | 3.581 | 5.064 |
| BH | LRL | 0.487 | 5.323 | | left | 5.902 | 8.347 |
| GT | RLR | -4.818 | 6.118 | | right | 8.128 | 11.495 |
| GT | LRL | 1.836 | 7.417 | | left | 11.890 | 16.815 |

Table 4: Statistics for Temporal Shift Algorithm

## REFERENCES


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

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

3. Beely, F., "Zur mechanik des stehens", Arch. f. klin. Chir. 27: pp.457-471, 1882

4. Fischer, O., "Der gang des menschen, I. Theil: Versuche am anbelasteten und belsteten menschen", Abhandlungen der mathematisch-Physischen classe der koniglich sashsischen, pp.153-322, Leipzig, 1885

5. Muybridge, E., "Complete human and animal locomotion -all 781 plates from the 1887 animal locomotion by E. Muybridge", Dover publications, New York, 1979

6. Borelli., De Motu Animalium, 1682.

7. Choa, E.Y. and Rim, K., "Application of optimization principles in determining the applied moments in human leg joints during gait", J. of Biomechanics, Vol. 6, pp. 479-510, 1973

8. Thornton-Trump, A. and Daher, R., "The prediction of reaction forces from gait data", J. of Biomechanics, Vol. 8, pp. 173-178, 1975

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

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

11. Splolek, G. and Lippert F., "An instrumented shoe- A portable force measuring device", J. of Biomechanics, Vol. 9, pp. 779-783, 1976

12. Winter, D.A., et al, "Analysis of instantaneous energy of normal gait", J. of Biomechanics, Vol. 9, pp. 253-257, 1976

13. Chao, E. et al , "Three-dimensional force analysis of finger joints in selected isometric hand functions", J.of Biomechanics, Vol. 9, pp. 387-396, 1976

14. Cappozzo , A. et al, "The interplay of muscular and external forces in human ambulation", J. of Biomechanics, Vol. 9,pp. 35-43, 1976

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

16. Onyshko, S. and Winter, D., "A mathematical model for the dynamics of human locomotion", J. of Biomechanics, Vol.13,pp. 361-368, 1980

17. Gola, M., "Mechanical design, constructional details and calibration of a new force plate", J. of Biomechanics, Vol. 13, pp. 113-128, 1980

18. Cavanagh, P. and Lafortune, M., "Ground reaction forces in distance running", J. of Biomechanics, Vol. 13, pp. 397-406, 1980

19. Winter, D. et al, "Kinematics of normal locomotion- a statistical study based on T.V. data", Vol. 7, pp. 479-486,1974

20. Townsend, M., "Dynamics and coordination of Torso motions in human locomotion", J. of Biomechanics, Vol. 14, pp. 727-738, 1981

21. Boccardi, S. et al, "Evaluation of muscular moments at the lower limb joints by an on-line processing of kinematic data and ground-reaction", J. of Biomechanics, Vol. 14, pp. 33-45, 1981

22. Ramey, M. and Yang, A., "A simulation procedure for human motion studies", J. of Biomechanics, Vol. 14, pp. 203-213, 1981

23. Cappozzo, A., "The forces and couples in the human trunk during level walking", J. of Biomechanics, Vol. 16, pp. 265-277,1982

24. Schneider, E., Chao, E., "Fourier analysis of ground reaction forces in normals and patients with knee disease", J. of Biomechanics, Vol. 16, pp. 591-601, 1983

25. Yamamoto, S. et al, "Quantitative gait evaluation of hip diseases using principal component analysis", J. of Biomechanics, Vol. 16, pp. 717-726, 1983

26. Takeshi S., "An estimation of center of gravity from force platform data", J. of Biomechanics, Vol. 17, pp. 53-60, 1984

27. Rohrle, H. et al, "Joint forces in the human pelvis-leg

skeleton", J. of Biomechanics, Vol. 17, pp. 409-424, 1984

28. Dickinson, J. et al, "The measurement of shock waves following heel strike while running", J. of Biomechanics, Vol. 18, pp. 415-421, 1985

29. Antonsson, E. et al, "The frequency content of gait", J. of Biomechanics, Vol. 18, pp. 39-48, 1985

30. Hoy, M., "Modulation of limb dynamics in the swing phase of locomotion", J. of Biomechanics, Vol. 18, pp. 49-60, 1985

31. Lanshammar, H., "A practical evaluation of differentiation techniques for human gait analysis", J. of Biomechanics, Vol. 15, pp. 99-105, 1982

32. Siegler, S., "Simulation of human gait with the aid of a simple mechanical model", J. of Biomechanics, Vol. 15, pp.415-425, 1982

33. Harris, G. et al, "A method for the display of balance center of pressure data", J. of Biomechanics, Vol. 15, pp. 741-745, 1982

34. Winter D. et al, "Moments of force and mechanical power in jogging", J. of Biomechanics, Vol. 16, pp. 91-102, 1983

35. Hiroshi A. et al, "A simple method of measuring the footsole pressure of normal subjects using Prescale pressure-detecting sheets", J. of Biomechanics, Vol. 16, pp. 157-165, 1983

36. Bates, B. et al, "An assessment of subject variability, subject-shoe interaction, and the evaluation of running shoes using ground reaction force data", J. of Biomechanics, Vol. 16, pp. 181-191, 1983

37. Chao, E. et al, "Normative data on knee joint motion and ground reaction forces in adult level walking", J. of Biomechanics, Vol. 16, pp. 219-233, 1983

38. Saunders J. et al, "The major determinants in normal and pathological gait", J. of Bone and Joint Surgery, Vol. 35A, No. 3, pp. 543-558, 1953

39. Murray, M. et al, "Walking patterns of normal men", J. of Bone and Joint Surgery, Vol. 46-A, pp. 335-360, 1964

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

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

42. Grundy, M. et al, "An investigation of the centers of pressure under the foot while walking", J. of Bone and Joint Surgery, Vol. 57-B, pp. 98-103, Feb., 1975

43. Murray, M. et al, "Center of gravity, center of pressure, and supportive forces during human activities", J. of Applied Physiology, Vol. 23, pp. 831-838, 1967

44. Eberhart, H. and Inman, V., "An evaluation of experimental procedures used in a fundamental study of human locomotion", Annals of New York Academy of Science (51), pp. 1213-1228, 1951

45. Taylor, C. and Blaschke, A., "A method for kinematic analysis of the shoulder, arm, and hand complex", Annals of New York Academy of Sciences" (51), pp. 1251-1265, 1951

46. Elftman, H., "The function of muscles in locomotion", American Journal of Physiology, 125, pp. 357-366, 1939 (b)

47. Elftman, H., "Forces and energy change in the leg during walking", American Journal of Physiology", 125, pp. 339-356, 1939(a)

48. Boccardi, S. et al, "A new procedure for the evaluation of normal and abnormal gait", American Journal of Physical Medicine, 56, pp. 163-182, 1977 (b)

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

50. Murray, M. 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

51. Frank, A., "An approach to the dynamic analysis and synthesis of biped locomotion machines", Medical and Biological Engineering & Computations, Vol. 8, pp. 465-476, 1970

52. Brugger, W. and Milner, M., "Computer aided tracking of body motions using a C.C.D.- image sensor", Med. and Biol. Engineering and Comp., 16, pp. 207-210, 1978

53. Cunningham, D., "Two devices for measuring the forces acting on the human body during walking", Proceedings of the Society for Experimental Stress Analysis, IX, 2, 75, 1952

54. Brodland, G., "Vertical reaction force reconstruction in human locomotion by developing a digital acquisition system and a heel-strike algorithm", M.Sc. Thesis, University of Manitoba, 1982

55. Balakrishnan, S., "The interpretation of medio-lateral

reaction forces in human locomotion using body-coordinate systems", Ph.D. Thesis, University of Manitoba, 1982

56. Quanbury, A. et al, "Biomechanical analysis of Legg Perthes braces", Proc. of the special conference of the Canadian Society for Biomechanics, pp. 74-75, 1980

57. advertisement for the "Oxford Dynamics Vicon force plate" taken from the J. of Biomechanics, Vol. 16

58. Beer\Johnson, (1977) Vector Mechanics for Engineers, Mcgraw-Hill, New York

59. Bauld, N., (1982) Mechanics of materials, Brooks/Cole Engineering Division, Monterey, California

60. Manning, R., "User's guide to the force board", unpublished report, 1985

61. Chi-Tsong Chen, One Dimensional Digital Signal Processing, M. Decker Inc., (1979), New York

62. Walpoles, R. and Myers,R.,Probability and Statistics for Engneers and Scientists, Macmillan Publishing, New York, 1978

Program Listings

```
    REM          CALROSS
    REM
    REM  This program is an edited version of programs written by Chris
    REM  Gadsby (COMPCAL and portions of COMPEDIT).Following completion
    REM  of this program, nine calibration factors in the array CALIB(I)
    REM  are written to a sequential file in drive b:.
    REM
    CLS
    FLAG%=0
    DIM CAL(384,5),CALIB(9),DCOFF(6),VERT(3)
    REC=1
    PRINT "CALIBRATION PROGRAM":PRINT "~~~~~~~~~~~~~~~~~~~~"
    INPUT "PLEASE ENTER DATE:",F$
    INPUT"VERT.CALIBRATION LOAD AT CENTROID (DEFAULTS TO 180 lbs.)";

    VERCAL:CFORCE=VERCAL/3:IF CFORCE=O THEN CFORCE=60
    INPUT"MED/LAT CALIBRATION LOAD AT CENTROID (DEFAULTS TO 40lbs)";

    MEDLAT:IF MEDLAT=0 THEN MEDLAT=40
    INPUT"POST/ANT CALIBRATION LOAD (DEFAULTS TO 20lbs.)";POSTANT:

    IF POSTANT=0 THEN POSTANT=20
    FOR C=0 TO 5
1   GOSUB 3
    FOR I=1 TO 64
    CHANNEL=1
    GOSUB 2
    CAL(I,C)=INT(X)
    CHANNEL=2
    GOSUB 2
    CAL(I+64,C)=INT(X)
    CHANNEL=3
    GOSUB 2
    CAL(I+128,C)=INT(X)
    CHANNEL=4
    GOSUB 2
    CAL(I+192,C)=INT(X)
    CHANNEL=5
    GOSUB 2
    CAL(I+256,C)=INT(X)
    CHANNEL=6
    GOSUB 2
    CAL(I+320,C)=INT(X)
    NEXT I
    IF FLAG = 1 THEN GOTO 4
    NEXT C
4   GOSUB 900
    IF FLAG=1 THEN C=CODE% ELSE GOTO 5
    GOTO 1
5   REM
    '**** ROUTINE TO CALCULATE CALIBRATION FACTORS
    OFF(I)=0:NEXT I OFF(I)=0:NEXT I'Zero DCOFF vector.
    'Skip first 10 pts. of calibration.
    '(mechanical vibration of system.)
    DCOFF(1)=DCOFF(1)+CAL(I,0)
```

```
DCOFF(2)=DCOFF(2)+CAL(I+64,0)
DCOFF(3)=DCOFF(3)+CAL(I+128,0)
DCOFF(4)=DCOFF(4)+CAL(I+192,0)
DCOFF(5)=DCOFF(5)+CAL(I+256,0)
DCOFF(6)=DCOFF(6)+CAL(I+320,0)
NEXT I
'The above loop gets the D.C. offsets from the unloaded plate data.
FOR I=1 TO 6:DCOFF(I)=DCOFF(I)/55:NEXT I 'AVERAGES VALUES
'The vector DCOFF now has the D.C. offsets of the plate at
'the time of calibration.  These D.C. offsets are used when
'the following calibrators are calculated.  NB. these D.C.
'offsets are not used on the force records. (the D.C. offsets
'at acquisition time are taken from the force records themselves
'NOT the calibration records.)
REMCode to determine vertical correction parameters.
ERT(I)=0:NEXT I ERT(I)=0:NEXT I'Zero vertical vector.
FOR I=10 TO 64
VERT(1)=VERT(1)+CAL(I,1)
VERT(2)=VERT(2)+CAL(I+64,1)
VERT(3)=VERT(3)+CAL(I+128,1)
NEXT I
FOR I=1 TO 3:VERT(I)=VERT(I)/55-DCOFF(I):NEXT I
'The vertical data is averaged, and then the D.C. offset is corrected.
FOR I=1 TO 3:CALIB(I)=VERT(I)/CFORCE:NEXT I
'Note that CFORCE is the force exerted by the calibration rig. on the
'centroid of the plate divided by THREE.
REMCode to determine M.L. calibrators.
LL1=0:MLL2=0     LL1=0:MLL2=0'Zero variables.
FOR I=10 TO 64
MLL1=MLL1+CAL(I+192,2)
MLL2=MLL2+CAL(I+256,2)
NEXT I
MLL1=MLL1/55:MLL2=MLL2/55        'AVERAGES VALUES
LR1=0:MLR2=0'Zero variables.
FOR I=10 TO 64
MLR1=MLR1+CAL(I+192,3)
MLR2=MLR2+CAL(I+256,3)
NEXT I
MLR1=MLR1/55:MLR2=MLR2/55        'AVERAGES VALUES
'Note the medio-laterals require different calibrators, depending on
'the sign of the applied force, also there are two M.L. transducers
'thus a total of four calibrators are required.
F(4):MLL2=MLL2-DCOFF(5) F(4):MLL2=MLL2-DCOFF(5)'Correct D.C. offset.
CALIB(4)=MLL1/(MEDLAT/2):CALIB(5)=MLL2/(MEDLAT/2)
CALIB(6)=MLR1/(MEDLAT/2):CALIB(7)=MLR2/(MEDLAT/2)
'The above two lines convert the plate data to lbsf assuming the
'plate was calibrated using the M/L calibration load (input)
REMCode to calculate P.A. calibrators.
AL=0'Clear vector.
'Grab data.
PAL=PAL+CAL(I+320,4)
NEXT I
PAL=PAL/55             'AVERAGES VALUES
PAR=0
FOR I=10 TO 64
```

```
    PAR=PAR+CAL(I+320,5)
    NEXT I
    PAR=PAR/55   'AVERAGES VALUES
    PAR=PAR-DCOFF(6):PAL=PAL-DCOFF(6)   'Correct for D.C. offset.
    CALIB(8)=PAL/POSTANT:CALIB(9)=PAR/POSTANT
    'The above line converts the calibration data to lbsf assuming that
    'the P/A calibrations were done using P/A cal. load (input)
    FOR I=1 TO 9:CALIB(I)=ABS(CALIB(I)):PRINT CALIB(I):NEXT I
    REM
    GOSUB 700
    STOP
2   REM subroutine for reading numbers off A-D board.
    CHANNEL=CHANNEL-1:ADDRESS=&H710
    OUT ADDRESS+4,128
    OUT ADDRESS+5,CHANNEL
    OUT ADDRESS+6,0
6   IF INP(ADDRESS+4)<128 THEN GOTO 6
    LOW = INP(ADDRESS+5)
    HIGH=INP(ADDRESS+6)
    X=256*HIGH + LOW
    IF X>32767 THEN X=X-65536!
    RETURN
3   REM MESSAGE OUT SUBROUTINE.
    IF C>0 THEN BEEP
    PRINT
    IF C=0 THEN PRINT "READY FOR D.C. OFFSET CALIBRATION" ELSE GOTO 10
    GOSUB 660
10 IF C=1 THEN PRINT "READY FOR VERTICAL CALIBRATION" ELSE GOTO 11
    GOSUB 660
11 IF C=2 THEN PRINT "READY FOR M.L LEFT CALIBRATION" ELSE GOTO 12
    GOSUB 660
12 IF C=3 THEN PRINT "READY FOR M.L RIGHT CALIBRATION" ELSE GOTO 13
    GOSUB 660
13 IF C=4 THEN PRINT "READY FOR P.A. LEFT CALIBRATION" ELSE GOTO 14
    GOSUB 660
14 IF C=5 THEN PRINT "READY FOR P.A. RIGHT CALIBRATION" ELSE GOTO 15
    GOSUB 660
15 RETURN
660  PRINT:PRINT "STRIKE [ENTER] WHEN READY.....";
     INPUT "",CRAP%
     PRINT "CALIBRATING."
     RETURN
700  REM SUBROUTINE FOR DUMPING CALIBRATION TABLE TO DISK.
     FCAL$=F$+" RCAL":FCAL$="B:"+FCAL$
     INPUT "PLEASE ENTER [CALIBRATION] DISK INTO DRIVE B.  STRIKE [ENTER] W
HEN READY
     ",CRAP%
     OPEN FCAL$ FOR OUTPUT AS #1
     FOR I=1 TO 9: WRITE#1,CALIB(I):NEXT I
     CLOSE#1
     RETURN
900  REM SUBROUTINE TO ALLOW FOR REDOING SECTIONS.
     FLAG=0
     PRINT "DO YOU WISH TO REDO ANY OF THE CALIBRATIONS [Y/N]:";
     INPUT REDO$
```

```
     IF REDO$="N" OR REDO$="n" THEN GOTO 999
     FLAG=1
     PRINT "CODES FOR REDO:"
     PRINT
     PRINT "    1)  D.C. OFFSET."
     PRINT "    2)  VERTICALS."
     PRINT "    3)  M.L. LEFT."
     PRINT "    4)  M.L. RIGHT."
     PRINT "    5)  P.A. LEFT."
     PRINT "    6)  P.A. RIGHT."
     PRINT
     INPUT "ENTER REDO CODE";CODE%
     CODE%=CODE%-1
999  RETURN
```

```
TITLE  GAIT DATA ACQUISITION (VERSION 05/83)
;
;
SUBTTLPROLOG AND PROGRAM DESCRIPTION
PAGE
COMMENT #
#
SUBTTL MACRO LIBRARY
INCLUDE CGMACRO.LIB;Brings in macro library.
;
SUBTTL SEGMENT SETUP
PAGE
;
S_SEGSEGMENT PARA STACK 'STACK'
DB 128 DUP('STACK   ');Allocate area.
TOPSEQU THIS WORD;Get top.
S_SEGENDS
;
MAXCHARSEQU 25;Maximum characters allowed on input.
;
TEXTPARMSSTRUC;Input structure driven by DOS interrupts.
BUFSIZEDB MAXCHARS;Limit of buffer size.
NAMESIZEDB ?;Byte containing string length.
TEXTINDB MAXCHARS DUP(?);Actual buffer space.
TEXTPARMSENDS
;
FILEPARMSSTRUC;Structure for file control.
DRIVEDB ?;Drive number.
FNAMEDB '  ';File name for disk (1st two letters.)
FDATEDB 6 DUP(?);Buffer for file date.
FNAMEEXTDB ?;Extension (file number).
DB '  ';Pad right with blanks.
NOWBLOCKDW ?;Current block number.
RECSIZEDW ?;Defined record size.
SYSDB 16 DUP (?);Buffer for DOS use.
CRNUMDB ?;Record relative to NOWBLOCK.
RRECNODB 4 DUP(?);Record no. relative to file start.
DB '$'
FILEPARMSENDS
;
DATA_STRUCSTRUC;Data structure for final data set.
T1DW 3072 DUP('1');First traverse data.
T2DW 3072 DUP('2');Second traverse data.
T3DW 3072 DUP('3')  ;Third traverse data.
T4DW 3072 DUP('4')  ;Fourth traverse data.
T5DW 3072 DUP('5');Fifth traverse data.
T6DW 3072 DUP('6')        ;Sixth traverse data.
DATA_STRUCENDS
;
TIMER_COUNTSTRUC;Structure for counter values.
;
;The counter is set to count in terms of BCD.
;The counter is set to decrement at 10000 hz until
;it is zero.  Thus for exactly one hertz the counter would
;be 9999hz.  Recall this counts down to and including zero.
```

```
;
DW 333h;Counter for 30hz.
DW 200h;Counter for 50hz.
DW 168h;Counter for 60hz.
DW 50h;Counter for 200hz.
DW 25h;Counter for 400hz.
DW 20h;Counter for 500hz.
DW 10h;Counter for 1000hz.
TIMER_COUNTENDS
;
D_SEG1SEGMENT PARA PUBLIC 'DATA'
;
FDATADATA_STRUC <>;Incorporate structure for final data.
INFODW 384 DUP(?)
;
;
PARMATEXTPARMS <>;Incorporate input STRUC (renamed).
PARMBFILEPARMS <>;Incorporate file STRUC (renamed).
PARMCTIMER_COUNT <>;Incorporate timer STRUC (renamed).
;
DOS_RETDW ?
DATEDB 6 DUP (?);Buffer for date.
PNAMEDB 25 DUP(?);Buffer for patient name.
TIMER_COUNTLDB ?
TIMER_COUNTHDB ?
CHANDB ?
THRESH_HOLDDW ?
OVER_THRESHHOLDS_MADEDB ?
UNDER_THRESHHOLDS_MADE  DB ?
SUM_7_8           DW 0
TRAVERSE_NUM  DB ?
TRAV_BASEDW ?
FLAGGERDB ?
;
;
MESSG1DB 22 DUP('*'),0Dh,0Ah
DB '*  GAIT ACQUISITION  *',0Ah,0Dh
DB '*  VERSION    05/83  *',0Ah,0Dh
DB 22 DUP('*'),0Dh,0Ah,0Ah,0Ah
DB '$';Text delimeter.
MESSG2DB 'Please enter date (ddmmyy)',0Ah,0Dh
DB 'No delimiters please.',0Ah,0Dh
DB '$';Text delimeter.
MESSG3DB "Please enter patient's surname:"
DB '$'
MESSG4DB 'Please enter patient disk in right hand drive,'
DB 0Ah,0Dh,'Strike [ENTER] when ready',0Ah,0Dh
DB '$'
MESSG5DB 'The data buffers filled, do you want to redo '
DB 'the traverse [Y/N]?','$'
MESSG6DB 'Do you wish to redo any of the traverses [Y/N]?'
DB '$'
MESSG7DB 0Ah,0Dh,'Which traverse [1-6]?'
DB '$'
MESSG9DB 'Do you wish to make another run? [Y/N]:'
```

```
DB '$'
MESSG10DB 'The data disk is full, ',0Ah,0Dh
DB 'please insert another formatted disk.',0Ah,0Dh
DB 'Strike [ENTER] when ready.'
DB '$'
MESSG11DB 'Available sampling rates'
DB ' (default is 120hz):',0Ah,0Dh,0Dh
DB '1)   30hz.',0Ah,0Dh
DB '2)   50hz.',0Ah,0Dh
DB '3)   60hz.',0Ah,0Dh
DB '4)  200hz.',0Ah,0Dh
DB '5)  400hz.',0Ah,0Dh
DB '6)  500hz.',0Ah,0Dh
DB '7) 1000hz.',0Ah,0Dh,0Ah
DB 'Input number (1-7) of desired rate, or strike '
DB '[ENTER] for default:'
DB '$'
MESSG15DB 'Traverse '
MESSNODB ?
DB '$'
MESSNOEDB ' finished.....',0Ah,0Dh
DB '$'
D_SEG1ENDS
;
SUBTTL MAINLINE
PAGE
;
C_SEG1SEGMENT PARA PUBLIC 'CODE'
PROGRAMPROC FAR;Far enable call from DOS.
ASSUME CS:C_SEG1,DS:D_SEG1,SS:S_SEG,ES:D_SEG1
;
;
COMMENT#
The following code lines establish entry linkage
from DOS.
At entry:
DS=Segment address of program segment prefix area.
At exit:
CS=Segment address of program segment prefix area.

#
start:PUSH DS;Set return segment addr to stack.
;
MOV AX,D_SEG1;Get segment address.
MOV DS,AX;Set up data segment one.
MOV ES,AX;Set up extra segment.
;
POP CX;Recall segment return addr.
MOV DOS_RET,CX;Save segment return.
;
CALL CLRSCRN;Clear the screen.
PRTTEXT MESSG1;Welcome message.
PRTTEXT MESSG2;Prompt for the date.
READKEYB;Get the user response.
MOVWORDS DATE;Transfer the date to the buffer.
```

```
LINEFEED 2
MOV THRESH_HOLD,10
;
;The threshhold is set at 50 millivolts.  The threshhold may be
;calculated by dividing the number (in this case 10) by 204.7.
;The result represents the threshhold in volts.
;
;
REENTRY:NOP;Label for another run.
CALL FIXNAME
;
CALL OPTION_SET
CALL CLEAN_BUFFER;Sets data buffer to zero.
CALL BOARDSET;Configure and start A-D board.
;
;
;*******************ANALOG TO DIGITAL MODULE FOLLOWS******************
MOV TRAVERSE_NUM,1;First traverse.
TRAVERSE_ENTRY:NOP;Entry for successive traverses.
MOV OVER_THRESHHOLDS_MADE,0;Clear counter.
MOV UNDER_THRESHHOLDS_MADE,0;Clear counter.
MOV BX,0;Clear base register.
MOV CH,0;Clear upper byte.
MOV CL,TRAVERSE_NUM;Get traverse count.
PUSH CX;Save counter.
OR CL,00110000B;Convert to ASCII.
MOV MESSNO,CL;Put in buffer.
POP CX;Regain counter.
FIND_BASE:NOP
ADD BX,6144;Move to next traverse buffer.
LOOP FIND_BASE;Loop until CX zero.
SUB BX,6144;Allow for first traverse.
PRTTEXT MESSG15;Traverse message and number.
MOV TRAV_BASE,BX;Save traverse base.
MOV DX,CHANNUM;Get channel number port.
MOV AL,0;Choose the first channel.
OUT DX,AL;Set board.
MOV CHAN,0;Zero counter.
MOV DI,0;Offset into traverse one.
MOV DX,CTRL_9513;Get control port.
MOV AL,01110000B;Select counter five.
OUT DX,AL;Load and arm.
;Counter is now running.
;BX has traverse offset.
NEXT_SWEEP1:NOP
MOV SUM_7_8,0
NOT_READY1:MOV DX,STATUS_PRT;Get status port.
IN AL,DX;Get status byte.
TEST AL,10000000B;Test for done.
JNS NOT_READY1;Conversion not complete.
;
MOV DX,LOW_BYTE;Get low byte port.
IN AL,DX;Read low data byte.
XCHG AL,AH;Exchange to save.
MOV DX,HIGH_BYTE;Get high byte port.
```

```
IN AL,DX;Get high data byte.
XCHG AL,AH;Reswap bytes.
MOV [BX+DI],AX;Save data
INC CHAN;Inc channel counter to second chan.
;Data from channel one is now stored.
MOV DX,CONTROL_PRT;Get control port.
MOV AL,10000000B;Control byte.
OUT DX,AL;Disable external starts.
;and auto-inc.
NEXT_CHAN1:NOP
MOV DX,CHANNUM;Get channel port.
MOV AL,CHAN;Get channel number.
OUT DX,AL;Set channel.
;
MOV DX,CONVERT_START
OUT DX,AL;Start a conversion.
MOV DX,STATUS_PRT;Get status port.
NOT_READY2:IN AL,DX;Read board status.
TEST AL,10000000B;Test for done.
JNS NOT_READY2;Board not finished.
;
MOV DX,LOW_BYTE;Get low byte data.
IN AL,DX;Get byte.
XCHG AL,AH;Swap bytes.
MOV DX,HIGH_BYTE;Get high byte data port.
IN AL,DX;Get byte.
XCHG AL,AH;Reswap.
;
ADD BX,768;Increment to next channel.
MOV [BX+DI],AX;Stuff data in buffer.
                CMP CHAN,6
                JL  ZZZ                 ;SKIP FOLLOWING IF CH>3
                ADD SUM_7_8,AX          ;SOFTWARE ADDITION. AX HAS CURRENT
DATA
ZZZ:            INC CHAN
CMP CHAN,7;Swept too far?
JLE NEXT_CHAN1;No, read again.
MOV CHAN,0;Clear channel counter.
PUSH AX
;
MOV DX,CHANNUM;Get software channel port.
MOV AL,0;Choose first channel
OUT DX,AL;Set TECMAR.
;
MOV BX,TRAV_BASE;Refresh traverse base pointer.
MOV DX,CONTROL_PRT;Get control port.
MOV AL,10000100B;Enable external starts.
OUT DX,AL;Send to board.
ADD DI,2;Increment destination index.
POP AX;Regain data.
;
;AX now has the 8th channel's data in it.
;Recall this is the threshhold channel.
;
PUSH BX
```

```
MOV BX,400
CMP BX,SUM_7_8
POP BX
JLE MAYBE_GO;Yes, check for noise.
MOV OVER_THRESHHOLDS_MADE,0 ;Otherwise zero counter.
CMP DI,30;Enough preview points? (15)
JLE SEND1;No go back and sweep again.
MOV DI,0;Otherwise start to overwrite.
SEND1:JMP NEXT_SWEEP1;Back until thresh passed.
;
MAYBE_GO:NOP;Save last SUM_7_8 for comparison
INC OVER_THRESHHOLDS_MADE;One more thresh.
CMP OVER_THRESHHOLDS_MADE,5 ;At least five consecutive
;points are needed to ensure
;protection against noise.
JE THRESH_TEST_PASSED
JMP FAR PTR NEXT_SWEEP1
THRESH_TEST_PASSED:NOP;Label for traverse start.
;
;At this point the system is in full operation
;the same type of operations must still be done, but
;now the computer must look for the end of the current
;traverse, and still prevent an overflow.
;
;****************** FIND NEW THRESHOLD ******************************
;
PUSH AX
MOV SI,0
MOV AX,[SI]
MOV SI,2
ADD AX,[SI]
MOV SI,4
ADD AX,[SI]
MOV SI,6
ADD AX,[SI]
MOV SI,8
ADD AX,[SI]
MOV SI,768
ADD AX,[SI]
MOV SI,770
ADD AX,[SI]
MOV SI,772
ADD AX,[SI]
MOV SI,774
ADD AX,[SI]
MOV SI,776
ADD AX,[SI]
MOV SI,1536
ADD AX,[SI]
MOV SI,1538
ADD AX,[SI]
MOV SI,1540
ADD AX,[SI]
MOV SI,1542
ADD AX,[SI]
```

```
MOV SI,1544
ADD AX,[SI]
PUSH BX
PUSH DX
MOV DX,0
MOV BX,5
DIV BX
ADD AX,10;Ten is the old threshold value
MOV THRESH_HOLD,AX
POP DX
POP BX
POP AX
;
;*******************START TESTS PASSED*******************************
;
NEXT_SWEEP2:NOP
NOT_READY3:MOV DX,STATUS_PRT
IN AL,DX;Get status byte.
TEST AL,10000000B;Test for done.
JNS NOT_READY3;Return until ready.
;
MOV DX,LOW_BYTE;Get low byte port.
IN AL,DX;Read low data port.
XCHG AL,AH;Exchange to save.
MOV DX,HIGH_BYTE;Get high byte port.
IN AL,DX;Read high data port.
XCHG AL,AH;Reswap.
MOV [BX+DI],AX;Store data.
                MOV SUM_7_8,AX                ;LOAD CH1 INTO SUM_7_8
INC CHAN;Increment channel counter.
;
;Data from channel one is now stored.
;
MOV DX,CONTROL_PRT;Get TECMAR channel port.
MOV AL,10000000B;Disable external starts.
OUT DX,AL;Send to device.
;
NEXT_CHAN2:NOP
MOV DX,CHANNUM;Get channel-setting port.
MOV AL,CHAN;Get channel number.
OUT DX,AL;Set board.
;
MOV DX,CONVERT_START;Get software start port.
OUT DX,AL;Initialize conversion.
MOV DX,STATUS_PRT;Get status port.
NOT_READY4:IN AL,DX;Read status byte.
TEST AL,10000000B;Test for done.
JNS NOT_READY4;Wait until finished.
;
MOV DX,LOW_BYTE;Get low byte port.
IN AL,DX;Read low data byte.
XCHG AL,AH;Exchange to save.
MOV DX,HIGH_BYTE;Get high byte port.
IN AL,DX;Read high data byte.
XCHG AL,AH;Reswap.
```

```
;
                CMP CHAN,2                  ; 2 MEANS CH.3
                JG NEW_LABEL                ;SKIP FOLLOWING FOR CH>3
                ADD SUM_7_8,AX                 ;ADD 2nd OR 3rd CH TO SUM_7_8
NEW_LABEL:ADD BX, 768;Increment base to next ch.
MOV [BX+DI],AX;Move to buffer.
INC CHAN;Increment channel counter.
CMP CHAN,7;Swept too far?
JLE NEXT_CHAN2;No, read again.
MOV CHAN,0;Reset channel counter.
;
PUSH AX
MOV DX,CHANNUM;Get software channel start.
MOV AL,0;Reset channel to 0 (one).
OUT DX,AL;Reset board.
;
MOV BX,TRAV_BASE;Reset base pointer to trav.
MOV DX,CONTROL_PRT;Get control port.
MOV AL,10000100B;Enable external starts.
OUT DX,AL;Send to TECMAR.
POP AX;Regain data from stack.
;
;AX now has the eight channel's data in it (threshhold channel).
;
ADD DI,2;Increment destination index.
CMP DI,400
JLE YYY
PUSH BX
MOV BX,THRESH_HOLD
CMP BX,SUM_7_8              ;Data under threshhold.
                MOV SUM_7_8,0                  ; CLEAR SUM_7_8 FOR NEXT ROUND
POP BX
JGE MAYBE_FINISHED;Yes check for noise.
MOV UNDER_THRESHHOLDS_MADE,0;Reset done counter.
YYY:CMP DI,766;End of buffer?
JLE SEND2;No, go back and sweep again.
JMP DATA_OVERFLOW       ;Otherwise force end of traverse
MAYBE_FINISHED:NOP
INC UNDER_THRESHHOLDS_MADE     ;Inc end of trav counter
CMP UNDER_THRESHHOLDS_MADE,15  ;15 consecutive points.
     ;under threshhold needed
JE TRAVERSE_DONE;DONE!
SEND2:JMP FAR PTR NEXT_SWEEP2;Otherwise go back again.
DATA_OVERFLOW:NOP
CRETURN
PRTTEXT MESSG5;Inform user of overflow.
READKEYB;Redo traverse?
CRETURN
LINEFEED 1
MOV AH,'N';Get comparator.
CMP AH,PARMA.TEXTIN;Redo?
JE KEEP_DATA;No, keep data.
MOV AH,'n';Get comparator.
CMP AH,PARMA.TEXTIN;Redo?
JE KEEP_DATA;No, keep data.
```

```
;
;Before taking fresh data the traverse buffer must be zeroed.
;
MOV CH,0;Clear high byte of counter.
MOV CL,TRAVERSE_NUM;Get current traverse number.
MOV BX,0;Clear base register.
FIND_BASE2:NOP
ADD BX,6144;Inc to next traverse.
LOOP FIND_BASE2;Loop until CL zero.
SUB BX,6144;Correct for loop.
MOV AX,0;Clear accumulator.
MOV CX,3072;Words/traverse.
WIPE_TRAV:NOP
MOV [BX],AX;Clear word.
ADD BX,2;Inc to next word.
LOOP WIPE_TRAV;Clear traverse.
JMP FAR PTR TRAVERSE_ENTRY;Go back for more.
KEEP_DATA:NOP
TRAVERSE_DONE:NOP
PRTTEXT MESSNOE;Adds 'finished' to traverse number.
;Following lines of code store the number of data points taken.
RCR DI,1;Divide DI by two.
MOV BX,OFFSET INFO;Get base.
MOV CH,0;Clear upper byte.
MOV CL,TRAVERSE_NUM;Get counter.
SAVE_PLACE:NOP
ADD BX,2;Increment base.
LOOP SAVE_PLACE;Get offset.
SUB BX,2;Correct for loop.
MOV [BX],DI;Save counter.
;Counter saved.
CMP FLAGGER,1;Redo traverse?
JE RUN_DONE;Yes jump to end.
INC TRAVERSE_NUM;Increment the trav num. count
CMP TRAVERSE_NUM,7
MOV AH,0Ch
MOV AL,01h;text two lines forces system to wait
INT 21H;until keypressed.
JL SEND_FOR_ANOTHER_TRAVERSE;Go back.
JMP RUN_DONE
SEND_FOR_ANOTHER_TRAVERSE:JMP FAR PTR TRAVERSE_ENTRY
RUN_DONE:NOP
MOV FLAGGER,0;Clear special flag.
CALL REDO_ANY_TRAVERSES;User desires redo?
CMP FLAGGER,1;Redo traverse?
JE SEND_FOR_ANOTHER_TRAVERSE;Send back.
;
CALL DATAWRITE;Write data.
;
ENQUIRE:NOP;Following checks for end of program.
LINEFEED 2
PRTTEXT MESSG9;Any more runs to be made?
READKEYB;Get reply.
CRETURN
LINEFEED 2
```

```
MOV AH,'Y';Get comparator.
CMP AH,PARMA.TEXTIN;More runs to be made?
JE SEND_BACK;Yes-set up for sendback.
MOV AH,'y';Get comparator.
CMP AH,PARMA.TEXTIN;More runs to be made?
JE SEND_BACK;Yes-set up for sendback.
MOV AH,'N';Get comparator.
CMP AH,PARMA.TEXTIN;Finished?
JE END_PROG;Yes! Go to DOS.
MOV AH,'n';Get comparator.
CMP AH,PARMA.TEXTIN;Finished?
JE END_PROG;Yes! Go to DOS.
BEEP;INVALID INPUTS.
JMP ENQUIRE;Try again.
SEND_BACK:JMP FAR PTR REENTRY;Send back for another run.
;
;
;
END_PROG:NOP;Destination when program finished.
COMMENT#
The next lines of code set up the stack for the far
return to DOS.
#
MOV CX,DOS_RET
PUSH CX;Put segment return on stack.
SUB AX,AX;Clear accumulator.
PUSH AX;Push on new IP (INT 20h).
RET;Far return to DOS.
PROGRAMENDP
INCLUDE SUB1.ASM
INCLUDE SUB2.ASM
C_SEG1 ENDS
ENDstart
```

```
CLRSCRNPROC NEAR;Clears and locates cursor.
;
MOV CX,0;Upper left (CH,CL)=0,0
MOV DX,2479h;Lower right (DH,DL)=24,79
MOV BH,7;Attribute for clear screen.
MOV AX,600h;AH=6, scroll entire window.
BIOSCALL;Invoke BIOS.
MOV BH,0;Screen number to use.
MOV DX,00;Provide locate coords.
MOV AH,2;AH=2, locate.
BIOSCALL;Invoke BIOS.
RET;Back to invoking mainline.
CLRSCRNENDP
;
FIXNAMEPROC NEAR;Sets up and opens file.
PRTTEXT MESSG3;Ask for patient surname.
READKEYB;Input response.
MOVWORDS PNAME;Puts name into buffer.
;
MOV SI,OFFSET PNAME;Prime source index.
MOV DI,OFFSET PARMB.FNAME ;Prime dest. index.
MOV CX,2;Filename = 2 letters + date.
REP MOVS PARMB.FNAME,PNAME;Move string.
MOV SI,OFFSET DATE;Prime source index.
MOV DI,OFFSET PARMB.FDATE ;Prime destination.
MOV CX,6;Set length.
REP MOVS PARMB.FDATE,DATE ;Move string.
;
CRETURN
LINEFEED 2
PRTTEXT MESSG4;Ask for patient disk.
READKEYB;Get response (enter).
MOV PARMB.DRIVE,2
MOV DX,OFFSET PARMB;Give DOS FCB location.
;
MOV PARMB.FNAMEEXT,3Ah;Prime '9'+1h
FIND:NOP
SUB PARMB.FNAMEEXT,1;Decrement extension.
CMP PARMB.FNAMEEXT,'0';First file?
JE FIRST_FILE;Yes-stop looking.
;
MOV AH,11h;Function selected=find file.
DOSCALL;Invoke DOS.
CMP AL,0;File found?
JNE FIND;No-go back and decrement.
;
;At this point the proper extension has been
;found.
;
FIRST_FILE:NOP;Entry if extension = '1'.
INC PARMB.FNAMEEXT;Go to next extension.
MOV AH,16h;Function=create file.
DOSCALL;Invoke DOS.
;
RET;Back to invoking line+1.
```

```
FIXNAMEENDP
;
DATAWRITEPROC NEAR;PROC for writing out dataset.
NEW_DISK_REWRITE:NOP
MOV PARMB.RECSIZE,768;Set record size.
MOV PARMB.CRNUM,0;Initialize current record no.
MOV DX,OFFSET PARMB;Get file control block.
MOV AH,24h;Set random record field.
DOSCALL;Invoke DOS.
;
;
MOV DX,OFFSET FDATA
MOV AH,1Ah;Set DTA.
DOSCALL;Invoke DOS.
;
;
MOV DX,OFFSET PARMB
MOV CX,49;Record count for run.
MOV AH,28h;Block write.
DOSCALL;Invoke DOS.
;
;Check to see if write successful.
;
CMP AL,01;Insufficient disk space?
JNE DISK_HAS_ROOM;OK--carry on.
MOV AH,10h;Close useless file.
DOSCALL
MOV AH,13h;Kill useless file.
DOSCALL
;
LINEFEED 2
PRTTEXT MESSG10;Ask for new disk.
READKEYB;Wait for OK.
MOV DX,OFFSET PARMB;Get FCB.
MOV AH,16h;Function selected=create file.
DOSCALL;Invoke DOS.
JMP FAR PTR NEW_DISK_REWRITE
;
DISK_HAS_ROOM:  NOP
MOV DX,OFFSET PARMB
MOV AH,10h;Close file.
DOSCALL;Invoke DOS.
;
RET;Back to invoking line+1.
DATAWRITEENDP
;
REDO_ANY_TRAVERSES PROC NEAR;Procedure to allow redoing trav.
 PRTTEXT MESSG6;Ask if user desires reacquisition.
READKEYB;Get response.
MOV AL,'N';Get comparator.
CMP PARMA.TEXTIN,AL
JE RETURN;Back to mainline.
MOV AL,'n';Get comparator.
CMP PARMA.TEXTIN,AL
JE RETURN;Back to mainline.
```

```
PRTTEXT MESSG7;Which traverse?
READKEYB;Get response.
CRETURN
LINEFEED 1
MOV AL,PARMA.TEXTIN;Get input byte.
AND AL,0Fh;Deconvert.
MOV TRAVERSE_NUM,AL;Save traverse number.
MOV CH,0;Clear upper byte.
MOV CL,AL;Get traverse number.
MOV BX,0;Clear base register.
MOV AX,0;Clear accumulator.
FIND_BASE3:NOP
ADD BX,6144;Move to next trav buffer.
LOOP FIND_BASE3;Loop until CX zero.
SUB BX,6144;Correct for loop.
MOV CX,3072;Words/traverse.
WIPEOUT:NOP
MOV [BX],AX;Clear a word.
ADD BX,2;Increment to next word.
LOOP WIPEOUT;Clear until CX zero.
MOV FLAGGER,1;Set flag.
RETURN:NOP
CRETURN
LINEFEED 1
RET
REDO_ANY_TRAVERSESENDP
```

```
COMMENT#
Following is the analog to digital conversion module.
The program makes use of the timer capabilities of the A-D
board.
#
;
;
ADDRESSEQU 710h;IO starting address.
CONTROL_PRTEQU ADDRESS+4;Main control port for board.
CHANNUMEQU ADDRESS+5;AD channel number in auto-inc.
;mode this is the lowest channel sampled.
CONVERT_STARTEQU ADDRESS+6;Software start port.
STATUS_PRTEQU ADDRESS+4;IO port for reading status.
LOW_BYTEEQU ADDRESS+5;Low byte port.
HIGH_BYTEEQU ADDRESS+6;High byte port.
;
;Reading this byte resets the done flip-flop on the board.
;
DATA_9513EQU ADDRESS+8;9513 timer data port.
CTRL_9513EQU ADDRESS+9;9513 timer control port.
;
BOARDSETPROC NEAR;Sets up TECMAR BOARD.
;Timer is set but not armed.
;
MOV DX,HIGH_BYTE;Get high byte port.
IN AL,DX ;Read and reset board.
;
MOV DX,CONTROL_PRT;Get master control port.
MOV AL,10000100B;Start configuration.
OUT DX,AL;Configure board:
;disable all interrupts.
;disable auto inc.
;enable external start.
; (will toggle.)
;
MOV DX,CTRL_9513;Get timer control port.
MOV AL,00010111B;Set to master mode register.
OUT DX,AL;Send to board.
;
MOV DX,DATA_9513;Get data port of timer.
MOV AL,00000000B;LOW master mode byte.
OUT DX,AL;Send to board.
MOV AL,10000000B;HIGH master mode byte.
OUT DX,AL;Send to board.
;
;Master mode is set as follows:
;BCD division.
;enable increment.
;8-bit data bus (8088).
;FOUT on.
;Divide FOUT by 16.
;FOUT source is F1, 1M hertz clock.
;Disable comparators and time of day.
;
MOV DX,CTRL_9513;Get 9513 control port.
```

```
MOV AL,05h;Counter mode register set to five.
OUT DX,AL;Send to board.
;
MOV DX,data_9513;Get 9513 data port.
MOV AL,00110001B;Set low byte.
OUT DX,AL;Send to board.
;
MOV AL,00011101B;Set high byte.
OUT DX,AL;Send to board.
;
;This configures the counter mode register as follows:
;No gating.
;Count rising edges.
;Use F3 as the source (10k hertz).
;Disable special gate.
;Reload from load.
;Repetitive count.
;BCD count.
;Count down.
;Active high TC pulse.
;
MOV AL,TIMER_COUNTL;Get required count value low byte.
OUT DX,AL;Prime counter.
MOV AL,TIMER_COUNTH;Get high count byte.
OUT DX,AL;Send to board.
;
;
;Counter five is now ready to go.
;
;By enabling and disabling the external start, the user
;can control the AD start conversions.
;
RET;Return to invoking program.
BOARDSETENDP
;
CLEAN_BUFFERPROC NEAR;Procedure to clean data buffers.
MOV CX,18816;Number of words/run.
MOV AX,0;Clear accumulator.
MOV BX,0;Start at the beginning of the dseg.
LOOP_CLEAN1:NOP
MOV [BX],AX;Clean a word.
ADD BX,2;Increment to next word.
LOOP LOOP_CLEAN1
;
RET
CLEAN_BUFFERENDP
;
OPTION_SETPROC NEAR;Procedure to set sampling rate.
PRTTEXT MESSG11;What sampling rate?
READKEYB;Input response.
CRETURN
LINEFEED 2
CMP PARMA.TEXTIN,0Dh;[ENTER] (default).
JE DEFAULT_HZ;Yes, go for 100 hz.
AND PARMA.TEXTIN,0Fh;Deconvert ASCII.
```

```
MOV CH,0;Clear high byte.
MOV CL,PARMA.TEXTIN
MOV BX,OFFSET PARMC;Get data struc for counter.
;
;CX now has counter in it.
;
FIND_COUNTER:NOP
ADD BX,2
LOOP FIND_COUNTER;Loop until CX zero.
SUB BX,2;Correct for loop.
MOV AX,[BX];Get counter.
;
MOV TIMER_COUNTL,AL;Prime low byte.
MOV TIMER_COUNTH,AH;Prime high byte.
RET;Back to mainline.
DEFAULT_HZ:NOP
MOV TIMER_COUNTL,82h;Move in default counter.
MOV TIMER_COUNTH,00h;10000/0082=120 hz [BCD].
RET;Back to mainline.
OPTION_SETENDP
```

```
SUBTTL MACRO DEFINITIONS
;
;
READKEYBMACRO;Macro to read the keyboard.
MOV DX,OFFSET PARMA
MOV AH,10;Function selected=read keyboard.
DOSCALL;Invoke DOS.
ENDM
;
LINEFEEDMACRO COUNT;No. of linefeeds is passed.
LOCAL LOOP1
MOV CX,COUNT
LOOP1:MOV DL,0Ah;ASCII for linefeed.
MOV AH,2;Function selected=output char.
DOSCALL;Invoke DOS.
LOOP LOOP1
ENDM
;
CRETURNMACRO;Macro to outpur carriage return.
MOV DL,0Dh;ASCII for carriage return.
MOV AH,2;Function selected=output char.
DOSCALL;Invoke DOS.
ENDM
;
PRTTEXTMACRO TEXT;Macro to output text to the console.
MOV DX,OFFSET TEXT;Get text address.
MOV AH,9;Function selected=output text.
DOSCALL;Invoke DOS.
ENDM
;
MOVWORDSMACRO DEST;Transfers console strings to buffers.
MOV CH,0;Clear MSB of counter.
MOV CL,PARMA.NAMESIZE;Gets length (less CR).
MOV SI,OFFSET PARMA.TEXTIN;Primes source index.
MOV DI,OFFSET DEST;Primes destination index (ES).
REP MOVS DEST,PARMA.TEXTIN;Performs move.
ENDM
;
BEEPMACRO;This macro sounds bell.
MOV DL,7h;ASCII  for beep.
MOV AH,2;Function selected=output char.
DOSCALL;Invoke DOS.
ENDM
;
BIOSCALLMACRO;This macro invokes BIOS.
INT 10h;Call interrupt.
ENDM
;
DOSCALLMACRO;This macro invokes DOS.
INT 21h;Call interrupt.
ENDM
```

```
10    CLS:PRINT "  THIS PROGRAM IS FOR PERFORMING RESOLUTION TESTS RE: CENT
RE OF PRESSURE , AND  PROVIDES THE  OPTION FOR A DATA PRINTOUT, OR PECK. IT
 WORKS ON ******** UNEDITED DATA *********"
20    PRINT
30    PRINT "BEFORE PROCEEDING FURTHER, PLEASE ENSURE THAT THE PLOTTER IS C
ONNECTED THROUGH  COM 2 AND THAT BOTH THE PRINTER AND PLOTTER ARE ON"
40    OPEN "COM2:2400,S,7,1,RS,CS,DS32000,CD"AS#3
50    PRINT
60    DIM DATACNT(384),DATAPTS(384,8),DCOFF(9),CALDATA(384,6),VSUM(384),
          CPX(384),CPY(384),VERT(6),CALIB(9),TYPE$(6),DOT(3,2)
70    GOSUB 470       '   subroutine to load in CALIB(I) factors
80    INPUT "Insert desired data disk (unedited) in drive b.",C%
90   PRINT:PRINT
100   PRINT "Available files:  "
110   PRINT: FILES "B:*.*"
120   PRINT
130   INPUT "Desired data file  ",N$
140   F$="B:"+N$                  'Ensure drive B.
150   CLS
160 '
170   IF T%=0 THEN GOSUB 660                          'Get traverse counters.
180   IF T%=0 THEN INPUT "SELECT AT TRAVERSE  ";T%
190   CLS
200   PRINT "Data points taken for traverse";T%;"=";DATACNT(T%)
210   GOSUB 830        '  get one traverse worth of data.
220  ' NOW HAS ONE TRAVERSE OF DATA
230   FOR H%=1 TO 3               'Loop through 3 channels.
240      SUM=0
250      IF DATACNT(T%)>381 GOTO 330
260      FOR X=0 TO 2
270        ENDER% = DATACNT(T%)
280        SUM=DATAPTS(ENDER%-X,H%)+SUM
290      NEXT X
300     'The above x driven loop gets last 3 pts of traverse for D.C.
310     'offset purposes.
320      SUM=SUM/3        'Average D.C. offset.
330      FOR Q%=1 TO DATACNT(T%)
340        DAT=DATAPTS(Q%,H%)       'GET DATAPTS
350        DAT=DAT-SUM     'CORRECT DC OFFSET
360        DAT=DAT/CALIB(H%)
370        DATAPTS(Q%,H%)=DAT
380       NEXT Q%
390       PRINT "CHANNEL ";H%;" EDITED"
400   NEXT H%         'Next channel.
410   REM      At this point one complete traverse has been edited.
420   PRINT:PRINT"FILE: "N$" TRAVERSE "T% "HAS NOW BEEN EDITED ":
               PRINT"NOW FINDING CPX,ETC.":PRINT
430   GOTO 1050 : 'BLOCK 2
440 '
450 '***** BLOCK 1 SUBROUTINES FOLLOW *******************
460 '
470 '****SUBROUTINE TO READ IN CALIB(I) FACTORS FROM CALROSS ********
480     REM
490     PRINT "Please enter calibration disk into drive B."
500     INPUT "Strike [ENTER] when ready...",CRAP%
```

```
510    PRINT:FILES "B:*.*"
520    PRINT:INPUT "SELECT FILE (without filename extension ASSUMES NO 'R '
):",CAL$
530    CAL$="B:"+CAL$ +" RCAL"  'ENSURES CALROSS FILES ONLY USED
540    OPEN CAL$ FOR INPUT AS #1
550    CLS:PRINT "CALIBRATION FACTORS READ FROM FILE":PRINT
560    FOR I=1 TO 9
570    INPUT#1,CALIB(I)
580    PRINT CALIB(I)
590    NEXT I
600    INPUT "CONTINUE WITH EDIT (Y/N)";ANS$
610    IF ANS$="N" OR   ANS$="n" THEN END
620    CLOSE#1
630    RETURN
640 '
650 '
660 'SUBROUTINE  IS USED TO GET THE TRAVERSE COUNTERS.
670     OPEN F$ AS #1 LEN=768
680     FIELD #1,192 AS A$,192 AS B$,192 AS C$,192 AS D$
690     REC=49  'Last record contains desired data.
700     GET #1,REC      'Read data.
710     FOR I=1 TO 96
720       K=I*2-1
730       DATACNT(I)=CVI(MID$(A$,K,2))
740       DATACNT(I+96)=CVI(MID$(B$,K,2))
750       DATACNT(I+192)=CVI(MID$(C$,K,2))
760       DATACNT(I+288)=CVI(MID$(D$,K,2))
770NEXT I
780 'DATACNT NOW IS PRIMED WITH TRAVERSE COUNTERS.
790 'FILE IS LEFT OPEN, AS DATA WILL BE READ IN MAIN-LINE.
800RETURN'Back to mainline.
810   REMEND OF SUBROUTINE
820 '
830REM Subroutine for reading unedited data.
840 'FILE IS ALREADY OPEN.
850 '
860    FOR CH%=1 TO 3          'Loop through 3 channels.
870REC=(CH%+(T%-1)*8)
880GET #1,REC
890PRINT CH%;" READ.."
900FOR I=1 TO 96
910K=I*2-1
920DATAPTS(I,CH%)=CVI(MID$(A$,K,2))
930DATAPTS(I+96,CH%)=CVI(MID$(B$,K,2))
940DATAPTS(I+192,CH%)=CVI(MID$(C$,K,2))
950DATAPTS(I+288,CH%)=CVI(MID$(D$,K,2))
960NEXT I
970FOR I=1 TO 384
980     IF DATAPTS(I,CH%) > 32767 THEN DATAPTS(I,CH%)=DATAPTS(I,CH%)-65536!
990NEXT I
1000NEXT CH%
1010PRINT "TRAVERSE ";T%;" READ"
1020RETURN'Back to mainline.
1030   REMEND OF SUBROUTINE
1040 '
```

```
1050  '******* END OF BLOCK 1 (EDITING SECTION) **************
1060  '
1070  '****** START BLOCK 2 *********
1080  VTHRESH=10: '10 LBS. MIN. BEFORE CPX,ETC. CALCULATED
1090  TBASE=92.5:'BASE WIDTH INCHES
1100  HEIGHT=45.6:'PERPENDICULAR HEIGHT AT APEX
1110  ' ROUTINE TO FIND CPX,ETC. FOLLOWS
1120      FOR Q%=1 TO DATACNT(T%)
1130           VSUM(Q%)=DATAPTS(Q%,1)+DATAPTS(Q%,2)+DATAPTS(Q%,3)
1140           'FOLLOWING IS NEEDED FOR IDENTIFYING H.STRIKES
1150           'following finds CPX,FCPX
1160           'the alternate formulas for determing CPX,CPY
                  are commented out
1170           NUM1=(DATAPTS(Q%,2)+2*DATAPTS(Q%,3))*TBASE/2
1180           CPX(Q%)=NUM1/VSUM(Q%) :'inches -largest value 92.5
1190           'cpx=0 at NE corner,increases going west
1200           'following finds CPY,FCPY
1210           NUM2=DATAPTS(Q%,2)*HEIGHT
1220           CPY(Q%)=NUM2/VSUM(Q%) : 'inches-largest value is 45.6
1230           'cpy=0 at N side,increases going south
1240      NEXT Q%
1250  ' MAIN MENU FOLLOWS
1260  PRINT"SELECT ONE OF:   (1) DATA PRINTOUT       (2) PECK
                               (3) SCREEN PRINTOUT     (4) GET NEW TRAVE
RSE                                (5) GET NEW FILE    "
1270  INPUT CD$
1280  IF CD$="1" THEN CLS:GOTO 1340: ' PRINT ROUTINE
1290  IF CD$="4" THEN CLS:INPUT"SELECT TRAVERSE NO. ";T%:
         GOTO 200 :'NEW TRAVERSE
1300  IF CD$="5" THEN CLS:T%=0:CLOSE#1: GOTO 100  :'NEW FILE
1310  IF CD$="2" THEN CLS:GOTO 1460 :'PECK PLOT
1320  IF CD$="3" THEN GOTO 1880:'SCREEN PRINT
1330  GOTO 1260: 'TRAP
1340 ' PRINT ROUTINE FOLLOWS*************************************
1350  BEEP:INPUT" PRESS RETURN IF PRINTER IS READY",POOP$
1360  LPRINT "FILENAME: "N$" TRAVERSE NO: "T%:LPRINT
1370  LPRINT"PT       VSUM     CPX      CPY     "
1380  FOR Q%=1 TO DATACNT(T%) :'USE DATACNT LATER
1390    LPRINT Q% TAB(6);:LPRINT USING"  ###.##";VSUM(Q%),CPX(Q%),CPY(Q%)
1400  NEXT Q%
1410  LPRINT:LPRINT:LPRINT
1420  CLS
1430  GOTO 1260 :' MAIN MENU
1440 ' END OF PRINTOUT SECTION
1450 '
1460 'PECK ROUTINE FOLLOWS ***********************************
1470  INPUT"SELECT  (1)C/W BACKGROUND   (2)POINTS ONLY   ",PSE
1480  IF PSE<>1 AND PSE<>2 THEN 1470:'TRAP
1490  BEEP:INPUT"PRESS RETURN IF PLOTTER IS READY",POOP$
1500  IF PSE=2 THEN PRINT #3,"IN;IP 1150 1825 9150 5825;SC 0.000 92.50 0.00
0 45.60;SP1;":GOTO 1790
1510  PRINT #3,"IN;SC0,9999,0,9999;SP1;"
1520  PRINT #3,"DT";+CHR$(1);";DI 0,1;PU 0,2000;LB FILENAME:";N$+CHR$(1);";
"
1530  IF ALREADYREAD=1 THEN GOTO 1600
```

```
1540  DATA 78.375,6.8756,24.25,18.5,50.75,30.75
1550  FOR A=1 TO 3
1560   READ X,Y
1570   DOT(A,1)=X:DOT(A,2)=Y
1580  NEXT A
1590  ALREADYREAD=1
1600  PRINT #3,"DI 0 1;"
1610  PRINT #3,"IN;IP 1150 1825 9150 5825;SC 0.000 8.000 0.000 4.000;SP 1;P
A 0 0;PD 8 0;PA 4 4 0 0;PU 0.5 3;PD 1.5 3;PU 1 3.5;PD 1 2.5;" :' DRAWS TRIA
NGLE
1620  PRINT #3,"DT";+CHR$(1);";"
1630  PRINT #3,"DI 0,1;PU 1 2.25;LB N";+CHR$(1);";"
1640  PRINT #3,"DI 0,1;PU 1 3.5;LB S";+CHR$(1);";"
1650  PRINT #3,"DI 0,1;PU 0.4 2.875;LB E";+CHR$(1);";"
1660  PRINT #3,"DI 0,1;PU 1.65 2.875;LB W";+CHR$(1);";"
1670  PRINT #3,"SC 0.000 92.50 0.000 45.60;"
1680  PRINT:INPUT" HOW MANY DOTS DO YOU WANT PLOTTED:",TOTDOT
1690  PRINT:INPUT" HOW MANY NON-STANDARD DOTS ARE INCLUDED IN THIS TOTAL",E
XTRA
1700  IF EXTRA=0 THEN 1750
1710  FOR A=4 TO EXTRA+3
1720     PRINT:INPUT" ENTER ~ X,Y ~ COORDINATES OF EXTRA POINT: ",DOT(A,1),
DOT(A,2       )
1730     PRINT,"THE NUMBER NOW ASSOCIATED WITH THIS DOT IS: ";A
1740  NEXT A
1750  FOR I=1 TO TOTDOT
1760    PRINT:INPUT"DOT NUMBER ?",DOTNUM
1770    PRINT #3,"PU ";DOT(DOTNUM,1);" ";DOT(DOTNUM,2);";CI .5;LB ";STR$(DO
TNUM)+CHR$(1);";"
1780  NEXT I
1790  FOR J=1 TO DATACNT(T%)
1800   X=CPX(J):Y=CPY(J)
1810   IF VSUM(J)<10 OR (Y=0) OR (X=0) THEN GOTO 1830: 'DON'T UNRELIABLES
1820   PRINT #3,"PU;PA ";X;" ";Y;";PD;"
1830  NEXT J
1840  PRINT #3,"PU;SP1;PU 0,0;"
1850  CLS
1860  GOTO 1260: 'MAIN MENU
1870 '
1880 ' SCREEN PRINTOUT FOLLOWS *********************************
1890  CLS:PRINT "NO."," VSUM"," CPX"," CPY"
1900  FOR I=1 TO DATACNT(T%)
1910   PRINT I,VSUM(I),CPX(I),CPY(I)
1920   IF I<> 20 AND I<>40 AND I<>60 AND I<>80 AND I<>100 THEN 1960
1930   LOCATE 23,20:PRINT" PAUSE (STRIKE ANY KET TO CONTINUE)"
1940   X$=INKEY$:IF X$="" THEN 1940
1950   CLS: 'LOCATE 23,20:PRINT "                                        "
1960  NEXT I
1970  LOCATE 23,20:PRINT" STRIKE ANY KEY TO CONTINUE"
1980  ZZ$=INKEY$:IF ZZ$="" THEN 1980
1990  CLS
2000  GOTO 1260: 'MAIN MENU
```

```
10    '!!!!!!!!!!!!! START OF EDIT BLOCK 1 !!!!!!!!!!!!!!!!!!!!
15 'THIS PROGRAM WAS FORMERLY CALLED PSEP1TO
20  COLOR 0,7 : 'REVERSES SCREEN
30  CLS: PRINT " MAKE SURE PLOTTER IS TURNED ON"
40  OPEN "COM2:2400,S,7,1,RS,CS,DS32000,CD"AS#3
50  DIM DATAPTS(384,8),C(8),STRIKE(5),HIT(4,10),VSUM(384),CPX(384),CPY(384)
,        VCPX(384),VCPY(384),RVCP(384),VA(384),TYPE$(6)
60  DIM AXES(32),AXES2(32),H(10),SDVSUM(384),HS(5),FDVSUM(384),ZC(5),
        DATACNT(10),DCOFF(9),VERT(6),CALIB(9),ALTF(200)
70 ' FOLLOWING READS IN DATA USED FOR DRAWING AXES
80   FOR I=1 TO 32: ' FOR CPX,ETC.PLOT
90   READ AXES(I)
100   NEXT I
110   DATA 200,1700,2650,1700,2650,9900,2650,1700 :'vert.forces
120   DATA 3140,1700,5100,1700,4120,9900,4120,1700 :'CPX,CPY
130   DATA 5590,1700,7550,1700,6570,9900,6570,1700 :'VCPX,VCPY
140   DATA 8039,1700,9999,1700,9019,9900,9019,1700 :'RVCP,VA
150   FOR I=1 TO 24 : 'FOR FORCE PLOT
160  READ AXES2(I)
170  NEXT I
180  DATA 700,1700,3300,1700,3300,9900,3300,1700
190  DATA 3700,1700,6300,1700,5000,9900,5000,1700
200  DATA 6700,1700,9300,1700,8000,9900,8000,1700
210  GOSUB 720       '   subroutine to load in CALIB(I) factors
220  ' NOW HAVE CALIBRATORS
230  INPUT "Insert desired data disk (unedited) in drive b.",C%
240  PRINT:PRINT "Available files:  "
250  PRINT:FILES "B:*.*"
260  PRINT:INPUT "Desired data file  ",E$
270  F$="B:"+E$                 'Ensure drive B.
280  IF T%=0 THEN GOSUB 890 :'Get traverse counters ONLY ONCE PER FILE
290  INPUT"SELECT TRAVERSE ",T%
300  PRINT "Data points taken for traverse";T%;"=";DATACNT(T%)
310  GOSUB 1030       '  get one traverse worth of data.
320  PRINT"SELECT THE FIRST OR LAST THREE DATAPTS FOR D.C.OFFSET CORRECTION
"
330  INPUT " (F)  (L)  ",LDCOFF$
340  IF LDCOFF$<>"F"AND LDCOFF$<>"f"AND LDCOFF$<>"L"AND LDCOFF$<>"l" THEN 3
30
350  ' commence edit
360  FOR H%=1 TO 6 : 'Loop through 6 channels. 7 & 8 EDITED IN BLOCK2
370    SUM=0
380    IF DATACNT(T%)>381 THEN BEEP:PRINT"******* DATACOUNT>381 ***********
***":      GOTO 470 : 'CANNOT CORRECT FOR OFFSET
390    FOR X=0 TO 2
400      ENDER% = DATACNT(T%)
410      IF LDCOFF$="F" OR LDCOFF$="f"THEN SUM=DATAPTS(X+1,H%)+SUM
420      IF LDCOFF$="L" OR LDCOFF$="l"THEN SUM=DATAPTS(ENDER%-X,H%)+SUM
430    NEXT X
440  ' The above x driven loop gets the first or last 3 pts of traverse for
 D.C.
450  '   offset purposes.
460    SUM=SUM/3        'Average D.C. offset for this channel
470    FOR Q%=1 TO DATACNT(T%)-1
480      DAT=DATAPTS(Q%,H%)     'GET DATAPTS
```

```
490       DAT=DAT-SUM    'CORRECT DC OFFSET
500       IF H% >3 THEN GOTO 530
510       DAT=DAT/CALIB(H%)
520       GOTO 620   'LOOP AGAIN
530       REM entrance point fo M.L. correction
540       IF H%<>4 THEN 560
550       IF DAT<0 THEN DAT=DAT/CALIB(6) ELSE DAT=DAT/CALIB(4)
560       REM entrance point for M.L. correction
570       IF H%<> 5 THEN 590
580       IF DAT<0 THEN DAT=DAT/CALIB(7) ELSE DAT=DAT/CALIB(5)
590       REM entrance for P.A.
600       IF H%<>6 THEN 620 : 'NO EDIT OF CHS 7,8
610       IF DAT<0 THEN DAT=DAT/CALIB(9) ELSE DAT=DAT/CALIB(8)
620       DATAPTS(Q%,H%)=DAT
630    NEXT Q%
640    PRINT "CHANNEL  ";H%;" EDITED"
650   NEXT H%          'Next channel.
660   REM        At this point one complete traverse has been edited.
670 '      N$=MID$(F$,3,LEN(F$)): 'FILE NAME USED IN BLOCK 2
680   PRINT:PRINT "FILE " E$   " TRAVERSE " T% "HAS NOW BEEN EDITED"
690   PRINT "PROCEEDING TO BLOCK 2"
700   GOTO 1250: 'BLOCK 2
710 ' BLOCK 1 SUBROUTINES FOLLOW
720 '****SUBROUTINE TO READ IN CALIB(I) FACTORS FROM CALROSS ********
730     PRINT "Please enter calibration disk into drive B."
740     INPUT "Strike [ENTER] when ready...",CRAP%
750     FILES "B:*.*"
760     INPUT "Desired file (without filename extension):  ",CAL$
770     CAL$="B:"+CAL$+" RCAL": 'ENSURES DRIVE B:
780     OPEN CAL$ FOR INPUT AS #1
790     CLS:PRINT "CALIBRATION FACTORS READ FROM FILE":PRINT
800     FOR I=1 TO 9
810      INPUT#1,CALIB(I)
820      PRINT CALIB(I)
830     NEXT I
840     INPUT "CONTINUE WITH EDIT (Y/N)";ANS$
850     IF ANS$="N" OR   ANS$="n" THEN END
860     CLOSE#1
870     RETURN
880 '
890 'SUBROUTINE  IS USED TO GET THE TRAVERSE COUNTERS.
900OPEN F$ AS #1 LEN=768
910      FIELD #1,192 AS A$,192 AS B$,192 AS C$,192 AS D$
920      REC=49  'Last record contains desired data.
930      GET #1,REC       'Read data.
940      FOR I=1 TO 10 : ' ONLY 6 COUNTS USED
950        K=I*2-1
960        DATACNT(I)=CVI(MID$(A$,K,2))
970NEXT I
980 'DATACNT NOW IS PRIMED WITH TRAVERSE COUNTERS.
990 'FILE IS LEFT OPEN, AS DATA WILL BE READ IN MAIN-LINE.
1000RETURN'Back to mainline.
1010    REMEND OF SUBROUTINE
1020 '
1030REM Subroutine for reading unedited data.
```

```
1040 'FILE IS ALREADY OPEN.
1050FOR CH%=1 TO 8'Loop through 8 channels.
1060REC=(CH%+(T%-1)*8)
1070GET #1,REC
1080    PRINT "CHANNEL  "CH%;" READ.."
1090FOR I=1 TO 96
1100       K=I*2-1
1110       DATAPTS(I,CH%)=CVI(MID$(A$,K,2))
1120       DATAPTS(I+96,CH%)=CVI(MID$(B$,K,2))
1130       DATAPTS(I+192,CH%)=CVI(MID$(C$,K,2))
1140       DATAPTS(I+288,CH%)=CVI(MID$(D$,K,2))
1150    NEXT I
1160    FOR I=1 TO 384
1170    IF DATAPTS(I,CH%) > 32767 THEN DATAPTS(I,CH%)=DATAPTS(I,CH%)-65536!
1180    NEXT I
1190    NEXT CH%
1200    PRINT
1210    RETURN  'Back to mainline.
1220   REMEND OF SUBROUTINE
1230 '!!!!!!!!!!!! END OF EDIT BLOCK 1!!!!!!!!!!!!!!!!!!!
1240 '
1250 '@@@@@@@@@@@@ START BLOCK 2 @@@@@@@@@@@@@@@@@@@@
1260  VTHRESH=10: '10 LBS. MIN. BEFORE CPX,ETC. CALCULATED
1270  TBASE=92.5:'BASE WIDTH INCHES
1280  HEIGHT=45.6:'PERPENDICULAR HEIGHT AT APEX
1290  SWITCH=200: ' THIS IS ABOUT 1 VOLTS ON A/D
1300  H(1)=-.97226:H(2)=.4462:H(3)=-.25667:H(4)=.15497:H(5)=-.092024
          :H(6)=.051667:H(7)=-.0268:H(8)=.01347:H(9)=-8.889998E-03: 'FOR D
IG DIFF.
1310  PRINT:INPUT "GIVE TRAVERSE DESCRIPTOR TYPE (e.g.WRLR )  ",TYPE$
1320  PRINT:PRINT "A FEW MOMENTS PLEASE WHILE I DETERMINE CPX,CPY,ETC."
1330  ' ROUTINE TO FIND CPX,ETC. FOLLOWS
1340  FOR Z=DATACNT(T%) TO 384:VSUM(Z)=0:CPX(Z)=0:CPY(Z)=0:NEXT:
                'CLEAR ARRAY  FROM LAST USE
1350  FOR Q%=1 TO DATACNT(T%)
1360    IF DATAPTS(Q%,7)<SWITCH THEN DATAPTS(Q%,7)=0 ELSE
             DATAPTS(Q%,7)=1: 'PUTS 1'S OR 0'S IN SWITCH CHANNEL 7
1370    IF DATAPTS(Q%,8)<SWITCH THEN DATAPTS(Q%,8)=0 ELSE
             DATAPTS(Q%,8)=1: 'PUTS 1'S OR 0'S IN SWITCH CHANNEL 8
1380    VSUM(Q%)=DATAPTS(Q%,1)+DATAPTS(Q%,2)+DATAPTS(Q%,3)
1390    ' following is later used for h.s. ident routine using sdvsum
1400    IF (Q%<20) OR (DATACNT(T%)-Q%) <20 THEN GOTO 1460 : 'IGNORE ENDS
1410    FVSUM=DATAPTS(Q%+1,1)+DATAPTS(Q%+1,2)+DATAPTS(Q%+1,3): 'FUTURE SUM
1420    PVSUM=DATAPTS(Q%-1,1)+DATAPTS(Q%-1,2)+DATAPTS(Q%-1,3): 'PAST SUM
1430    FDVSUM(Q%)=FVSUM-VSUM(Q%): 'PRESENT FIRST DIFFERENCE
1440    PFDVSUM=VSUM(Q%)-PVSUM: 'PAST FIRST DIFFERENCE
1450    SDVSUM(Q%)=FDVSUM(Q%)-PFDVSUM :' SECOND DIFFERENCE
1460    IF VSUM(Q%)<VTHRESH THEN CPX(Q%)=0:CPY(Q%)=0:VCPX(Q%)=0:
             VCPY(Q%)=0:RVCP(Q%)=0:VA(Q%)=0:GOTO 1560
1470    'BOARD MUST HAVE MIN LOAD BEFORE FINDING CPX,ETC.
1480    'following finds CPX
1490    NUM1=(DATAPTS(Q%,2)+2*DATAPTS(Q%,3))*TBASE/2
1500    CPX(Q%)=NUM1/VSUM(Q%) :'inches -largest value 92.5
1510    'cpx=0 at NE corner,increases going west
1520    'following finds CPY
```

```
1530    NUM2=DATAPTS(Q%,2)*HEIGHT
1540    CPY(Q%)=NUM2/VSUM(Q%) : 'inches-largest value is 45.6
1550    'cpy=0 at N side,increases going south
1560   NEXT Q%
1570 ' ROUTINE TO FIND VCPX,VCPY,RVCP,VA USING F.I.R.DIFF.
1580 ' TAKEN FROM  ONE-DIM.DIG.SIGNAL PROC.CHI-TSONG CHENG PAGE 204
1590   FOR Z=DATACNT(T%)-20 TO 384:VCPX(Z)=0:VCPY(Z)=0:RVCP(Z)=0: NEXT Z
             : 'CLEARS ARRAY FROM LAST USE
1600   FOR Q%=10 TO DATACNT(T%)-10
1610     VCPX(Q%)=0:VCPY(Q%)=0:RVCP(Q%)=0:'CLEARS ARRAYS (STARTING VALUES
              FOR THIS LOOP)
1620     IF VSUM(Q%-9)<VTHRESH OR VSUM(Q%+9)<VTHRESH THEN 1710
1630     FOR N=-9 TO 9
1640      IF N=0 THEN GOTO 1680
1650      IF N<0 THEN CC=+H(-N) ELSE CC=-H(N):' SIGNS OPPOSITE OF IN BOOK
1660      VCPX(Q%)=VCPX(Q%)+CC*CPX(Q%+N)
1670      VCPY(Q%)=VCPY(Q%)+CC*CPY(Q%+N)
1680     NEXT N
1690     VA(Q%)=180/3.14159*ATN(VCPY(Q%)/VCPX(Q%)): ' IN DEGREES
1700     RVCP(Q%)=SQR(VCPX(Q%)^2+VCPY(Q%)^2)
1710   NEXT Q%
1720 ' ROUTINE TO FIND HEEL STRIKES BASED ON RVCP
1730   J=15
1740   FOR I=20 TO DATACNT(T%)-10
1750    DELTARVCP=RVCP(I+1)-RVCP(I)
1760    IF J>=15 AND DELTARVCP>=.7 AND (RVCP(I+1)+RVCP(I+2)+RVCP(I+3)+RVCP(
I+4)+        RVCP(I+5))/5>=1 THEN PRINT "(RVCP) HEEL STRIKE AT:";I:J=0
1770    J=J+1
1780   NEXT I
1790   PRINT
1800 ' ROUTINE TO FIND HEEL STRIKES BASED ON VERTICAL FORCE DERIVATIVES
1810   K=1:J=20: 'MIN OFFSET BETWEEN HEEL STRIKES
1820   FOR Z%=25 TO DATACNT(T%)-25: 'SKIP FIRST AND LAST PORTIONS
1830    J=J+1
1840    FOR XX%=0 TO 3:SUM=SUM+RVCP(Z%+XX%):NEXT XX%
1850     AVRVCP=SUM/4:SUM=0:IF AVRVCP < 1.5 THEN GOTO 1870
1860     IF SDVSUM(Z%)>1.5 AND SDVSUM(Z%+1)>1.5 AND((SDVSUM(Z%)+SDVSUM(Z%+1
))/2)         >5 AND J>=20 THEN PRINT"(FORCE DER.) HEEL STRIKE  AT " Z% :HS
(K)=Z%:            K=K+1:J=0
1870  NEXT Z%
1880  INPUT"FINISHED READING  HEELSTRIKES?",GARBAGE$
1890  ' MAIN MENU FOLLOWS
1900  CLS:PRINT "WORKING WITH DATA FOR FILE "E$"  TRAVERSE " T%:PRINT
1910  PRINT"SELECT ONE OF:   (1) DATA PRINTOUT    (2) FORCES PLOT (V,M/L,P/
A)                                (3) PLOT CPX,ETC.    (4) PECK
                                       (5) GET NEW TRAVERSE (6) GET NEW FIL
E"
1920  PRINT"                 (7) SCREEN PRINTOUT  (8) DUMP EXTRACTED RECORD
"
1930  CD$=INKEY$:IF CD$="" THEN 1930
1940  IF CD$="1" THEN GOTO 2240: ' PRINT ROUTINE
1950  IF CD$="5" THEN FLAG=0:GOTO 290 :'NEW TRAVERSE
1960  IF CD$="6" THEN FLAG=0:T%=0:CLOSE#1:GOTO 230 :'NEW FILE
1970  IF CD$="7" THEN GOTO 6290: 'SCREEN PRINTOUT
1980  IF CD$="8" THEN GOTO 6500: 'DUMP EXRACTED RECORD
```

```
1990  IF CD$<>"2" AND CD$<>"3" AND CD$<>"4" THEN 1930: 'TRAP
2000  IF FLAG=1 GOTO 2200
2010  FLAG=1
2020  ' SOME PLOT ROUTINE MUST HAVE BEEN SELECTED OR ELSE AN INVALID ANSWER
2030  PRINT:PRINT:INPUT "DO YOU WANT TO GIVE ME THE TEMPORAL EVENTS? (Y/N)
",TE$      :PRINT
2040  IF TE$="N" OR TE$="n" THEN GOTO 2200 :'SKIP FOLLOWING IF N/A
2050  IF TE$<>"Y" AND TE$<>"y" THEN GOTO 2030:'TRAP
2060 'ROUTINE TO INPUT TEMPORAL EVENTS FOLLOWS
2070  PRINT:INPUT;"NUMBER OF RIGHT HEEL STRIKES=";STRIKE(1)
2080  PRINT:INPUT;"NUMBER OF LEFT HEEL STRIKES=";STRIKE(2)
2090  PRINT:INPUT;"NUMBER OF RIGHT TOE OFFS=";STRIKE(3)
2100  PRINT:INPUT;"NUMBER OF LEFT TOE OFFS=";STRIKE(4)
2110  FOR S=1 TO 4
2120   IF STRIKE(S)=0 THEN GOTO 2190
2130   FOR P=1 TO STRIKE(S)
2140     IF S=1 THEN PRINT:PRINT,"PT. OF RIGHT HEEL STRIKE NUMBER: ";P;" ="
          :INPUT;HIT(1,P)
2150     IF S=2 THEN PRINT:PRINT,"PT. OF LEFT HEEL STIKE NUMBER: ";P;" =":
          INPUT;HIT(2,P)
2160     IF S=3 THEN PRINT:PRINT,"PT. OF RIGHT TOE OFF NUMBER: ";P;" =":
          INPUT;HIT(3,P)
2170     IF S=4 THEN PRINT:PRINT,"PT OF LEFT TOE OFF NUMBER: ";P;" =":
          INPUT;HIT(4,P)
2180    NEXT P
2190  NEXT S
2200  PRINT:PRINT
2210  IF CD$="2" THEN GOTO 4380 :'FORCES PLOT
2220  IF CD$="3" THEN GOTO 2490 :'CPX,ETC.PLOT
2230  IF CD$="4" THEN GOTO 3990 :'PECK PLOT
2240 ' PRINT ROUTINE FOLLOWS*************************************
2250  BEEP:INPUT" PRESS RETURN IF PRINTER IS READY",POOP$
2260  CLS:PRINT:INPUT "SELECT ONE   (1)PRINT ALL DATA F/L   (2)PRINT SECTIO
NS ONLY  " ,PD$:IF PD$="1"THEN 2350 ELSE IF PD$="2"THEN 2280 ELSE 2260
2270 ' ABBREVIATED PRINT OUT ROUTINE
2280  PRINT:INPUT"GIVE ME THE NO.OF ZONES AND THE ZONE RANGE (NO.PTS EITHER
 SIDE OF CENTER PT.)         ",NZ,ZR
2290  ON NZ GOTO 2300,2310,2320,2330,2340
2300  INPUT "GIVE ME THE ZONE CENTER      ",ZC(1):GOTO 2360
2310  PRINT "GIVE ME THE "NZ;:INPUT " ZONE CENTERS     ",ZC(1),ZC(2):GOTO 2
360
2320  PRINT "GIVE ME THE "NZ;:INPUT " ZONE CENTERS     ",ZC(1),ZC(2),ZC(3)
       :GOTO 2360
2330  PRINT "GIVE ME THE "NZ;:INPUT " ZONE CENTERS     ",ZC(1),ZC(2),ZC(3),
ZC(4)     :GOTO 2360
2340  PRINT "GIVE ME THE "NZ;:INPUT " ZONE CENTERS     ",ZC(1),ZC(2),ZC(3),
ZC(4),     ZC(5):GOTO 2360
2350  NZ=1:FIR=1:LAS=DATACNT(T%)
2360  LPRINT "FILENAME: "E$" TRAVERSE NO: "T%" TYPE: "TYPE$
       :LPRINT:LPRINT
2370  LPRINT"PT.   L    R    VSUM  M/L   P/A    FDVSUM  SDVSUM  VCPX    VC
PY    RVCP  ":LPRINT
2380  FOR T=1 TO NZ
2390   IF PD$="1" THEN 2400 ELSE FIR=ZC(T)-ZR:LAS=ZC(T)+ZR
2400   FOR Q%= FIR TO LAS
```

```
2410    LPRINT Q% TAB(6)DATAPTS(Q%,7)TAB(11)DATAPTS(Q%,8)TAB(16)INT(VSUM(Q%
))TAB(22)INT(DATAPTS(Q%,4)+DATAPTS(Q%,5))TAB(28)INT(DATAPTS(Q%,6))TAB(33);:
LPRINT USING"  ###.##";FDVSUM(Q%),SDVSUM(Q%),VCPX(Q%),VCPY(Q%),RVCP(Q%)
2420   NEXT Q%
2430  LPRINT
2440  NEXT T
2450  GOTO 1890: 'MAIN MENU
2460 ' END OF PRINTOUT SECTION
2470 '********************************************************
2480 '
2490 ' CPX,ETC. PLOT ROUTINE FOLLOWS ***************************
2500 BEEP:INPUT"PRESS RETURN IF PLOTTER IS READY",POOP$
2510 PRINT:PRINT"SELECT:   (1)PLOT VA    (2)PLOT SDVSUM    (3)PLOT NEITHER"
2520 VAF$=INKEY$: IF VAF$="" THEN 2520
2530 IF VAF$<>"1" AND VAF$<> "2" AND VAF$<>"3" THEN 2520: 'TRAP
2540 GOSUB 3550:'AXES AND LABELS SUBROUTINE
2550 GOSUB 2590 :'DATA PLOTTING ROUTINE
2560 GOTO 1900 : 'MAIN MENU
2570 '
2580 '**********************************************************
2590 ' CPX.ETC. DATA PLOTTING SUBROUTINE
2600 'PLOTS VSUM
2610 Y=1700
2620 PRINT #3,"PU;SP2;PU 2650 ";Y;";PD;"
2630 INCR=8200/DATACNT(T%): 'time interval depends on count
2640 VSCALE1=(2650-200)/250
2650 FOR I= 1 TO DATACNT(T%)
2660    Y=Y+INCR
2670    X=2650-VSUM(I)*VSCALE1
2680    PRINT #3,"PA ";X;" ";Y;";";
2690 NEXT I
2700 'PLOTS CPX
2710 Y=1700
2720 PRINT #3,"PU;SP1;PU 4120 ";Y;";PD;"
2730 INCR=8200/DATACNT(T%)
2740 SCALEX=(4120-3140)/92.5
2750 FOR I= 1 TO DATACNT(T%)
2760    Y=Y+INCR
2770    X=4120-CPX(I)*SCALEX
2780    PRINT #3,"PA ";X;" ";Y;";";
2790 NEXT I
2800 '
2810 'PLOTS CPY
2820  POSI = 6570
2830  Y=1700
2840  SCALEY=(6570-5590)/45.6
2850  PRINT #3,"PU ";POSI;" ";Y;";PD;"
2860  FOR I=1 TO DATACNT(T%)
2870     Y=Y+8200/DATACNT(T%)
2880     X=POSI-CPY(I)*SCALEY
2890     PRINT #3,"PA ";X;" ";Y;";";
2900 '
2910  NEXT I
2920 '
2930 'PLOTS RVCP
```

```
2940  POSI = 9019
2950  Y=1700
2960  SCRVCP=2*SCALEX
2970  PRINT #3,"PU ";POSI;" ";Y;";PD;"
2980  FOR I=1 TO DATACNT(T%)
2990    Y=Y+8200/DATACNT(T%)
3000    X=POSI-RVCP(I)*SCRVCP :' 2 x's value for cpx
3010    PRINT #3,"PA ";X;" ";Y;";";
3020  NEXT I
3030 '
3040 PRINT #3,"SP 2;"  'CHANGE TO PEN TWO
3050 '
3060 'PLOTS VCPX, VCPY, VA OR SDVUM OR NEITHER
3070 POSI = 4120
3080 SCVX=SCALEX*2
3090 SCVY=SCALEY*2
3100 FOR GR =1 TO 3
3110   IF VAF$="3" AND GR=3 THEN 3220
3120   Y=1700
3130   PRINT #3,"PU ";POSI;" ";Y;";PD;"
3140   FOR I=1 TO DATACNT(T%)
3150     Y=Y+8200/DATACNT(T%)
3160     IF GR = 1 THEN X=POSI-VCPX(I)*SCVX: '2 x's value for cpx
3170     IF GR = 2 THEN X=POSI-VCPY(I)*SCVY:'2 x's value for cpy
3180     IF GR = 3 THEN IF VAF$="1" THEN X=POSI-VA(I)*(9019-8039)/90
              ELSE IF VAF$="2" THEN X=POSI-SDVSUM(I)*(9019-8039)/90
3190     PRINT #3,"PA ";X;" ";Y;";";
3200    NEXT I
3210  POSI=POSI+2449.5
3220  NEXT GR
3230 '
3240 'FOLLOWING DRAWS TEMPORAL LINES (IF GIVEN ) ON CPX,ETC. PLOT
3250 PRINT #3,"PU;SP1;PU 0,0;"
3260 IF TE$="n" OR TE$="N" THEN GOTO 3410
3270 PRINT:INPUT"DRAW TEMPORAL EVENTS (Y/N) "; D3$:IF D3$<>"Y" AND D3$<>"y"
 AND D3$<>      "N" AND D3$<>"n" THEN GOTO 3270 :'TRAP
3280 IF D3$="N" OR D3$="n" THEN GOTO 3410
3290 PRINT #3,"SP 2;DI 0,1;SI .15 .2;DT";+CHR$(1);";"
3300 FOR S=1 TO 4
3310   IF STRIKE(S)=0 THEN GOTO 3380' NEXT STRIKE
3320   FOR P=1 TO STRIKE(S)
3330      IF S=1 THEN X=2650-VSCALE1*VSUM(HIT(1,P)):Y=1700+HIT(1,P)*8200/DAT
ACNT(T%):PRINT #3,"PU ";2775;" ";Y-190;";PD;LB RH";+CHR$(1);";PU ";X;" ";Y;
";PD 9999 ";Y;";PU;"
3340      IF S=2 THEN X=2650-VSCALE1*VSUM(HIT(2,P)):Y=1700+HIT(2,P)*8200/DAT
ACNT(T%):PRINT #3,"PU ";2775;" ";Y-190;";PD;LB LH";+CHR$(1);";PU ";X;" ";Y;
";PD 9999 ";Y;";PU
3350      IF S=3 THEN X=2650-VSCALE1*VSUM(HIT(3,P)):Y=1700+HIT(3,P)*8200/DAT
ACNT(T%):PRINT #3,"PU ";2900;" ";Y-250;";PD;LB RTO";+CHR$(1);";PU ";X;" ";Y
;";PD 9999 ";Y;";PU;"
3360      IF S=4 THEN X=2650-VSCALE1*VSUM(HIT(4,P)):Y=1700+HIT(4,P)*8200/DAT
ACNT(T%):PRINT #3,"PU ";2900;" ";Y-250;";PD;LB LTO";+CHR$(1);";PU ";X;" ";Y
;";PD 9999 ";Y;";PU;"
3370   NEXT P
3380 NEXT S
```

```
3390 PRINT #3,"PU 0 0;"
3400 'BEGINNING OF TEMPORAL SHIFT
3410 PRINT:INPUT"DO YOU WANT VSUM SHIFT? ",V$:IF V$="N" OR V$="n" THEN GOTO
 3530 ELSE IF V$<>"Y" AND V$<>"y" THEN GOTO 3400
3420 PRINT:INPUT"SOURCE POINT: ",SOURCE:PRINT:INPUT"DESTINATION POINT: ",DE
ST
3430 INCR=8200/DATACNT(T%):VSCALE1=(2650-200)/250
3440 Y=1700+DEST*INCR:X=2650-VSUM(SOURCE)*VSCALE1
3450 PRINT #3,"SP1;LT 6;PU ";X;" ";Y;"; PD;"
3460 FOR I=SOURCE+1 TO DATACNT(T%)
3470    Y=Y+INCR
3480    X=2650-VSUM(I)*VSCALE1
3490    PRINT #3,"PD ";X;" ";Y;";"
3500 NEXT I
3510 'END OF TEMPORAL SHIFT
3520 PRINT #3,"PU;SP1;PU 0,0;"
3530 RETURN
3540 ' ***************************************************************
3550 ' AXES AND LABELS SUBROUTINE FOR CPX,ETC.
3560 PRINT #3,"IN;SC0,9999,0,9999;SP1;"
3570 PRINT #3,"DT";+CHR$(1);";DI 0,1;PU 0,2000;LB FILENAME:";E$+"  TRAVERSE
:";STR$(T%)+"   TYPE: "+TYPE$+CHR$(1);";"
3580 PRINT #3,"DT";+CHR$(1);";DI 0,1;PU 0,8000;TYPE";TYPE$+CHR$(1);";"
3590 FOR I= 1 TO 32 STEP 4' DRAWS AXES
3600    X=INT(AXES(I))
3610    Y=INT(AXES(I+1))
3620    PRINT #3,"LT;PU",X,",",Y,";PD;"
3630    X=INT(AXES(I+2))
3640    Y=INT(AXES(I+3))
3650    PRINT #3,"PA",X,",",Y,";"
3660 NEXT I
3670 '
3680 'DRAWS LABELS
3690  PRINT #3,"DI 0,1;PU 700,7000;LB VERTICAL FORCES";+CHR$(1)
3700 PRINT #3,";PU 3180,7000;LB C.P.X";+CHR$(1)
3710 PRINT #3,";PU 5600,7000;LB C.P.Y.";+CHR$(1)
3720 PRINT #3,";PU 8100,7000;LB RESULTANT VEL.(C.P.)";+CHR$(1)
3730 PRINT #3,"SP 2;PU 3380,7000;LB VELOCITY C.P.X.";+CHR$(1)
3740 PRINT #3,";PU 5800,7000;LB VELOCITY C.P.Y.";+CHR$(1)
3750 IF VAF$="1" THEN PRINT #3,";PU 8300,7000;LB VELOCITY ANGLE ";+CHR$(1)
        ELSE IF VAF$="2" THEN PRINT #3,";PU 8300,7000;LB 2nd DIFF. VSUM
";+CHR$(1)
3760 PRINT #3,"SP 1;"
3770 '
3780 ' FOLLOWING MARKS SCALE VALUES ON AXES
3790 '*VERTICAL FORCES *
3800 X=2650
3810 FOR I=0 TO 250 STEP 50
3820    PRINT #3,"PU ";X;" 1000;LB";STR$(I)+CHR$(1);";"
3830    IF I>225 THEN GOTO 3890
3840       FOR J=1 TO 5
3850           PRINT #3,"PU ";X;" 1700;PD;XT;"
3860          X=X-2450/25
3870       NEXT J
3880 NEXT I
```

```
3890 PRINT #3,"PU 200 1700;PD;XT;"
3900  '*CPX *
3910 PRINT #3,"PU 5100  850;LB";STR$(-92.5)+CHR$(1);";PU 3140 850;LB";STR$(
92.5)+CHR$(1);";"
3920 PRINT #3,"PU 7550 850;LB";STR$(-45.6)+CHR$(1);";PU 5590 850;LB";STR$(4
5.6)+CHR$(1);";"
3930 PRINT #3,"PU 9999 850;LB";STR$(-92.5)+CHR$(1);";PU 8039 850;LB";STR$(9
2.5)+CHR$(1);";"
3940 PRINT #3,"PU 7550 200;LB /3";+CHR$(1);";PU 5590 200;LB /3";+CHR$(1);";
"
3950 PRINT #3,"PU 5100 200;LB /3";+CHR$(1);";PU 3140 200;LB /3";+CHR$(1);";
"
3960 RETURN
3970 '
3980  '************************************************
3990 'PECK ROUTINE
4000 BEEP:INPUT"PRESS RETURN IF PLOTTER IS READY",POOP$
4010 PRINT #3,"IN;SC0,9999,0,9999;SP1;"
4020 PRINT #3,"DT";+CHR$(1);";DI 0,1;PU 0,2000;LB FILENAME:";E$+"  TRAVERSE
:";STR$(T%)+"   TYPE: "+TYPE$+CHR$(1);";"
4030 PRINT #3,"DT";+CHR$(1);";DI 0,1;PU 0,8000;TYPE";TYPE$+CHR$(1);";"
4040 PRINT #3,"PU 6000 10000;PD 4000 10000;"
4050 IF LEFT$(TYPE$,1)="W" OR LEFT$(TYPE$,1)="w" THEN
          PRINT #3,"PU 4120 9880;PD 4000 10000;PD 4120 10120;" ELSE
                PRINT #3,"PU 5880 9880;PD 6000 10000;PA 5880 10120;":' DRAWS
 ARROWHEADS
4060 PRINT #3,"DI 0 1;"
4070 PRINT #3,"IN;IP 1150 1825 9150 5825;SC 0.000 8.000 0.000 4.000;SP 1;PA
 0 0;PD 8 0;PA 4 4 0 0;PU 0.5 3;PD 1.5 3;PU 1 3.5;PD 1 2.5;" :' DRAWS TRIAN
GLE
4080 PRINT #3,"DT";+CHR$(1);";"
4090 PRINT #3,"DI 0,1;PU 1 2.25;LB N";+CHR$(1);";"
4100 PRINT #3,"DI 0,1;PU 1 3.5;LB S";+CHR$(1);";"
4110 PRINT #3,"DI 0,1;PU 0.4 2.875;LB E";+CHR$(1);";"
4120 PRINT #3,"DI 0,1;PU 1.65 2.875;LB W";+CHR$(1);";"
4130 PRINT #3,"SC 0.000 92.50 0.000 45.60;"
4140 FOR J=1 TO DATACNT(T%)
4150   X=CPX(J):Y=CPY(J)
4160 IF (Y=0) OR (X=0) THEN GOTO 4180: 'DON'T PLOT ZEROS
4170 PRINT #3,"PU;PA ";X;" ";Y;";PD;"
4180 NEXT J
4190 PRINT #3,"DF;DT";+CHR$(1);";DI 0,1;PU 1030 1825;LB 0,0";+CHR$(1);";PU
9270 1825;LB 92.5,0";+CHR$(1);";PU 5150 5945;LB 46.25,45.6";+CHR$(1);";PU;"
4200 PRINT #3,"PU;SP1;PU 0,0;"
4210 ' PLOTS HEEL STRIKES & TOE OFFS ON PECK
4220 IF TE$="n" OR TE$="N" THEN GOTO 4370
4230 PRINT:INPUT"DRAW TEMPORAL EVENTS ON PECK (Y/N) "; D$:IF D$<>"Y" AND D$
<>"y" AND D$<>"N"      AND D$<>"n" THEN GOTO 4230
4240 IF (D$="N" OR D$="n") OR (TE$<>"Y" AND TE$<>"y") THEN GOTO 4370
4250  PRINT #3,"IN;IP 1150 1825 9150 5825;SC 0.000 92.50 0.000 45.60;"
4260 PRINT #3,"SP 2;DI 0,1;DT";+CHR$(1);";"
4270 FOR S=1 TO 4
4280   IF STRIKE(S)=0 THEN GOTO 4350
4290   FOR P=1 TO STRIKE(S)
4300     IF S=1 THEN X=CPX(HIT(1,P)):Y=CPY(HIT(1,P)):PRINT #3,"PU ";X;" ";Y
```

```
+1;";PD ";X;" ";Y+6;";LB RH";+CHR$(1);";"
4310     IF S=2 THEN X=CPX(HIT(2,P)):Y=CPY(HIT(2,P)):PRINT #3,"PU ";X;" ";Y
-1;";PD ";X;" ";Y-6;";PU ";X;" ";Y-10;";PD;LB LH";+CHR$(1);";"
4320     IF S=3 THEN X=CPX(HIT(3,P)):Y=CPY(HIT(3,P)):PRINT #3,"PU ";X;" ";Y
+1;";PD ";X;" ";Y+3;";LB RTO";+CHR$(1);";"
4330     IF S=4 THEN X=CPX(HIT(4,P)):Y=CPY(HIT(4,P)):PRINT #3,"PU ";X;" ";Y
-1;";PD ";X;" ";Y-3;";PU ";X;" ";Y-8;";LB LTO";+CHR$(1);";"
4340   NEXT P
4350 NEXT S
4360 PRINT #3,"PU;SP1;PU 0,0;"
4370 GOTO 1910: 'MAIN MENU
4380 ' FORCES PLOT FOLLOWS ******************************************
4390 INPUT"SELECT  (1) ALL FORCES PLOT   (2) VERTICAL ONLY PLOT  ";PFLAG
4400 IF PFLAG<>1 AND PFLAG<>2 THEN 4390: 'TRAP
4410 BEEP:INPUT"If the plotter is ready press [ENTER] key ";Z$
4420 PRINT #3,"IN;SC0,9999,0,9999;SP1;"
4430 PRINT #3,"DT";+CHR$(1);";DI 0,1;PU 0,2000;LB FILENAME:";E$+"  TRAVERSE
:";STR$(T%)+"   TYPE: "+TYPE$+CHR$(1);";"
4440 PRINT #3,"DT";+CHR$(1);";DI 0,1:PU 700,6000;LB VERTICAL FORCE";+CHR$(1
);";"
4450 IF PFLAG=2 THEN QQ=8:GOTO 4470 ELSE QQ=24
4460 PRINT #3,";PU 3700,6000;LB MEDIO LATERAL FORCES";+CHR$(1);";PU 6700,56
00;LB ANTERIOR POSTERIOR FORCES";+CHR$(1);";"
4470 FOR I= 1 TO QQ STEP 4 'DRAWS AXES FOR FORCE PLOT
4480   X=INT(AXES2(I))
4490   Y=INT(AXES2(I+1))
4500   PRINT #3,"LT;PU",X,",",Y,";PD;"
4510   X=INT(AXES2(I+2))
4520   Y=INT(AXES2(I+3))
4530   PRINT #3,"PA ";X;" ";Y;";";
4540 NEXT I
4550 '*VERTICAL FORCES *******************  MAKES TICK MARKS
4560 X=3300
4570 FOR I=0 TO 250 STEP 50
4580   PRINT #3,"PU ";X;" 1000;LB";STR$(I)+CHR$(1);";"
4590   IF I>225 THEN GOTO 4650
4600   FOR J=1 TO 5
4610      PRINT #3,"PU ";X;" 1700;PD;XT;"
4620      X=X-(3300-700)/25
4630   NEXT J
4640 NEXT I
4650 PRINT #3,"PU 700 1700;PD;XT;"
4660 IF PFLAG=2 THEN 4830
4670  '* MEDIO LATERAL *
4680 X=6300
4690 PRINT #3,"PU 6300 1000;LB";STR$(-40)+CHR$(1);";PU 3700 1000;LB";STR$(4
0)+CHR$(1);";"
4700 PRINT #3,"DT";+CHR$(1);";DI 0,1;PU 0,8000;TYPE";TYPE$+CHR$(1);";"
4710 FOR I=0 TO 16
4720    PRINT #3,"PU ";X;" 1700;PD;XT;"
4730    X=X-(6300-3700)/16 : 'MARKS EVERY 10 LBS.
4740 NEXT I
4750 '* ANTERIOR POSTERIOR *
4760 X=9300
4770 PRINT #3,"PU 9300 1000;LB";STR$(-40)+CHR$(1);";PU 6700 1000;LB";STR$(4
```

```
0)+CHR$(1);";"
4780 FOR I=0 TO 16
4790     PRINT #3,"PU ";X;" 1700;PD;XT;"
4800     X=X-(9300-6700)/16 : 'MARKS EVERY 10 LBS.
4810 NEXT I
4820 '
4830 ' PLOTTING ROUTINE FOR FORCE
4840 Y=1700
4850 PRINT #3,"PU;SP2;PU 3300,",Y,";PD;"
4860 INCR=8200/DATACNT(T%)
4870 VSCALE2=(3300-700)/250
4880 FOR I= 1 TO DATACNT(T%)-1
4890     Y=Y+INCR
4900     X=3300-VSUM(I)*VSCALE2
4910     PRINT #3,"PA ";X;" ";Y;";";
4920 NEXT I
4930 IF PFLAG=2 THEN 5080
4940 '
4950 POSI = 5000
4960 FOR GR =2 TO 3
4970     Y=1700
4980 SCMLPA=(5000-3700)/40
4990 PRINT #3,"PU",POSI,",",Y,";PD;"
5000 FOR I=1 TO DATACNT(T%)
5010   Y=Y+INCR
5020   IF GR = 2 THEN X=POSI-(DATAPTS(I,4)+DATAPTS(I,5))*SCMLPA
5030   IF GR = 3 THEN X=POSI-DATAPTS(I,6)*SCMLPA
5040   PRINT #3,"PA ";X;" ";Y;";";
5050   NEXT I
5060   POSI=POSI+3000
5070 NEXT GR
5080 PRINT #3,"PU;SP1;PU 0,0;"
5090 'FOLLOWING DRAWS TEMPORAL LINES (IF GIVEN ) ON FORCE PLOT
5100 IF TE$="n" OR TE$="N" THEN GOTO 5240
5110 PRINT:INPUT"DRAW TEMPORAL EVENTS (Y/N) "; D3$:IF D3$<>"Y" AND D3$<>"y"
 AND       D3$<> "N" AND D3$<>"n" THEN GOTO 5110 :'TRAP
5120 IF D3$="N" OR D3$="n" THEN GOTO 5240
5130 PRINT #3,"DI 0,1;SI .15 .2;DT";+CHR$(1);";"
5140 FOR S=1 TO 4
5150   IF STRIKE(S)=0 THEN GOTO 5220' NEXT STRIKE
5160   FOR P=1 TO STRIKE(S)
5170      IF S=1 THEN X=3300-VSCALE2*VSUM(HIT(1,P)):Y=1700+HIT(1,P)*8200/DAT
ACNT(T%):PRINT #3,"PU ";3425;" ";Y-190;";PD;LB RH";+CHR$(1);";PU ";X;" ";Y;
";PD 9999 ";Y;";PU;"
5180      IF S=2 THEN X=3300-VSCALE2*VSUM(HIT(2,P)):Y=1700+HIT(2,P)*8200/DAT
ACNT(T%):PRINT #3,"PU ";3425;" ";Y-190;";PD;LB LH";+CHR$(1);";PU ";X;" ";Y;
";PD 9999 ";Y;";PU;"
5190      IF S=3 THEN X=3300-VSCALE2*VSUM(HIT(3,P)):Y=1700+HIT(3,P)*8200/DAT
ACNT(T%):PRINT #3,"PU ";3550;" ";Y-250;";PD;LB RTO";+CHR$(1);";PU ";X;" ";Y
;";PD 9999 ";Y;";PU;"
5200      IF S=4 THEN X=3300-VSCALE2*VSUM(HIT(4,P)):Y=1700+HIT(4,P)*8200/DAT
ACNT(T%):PRINT #3,"PU ";3550;" ";Y-250;";PD;LB LTO";+CHR$(1);";PU ";X;" ";Y
;";PD 9999 ";Y;";PU;"
5210   NEXT P
5220 NEXT S
```

```
5230 PRINT #3,"PU 0 0;"
5240 CLS:INPUT "SELECT (T)TEMPORAL SHIFT   (F)FRACTIONAL SPLIT  (M) MAIN ME
NU ",        DI$
5250 IF DI$="T" OR DI$="t" THEN 5260 ELSE IF DI$="F" OR DI$="f" THEN 5970
        ELSE IF DI$="M" OR DI$="m" THEN 1890 ELSE 5240
5260 ' BEGINNING OF TEMPORAL VSUM SHIFT
5270 'ROUTINE TO SHIFT OVER TRAILING EDGE
5280 PRINT:INPUT"SOURCE POINT (HS) ";SHS:PRINT:INPUT"DESTINATION POINT (HS)
 ";DHS
5290 PRINT:PRINT"DELTA HS (VSUM(DHS)-VSUM(SHS))= "VSUM(DHS)-VSUM(SHS)
5300 PRINT:PRINT"DO YOU WISH TO ADJUST THE SCALE TO MATCH VALUES (Y/N)?"
5310 FFA$=INKEY$:IF FFA$="" THEN 5310
5320 IF FFA$<>"y" AND FFA$<>"Y" AND FFA$<>"N" AND FFA$<>"n" THEN 5310:'TRAP
5330 IF FFA$="y" OR FFA$="Y" THEN FSC=VSUM(DHS)/VSUM(SHS) ELSE FSC=1
5340 INCR=8200/DATACNT(T%):VSCALE2=(3300-700)/250
5350 Y=1700+DHS*INCR:X=3300-VSUM(SHS)*VSCALE2*FSC
5360 PRINT #3,"SP1;LT 6;PU ";X;" ";Y;";PD;"
5370 FOR I=SHS+1 TO SHS+20
5380    Y=Y+INCR
5390    X=3300-VSUM(I)*VSCALE2*FSC
5400    PRINT #3,"PD ";X;" ";Y;";"
5410 NEXT I
5420 IF PFLAG=2 THEN 5580
5430 ' FOLLOWING FOR TEMP SHIFT ON M/L AND P/A
5440 Y=1700+DEST*INCR:X=5000-(DATAPTS(SOURCE,4)+DATAPTS(SOURCE,5))*SCMLPA
5450 PRINT #3,"PU ";X;" ";Y;";PD;"
5460 FOR I=SOURCE+1 TO DATACNT(T%)
5470    Y=Y+INCR
5480    X=5000-(DATAPTS(I,4)+DATAPTS(I,5))*SCMLPA
5490    PRINT #3,"PD ";X;" ";Y;";"
5500 NEXT I
5510 Y=1700+DEST*INCR:X=8000-DATAPTS(SOURCE,6)*SCMLPA
5520 PRINT #3,"PU ";X;" ";Y;";PD;"
5530 FOR I=SOURCE+1 TO DATACNT(T%)
5540    Y=Y+INCR
5550    X=8000-DATAPTS(I,6)*SCMLPA
5560    PRINT #3,"PD ";X;" ";Y;";"
5570 NEXT I
5580 'ROUTINE TO SUBTRACTIVELY DETERMINE AND PLOT LEADING EDGE OTHER FOOT
5590 INCR=8200/DATACNT(T%):VSCALE2=(3300-700)/250
5600 Y=1700+DHS*INCR:X=3300-(VSUM(DHS)*VSCALE2-VSUM(SHS)*VSCALE2*FSC)
5610 PRINT #3,"SP1;LT 6;PU ";X;" ";Y;";PD;"
5620 FOR I=1 TO 20
5630    Y=Y+INCR
5640    X=3300-(VSUM(DHS+I)*VSCALE2-VSUM(SHS+I)*VSCALE2*FSC)
5650    PRINT #3,"PD ";X;" ";Y;";"
5660 NEXT I
5670 PRINT #3,"PU;SP1;PU 0,0;LT;"
5680 PRINT:INPUT"DO YOU WANT A FORWARD SHIFT OF LEAD FOOT ? ";V$:IF V$="N"
OR V$="n" THEN GOTO 5940 ELSE IF V$<>"Y" AND V$<>"y" THEN GOTO 5680
5690 PRINT:INPUT"SOURCE POINT (TOE-OFF) ";STOFF:PRINT:INPUT"DESTINATION POI
NT (TOE-OFF) ";DTOFF
5700 PRINT:PRINT"DELTA TOFF (VSUM(DTOFF)-VSUM(STOFF))= "VSUM(DTOFF)-VSUM(ST
OFF)
5710 PRINT:PRINT"DO YOU WISH TO ADJUST THE SCALE TO MATCH VALUES (Y/N)?"
```

```
5720 FFA$=INKEY$:IF FFA$="" THEN 5720
5730 IF FFA$<>"y" AND FFA$<>"Y" AND FFA$<>"N" AND FFA$<>"n" THEN 5720:'TRAP
5740 IF FFA$="y" OR FFA$="Y" THEN FSC2=VSUM(DTOFF)/VSUM(STOFF) ELSE FSC2=1
5750 INCR=8200/DATACNT(T%):VSCALE2=(3300-700)/250
5760 ' PLOTS LEADING EDGE IN NEW LOCATION
5770 Y=1700+DTOFF*INCR:X=3300-VSUM(STOFF)*VSCALE2*FSC2
5780 PRINT #3,"SP1;LT 6;PU ";X;" ";Y;";PD;"
5790 FOR I=1 TO 20 :'PLOTS BACKWARDS
5800    Y=Y-INCR: 'BACKWARDS
5810    X=3300-VSUM(STOFF-I)*VSCALE2*FSC2
5820    PRINT #3,"PD ";X;" ";Y;";"
5830 NEXT I
5840 'PRINT #3,"PU;SP1;PU 0,0;LT;"
5850 ' FOLLOWING SUBTRACTIVELY DETERMINES AND PLOTS THE TRAILING EDGE
5860 Y=1700+DTOFF*INCR:X=3300-(VSUM(DTOFF)*VSCALE2-VSUM(STOFF)*VSCALE2*FSC2
)
5870 PRINT #3,"SP1;LT 6;PU ";X;" ";Y;";PD;"
5880 FOR I=1 TO 20  :'PLOTS BACKWARDS
5890    Y=Y-INCR: 'BACKWARDS
5900    X=3300-(VSUM(DTOFF-I)*VSCALE2-VSUM(STOFF-I)*VSCALE2*FSC2)
5910    PRINT #3,"PD ";X;" ";Y;";"
5920 NEXT I
5930 PRINT #3,"PU;SP1;PU 0,0;LT;"
5940 GOTO 1890 :'MAIN MENU
5950 '@@@@@@@@@ END OF TEMPORAL SHIFT
5960 '
5970 '********* FRACTIONAL SPLIT DETERMINATION OF VERT FORCES*****
5980 ' TO BEST UNDERSTAND THE FOLLOWING, SEE LOG NOTES AUG 5 ENTRY
5990 INPUT"GIVE ME H.S AND TOE OFF TIMES ",HS,TOF
6000 HSCPX=CPX(HS):TOFCPX=CPX(TOF):'CPX @ HS, AND @ TOF
6010 TFVCPX=ABS(CPX(HS-2)-CPX(HS-12))/10: 'VEL OF CPX FOR TRAILING FOOT
6020 'ABOVE ASSUMES  BOTH VELOCITIES AS CONSTANT THRU THE DUAL SUPPORT PHAS
E
6030 '
6040 LFVCPX=ABS(CPX(TOF+12)-CPX(TOF+2))/10: 'VEL OF CPX FOR LEAD FOOT
6050 INCR=8200/DATACNT(T%):VSCALE2=(3300-700)/250
6060 'ROUTINE TO PLOT LEADING EDGE OF LEAD FOOT
6070 Y=1700+HS*INCR:X=3300 : 'STARTS AT ZERO FORCE
6080 PRINT #3,"SP1;LT 6;PU ";X;" ";Y;";PD;"
6090 FOR I=HS+1 TO TOF
6100    Y=Y+INCR
6110    NUM=ABS(CPX(I)-HSCPX)-ABS(TFVCPX*(I-HS)): 'THIS IS 'A ' ON MY FIGUR
E
6120    DEN=NUM+ABS(TOFCPX-CPX(I))-ABS(LFVCPX*(I-HS)): ' 'B' ON MY FIG
6130    RLLF=NUM/DEN : 'RELATIVE LOADING OF LEAD FOOT
6140    ALLF=RLLF*VSUM(I): 'ABSOLUTE LOADING OF THE LEAD FOOT
6150    ALTF(I)=VSUM(I)-ALLF:'ABSOLUTE LOAD ON TRAILING FOOT
6160    X=3300- ALLF*VSCALE2
6170    PRINT #3,"PD ";X;" ";Y;";"
6180 NEXT I
6190 'ROUTINE TO PLOT TRAILING EDGE OF TRAILIMG FOOT
6200 Y=1700+HS*INCR:X=3300-VSUM(HS)*VSCALE2
6210 PRINT #3,"SP1;LT 6;PU ";X;" ";Y;";PD;"
6220 FOR I=HS+1 TO TOF
6230    Y=Y+INCR
```

```
6240    X=3300-ALTF(I)*VSCALE2
6250    PRINT #3,"PD ";X;" ";Y;";"
6260 NEXT I
6270 PRINT #3,"PU;SP1;PU 0,0;LT;"
6280 GOTO 1900: 'MAIN MENU
6290 '************SCREEN PRINT ROUTINE*******
6300 CLS
6310 PRINT TAB(2)"PT"TAB(9)"L"TAB(15)"R"TAB(20)"VSUM"TAB(29)"RVCP"TAB(38)"S
DVSUM"
6320 FOR Q%=1 TO DATACNT(T%)
6330  IF Q%=1 OR ((Q%-1)/20 -INT((Q%-1)/20))<>0 THEN 6370
          :'CHECKS FOR MULTIPLES OF 20
6340  LOCATE 23,30:PRINT"HIT ANY KEY TO CONTINUE"
6350  Z$=INKEY$:IF Z$="" THEN 6350
6360  CLS
6370  PRINT TAB(2)Q%TAB(8)DATAPTS(Q%,7)TAB(14)DATAPTS(Q%,8);:
          PRINT USING"####.##  ";VSUM(Q%),RVCP(Q%),SDVSUM(Q%)
6380 NEXT Q%
6390 LOCATE 22,30:PRINT" (H) TO FIND DELTA HUMP FORCE"
6400 LOCATE 23,30:PRINT"  HIT ANY OTHER KEY TO CONTINUE"
6410 Z$=INKEY$:IF Z$="" THEN 6410
6420 CLS
6430 IF Z$<>"H" AND Z$<>"h" THEN 6490
6440 PRINT "GIVE ME THE LAST, AND FIRST H.S OF THE SAME FOOT"
6450 INPUT LHS,FHS
6460 DELT=VSUM(LHS)-VSUM(FHS)
6470 PRINT "DELTA HUMP FORCE IS LAST-FIRST"
6480 PRINT "DELTA = "VSUM(LHS)"-"VSUM(FHS)" = "DELT:PRINT:GOTO 6390
6490 GOTO 1910:'MAIN MENU
6500 ' ******** ROUTINE TO DUMP EXTRACTED RECORD TO FILE
6510 CLS
6520 RN$=LEFT$(E$,6):RN$=RN$+"-"+RIGHT$(E$,1)+".EX"+RIGHT$(STR$(T%),1)
6530 ' SAMPLE FILE NAME KM2406-1.EX (YEAR OMITTED, -T%, .EX FOR EXTRACTED)
6540 PRINT "USE RECORD NAME:  "RN$ " (Y/N)?"
6550 AN$=INKEY$:IF AN$="" THEN 6550
6560 IF AN$="Y" OR AN$="y" THEN 6580 ELSE IF AN$="N" OR AN$="n" THEN 6570
         ELSE 6550
6570 INPUT "GIVE ME THE NEW NAME ",NN$
6580 'OPEN FILE AND DUMP UNDER FILENAME
6590 '@@@@@@@@@@@@@@@ END BLOCK 2@@@@@@@@@@@@@@@@
```

```
10    CLS
20    SWITCH=200
30    RN=0
40    DIM DATACNT(384),DATAPTS(384,8),DCOFF(9),CALDATA(384,6),VSUM(384),
          CPX(384),CPY(384),VERT(6),CALIB(9),RN$(30),HS1%(30),HS2%(30),HS1
VSUM(30),      HS2VSUM(30),DELTA(30)
50    GOSUB 500        '   subroutine to load in CALIB(I) factors
60    INPUT "Insert desired data disk (unedited) in drive b.",C%
70   PRINT:PRINT
80   PRINT "Available files:  "
90   PRINT: FILES "B:*.*"
100   PRINT
110   INPUT "Desired data file  ",N$
120   F$="B:"+N$                'Ensure drive B.
130   CLS
140   IF T%=0 THEN GOSUB 690 :'Get traverse counters.ONLY ONCE PER FILE
150   PRINT:INPUT "SELECT A TRAVERSE  ";T%
160   PRINT:PRINT "Data points taken for traverse";T%;"=";DATACNT(T%)
170   GOSUB 860       '  get one traverse worth of data.
180  ' NOW HAS ONE TRAVERSE OF DATA
190   RN=RN+1:RN$(RN)=N$+STR$(T%):' GIVE THE RECORD A NAME
200 ' PRINT:INPUT "SELECT FIRST OR LAST DATAPTS FOR D.C. OFFSET CORR. (F/L)
 ";FL$
210 ' IF FL$<>"f" AND FL$<>"F" AND FL$<>"L" AND FL$<>"l" THEN 190:'TRAP
220   FOR H%=1 TO 8             'Loop through 3 channels.
230      IF H%>3 AND H%<7 THEN 430
240      SUM=0
250      IF DATACNT(T%)>381 THEN PRINT " DATACNT >381 ** NO D.C CORR. APPLI
ED"          :GOTO 330
260 '    LST%=DATACNT(T%)
270  '   IF FL$="L" OR FL$="l" THEN FOR X=1 TO 3:SUM=DATAPTS(LST%-X,H%)
             +SUM:NEXT X
280 '    IF FL$="F" OR FL$="f" THEN FOR X=1 TO 3:SUM=DATAPTS(X,H%)+SUM:NEXT
 X
290      FOR X=1 TO 3:SUM=DATAPTS(X,H%)+SUM:NEXT X
300     'The above x driven loop gets first or last 3 pts of traverse for D
.C.
310     'offset purposes.
320      SUM=SUM/3          'Average D.C. offset.
330      FOR Q%=1 TO DATACNT(T%)
340        IF H%<7 THEN 370
350        IF DATAPTS(Q%,H%)>SWITCH THEN DATAPTS(Q%,H%)=1 ELSE DATAPTS(Q%,H
%)=0
360        GOTO 410
370        DAT=DATAPTS(Q%,H%)      'GET DATAPTS
380        DAT=DAT-SUM    'CORRECT DC OFFSET
390        DAT=DAT/CALIB(H%)
400        DATAPTS(Q%,H%)=DAT
410       NEXT Q%
420       PRINT "CHANNEL ";H%;" EDITED"
430   NEXT H%           'Next channel.
440   REM        At this point one complete traverse has been edited.
450   PRINT:PRINT"RECORD  "RN$(RN)"  HAS NOW BEEN EDITED "
460   GOTO 1090 : 'BLOCK 2
470 '
```

```
480 '***** BLOCK 1 SUBROUTINES FOLLOW *******************
490 '
500 '****SUBROUTINE TO READ IN CALIB(I) FACTORS FROM CALROSS ********
510    REM
520    PRINT "Please enter calibration disk into drive B."
530    INPUT "Strike [ENTER] when ready...",CRAP%
540    PRINT:FILES "B:*.*"
550    PRINT:INPUT "SELECT FILE (without filename extension ASSUMES NO 'R '
):",CAL$
560    CAL$="B:"+CAL$ +" RCAL"  'ENSURES CALROSS FILES ONLY USED
570    OPEN CAL$ FOR INPUT AS #1
580    CLS:PRINT "CALIBRATION FACTORS READ FROM FILE":PRINT
590    FOR I=1 TO 9
600    INPUT#1,CALIB(I)
610    PRINT CALIB(I)
620    NEXT I
630    INPUT "CONTINUE WITH EDIT (Y/N)";ANS$
640    IF ANS$="N" OR  ANS$="n" THEN END
650    CLOSE#1
660    RETURN
670 '
680 '
690 'SUBROUTINE  IS USED TO GET THE TRAVERSE COUNTERS.
700     OPEN F$ AS #1 LEN=768
710     FIELD #1,192 AS A$,192 AS B$,192 AS C$,192 AS D$
720     REC=49  'Last record contains desired data.
730     GET #1,REC      'Read data.
740     FOR I=1 TO 96
750       K=I*2-1
760       DATACNT(I)=CVI(MID$(A$,K,2))
770       DATACNT(I+96)=CVI(MID$(B$,K,2))
780       DATACNT(I+192)=CVI(MID$(C$,K,2))
790       DATACNT(I+288)=CVI(MID$(D$,K,2))
800NEXT I
810 'DATACNT NOW IS PRIMED WITH TRAVERSE COUNTERS.
820 'FILE IS LEFT OPEN, AS DATA WILL BE READ IN MAIN-LINE.
830RETURN'Back to mainline.
840   REMEND OF SUBROUTINE
850 '
860REM Subroutine for reading unedited data.
870 'FILE IS ALREADY OPEN.
880 '
890    FOR CH%=1 TO 8          'Loop through  channels 1,2,3,7,8
900     IF CH%>3 AND CH%<7 THEN 1040
910REC=(CH%+(T%-1)*8)
920     GET #1,REC
930     PRINT CH%;" READ.."
940FOR I=1 TO 96
950K=I*2-1
960DATAPTS(I,CH%)=CVI(MID$(A$,K,2))
970DATAPTS(I+96,CH%)=CVI(MID$(B$,K,2))
980DATAPTS(I+192,CH%)=CVI(MID$(C$,K,2))
990DATAPTS(I+288,CH%)=CVI(MID$(D$,K,2))
1000NEXT I
1010FOR I=1 TO 384
```

```
1020      IF DATAPTS(I,CH%) > 32767 THEN DATAPTS(I,CH%)=DATAPTS(I,CH%)-65536
!
1030NEXT I
1040NEXT CH%
1050PRINT "TRAVERSE ";T%;" READ"
1060RETURN'Back to mainline.
1070    REMEND OF SUBROUTINE
1080 '
1090  '******* END OF BLOCK 1 (EDITING SECTION) **************
1100  '****** START BLOCK 2 *********
1110  INPUT "DO YOU WANT TO SEE THE DATA (Y/N) ";SDA$
1120  IF SDA$="y" OR SDA$="Y" THEN GOSUB 1560
1130  PRINT "GIVE ME THE HEEL STRIKE TIMES (FIRST,LAST) FOR RECORD: " RN$(R
N)
1140  INPUT HS1%(RN),HS2%(RN)
1150  HS1VSUM(RN)=DATAPTS(HS1%(RN),1)+DATAPTS(HS1%(RN),2)+DATAPTS(HS1%(RN),
3)
1160  HS2VSUM(RN)=DATAPTS(HS2%(RN),1)+DATAPTS(HS2%(RN),2)+DATAPTS(HS2%(RN),
3)
1170  DELTA(RN)=HS2VSUM(RN)-HS1VSUM(RN)
1180  'MENU FOLLOWS
1190  PRINT"SELECT ONE OF:   (1) GET NEW TRAVERSE  (2) GET NEW FILE
                              (3) SCREEN PRINTOUT   (4) FIND STATS FOR
HSVSUMS"
1200  INPUT CD$
1210  IF CD$="1" THEN CLS:GOTO 150 :'NEW TRAVERSE
1220  IF CD$="2" THEN CLS:T%=0:CLOSE#1: GOTO 80  :'NEW FILE
1230  IF CD$="3" THEN GOTO 1270
1240  IF CD$="4" THEN GOTO 1340
1250  GOTO 1190: 'TRAP
1260 '
1270 ' SCREEN DUMP ROUTINE
1280 CLS
1290 PRINT"RECORD"TAB(15)"HSTRIKES"TAB(25)"HS1VSUM"TAB(35)"HS2VSUM"TAB(46)"
DELTA"
1300 PRINT"-----------------------------------------------------------"
1310 FOR Z=1 TO RN :PRINT RN$(Z)TAB(14)HS1%(Z)TAB(19)HS2%(Z)TAB(24)HS1VSUM(
Z)TAB(34)HS2VSUM(Z)TAB(45)DELTA(Z):NEXT Z
1320 PRINT "HIT ANY KEY TO CONTINUE":INPUT CRAP$
1330 CLS:GOTO 1190:'MENU
1340 'STATS ROUITINE
1350 'FIND MEAN DELTA
1360 ZZ=0
1370 FOR Q=1 TO RN
1380  ZZ=ZZ+DELTA(Q)
1390 NEXT Q
1400 MEAN=ZZ/RN
1410 ' FIND SIGMA
1420 SMSQ=0
1430 FOR W=1 TO RN
1440  DEV=DELTA(W)-MEAN:DEV=DEV^2
1450  SMSQ=SMSQ+DEV
1460 NEXT W
1470 SIGMA=(SMSQ/RN)^.5
1480 PRINT:PRINT "MEAN DELTA= " MEAN "SIGMA =" SIGMA
```

```
1490 INPUT "DO YOU WANT A PRINTOUT (Y/N) (make sure printer is on)";PO$
1500 IF PO$="y" OR PO$="Y" THEN 1510 ELSE CLS:GOTO 1190
1510 LPRINT"RECORD"TAB(16)"HSTRIKES"TAB(26)"HS1VSUM"TAB(36)"HS2VSUM"TAB(46)
"DELTA"
1520 LPRINT"--------------------------------------------------------"
1530 FOR Z=1 TO RN :LPRINT RN$(Z)TAB(15)HS1%(Z)TAB(20)HS2%(Z)TAB(25)HS1VSUM
(Z)TAB(35)HS2VSUM(Z)TAB(45)DELTA(Z):NEXT Z
1540 LPRINT:LPRINT "MEAN DELTA = " MEAN "  SIGMA =" SIGMA
1550 CLS:GOTO 1190:'MENU
1560 ' SCREEN PRINT OF EDITED DATA
1570 CLS
1580 PRINT TAB(2)"PT"TAB(9)"L"TAB(15)"R"TAB(20)"VSUM"
1590 FOR Q%=1 TO DATACNT(T%)
1600 IF (Q%/20-INT(Q%/20))<>0 THEN 1640
1610 LOCATE 23,30:PRINT"HIT ANY KEY TO CONTINUE"
1620 Z$=INKEY$:IF Z$="" THEN 1620
1630 CLS
1640 PRINT TAB(2)Q%TAB(8)DATAPTS(Q%,7)TAB(14)DATAPTS(Q%,8);:PRINT USING"###
#.##"; DATAPTS(Q%,1)+DATAPTS(Q%,2)+DATAPTS(Q%,3)
1650 NEXT Q%
1660 LOCATE 23,30 :PRINT "HIT ANY KEY TO CONTINUE OR (R) TO REPEAT"
1670 Z$=INKEY$:IF Z$="" THEN 1670
1680 IF Z$="R" OR Z$="r" THEN 1570: 'REPEAT
1690 CLS
1700 RETURN
```

```
10    CLS
15    PRINT:INPUT "SELECT FIRST OR LAST DATAPTS FOR D.C. OFFSET CORR. (F/L)
 ";FL$
20    IF FL$<>"f" AND FL$<>"F" AND FL$<>"L" AND FL$<>"l" THEN 15 :'TRAP
30    RN=0
40    DIM DATACNT(384),DATAPTS(384,8),DCOFF(9),CALDATA(384,6),VSUM(384),
          CPX(384),CPY(384),VERT(6),CALIB(9),RN$(30),HS%(30),HSVSUM(30)
50    GOSUB 500        '   subroutine to load in CALIB(I) factors
60    INPUT "Insert desired data disk (unedited) in drive b.",C%
70   PRINT:PRINT
80   PRINT "Available files:  "
90   PRINT: FILES "B:*.*"
100   PRINT
110   INPUT "Desired data file  ",N$
120   F$="B:"+N$                'Ensure drive B.
130   CLS
140   IF T%=0 THEN GOSUB 690 :'Get traverse counters.ONLY ONCE PER FILE
150   PRINT:INPUT "SELECT A TRAVERSE  ";T%
160   PRINT:PRINT "Data points taken for traverse";T%;"=";DATACNT(T%)
170   GOSUB 860        '  get one traverse worth of data.
180  ' NOW HAS ONE TRAVERSE OF DATA
190   RN=RN+1:RN$(RN)=N$+STR$(T%):' GIVE THE RECORD A NAME
200 ' PRINT:INPUT "SELECT FIRST OR LAST DATAPTS FOR D.C. OFFSET CORR. (F/L)
 ";FL$
210 ' IF FL$<>"f" AND FL$<>"F" AND FL$<>"L" AND FL$<>"l" THEN 200:'TRAP
220   FOR H%=1 TO 8              'Loop through 3 channels.
230      IF H%>3 AND H%<7 THEN 430
240      SUM=0
250      IF DATACNT(T%)>381 THEN PRINT " DATACNT >381 ** NO D.C CORR. APPLI
ED"          :GOTO 330
260 '    LST%=DATACNT(T%)
270  '   IF FL$="L" OR FL$="l" THEN FOR X=1 TO 3:SUM=DATAPTS(LST%-X,H%)
            +SUM:NEXT X
280 '    IF FL$="F" OR FL$="f" THEN FOR X=1 TO 3:SUM=DATAPTS(X,H%)+SUM:NEXT
 X
290      FOR X=1 TO 3:SUM=DATAPTS(X,H%)+SUM:NEXT X
300     'The above x driven loop gets first or last 3 pts of traverse for D
.C.
310     'offset purposes.
320      SUM=SUM/3        'Average D.C. offset.
330      FOR Q%=1 TO DATACNT(T%)
340        IF H%<7 THEN 370
350        IF DATAPTS(Q%,H%)>SWITCH THEN DATAPTS(Q%,H%)=1 ELSE DATAPTS(Q%,H
%)=0
360        GOTO 410
370        DAT=DATAPTS(Q%,H%)       'GET DATAPTS
380        DAT=DAT-SUM     'CORRECT DC OFFSET
390        DAT=DAT/CALIB(H%)
400        DATAPTS(Q%,H%)=DAT
410       NEXT Q%
420       PRINT "CHANNEL ";H%;" EDITED"
430   NEXT H%           'Next channel.
440   REM        At this point one complete traverse has been edited.
450   PRINT:PRINT"RECORD  "RN$(RN)"  HAS NOW BEEN EDITED "
460   GOTO 1090 : 'BLOCK 2
```

```
470 '
480 '***** BLOCK 1 SUBROUTINES FOLLOW *******************
490 '
500 '****SUBROUTINE TO READ IN CALIB(I) FACTORS FROM CALROSS ********
510     REM
520     PRINT "Please enter calibration disk into drive B."
530     INPUT "Strike [ENTER] when ready...",CRAP%
540     PRINT:FILES "B:*.*"
550     PRINT:INPUT "SELECT FILE (without filename extension ASSUMES NO 'R '
):",CAL$
560     CAL$="B:"+CAL$ +" RCAL"  'ENSURES CALROSS FILES ONLY USED
570     OPEN CAL$ FOR INPUT AS #1
580     CLS:PRINT "CALIBRATION FACTORS READ FROM FILE":PRINT
590     FOR I=1 TO 9
600     INPUT#1,CALIB(I)
610     PRINT CALIB(I)
620     NEXT I
630     INPUT "CONTINUE WITH EDIT (Y/N)";ANS$
640     IF ANS$="N" OR  ANS$="n" THEN END
650     CLOSE#1
660     RETURN
670 '
680 '
690 'SUBROUTINE  IS USED TO GET THE TRAVERSE COUNTERS.
700      OPEN F$ AS #1 LEN=768
710      FIELD #1,192 AS A$,192 AS B$,192 AS C$,192 AS D$
720      REC=49  'Last record contains desired data.
730      GET #1,REC       'Read data.
740      FOR I=1 TO 96
750        K=I*2-1
760        DATACNT(I)=CVI(MID$(A$,K,2))
770        DATACNT(I+96)=CVI(MID$(B$,K,2))
780        DATACNT(I+192)=CVI(MID$(C$,K,2))
790        DATACNT(I+288)=CVI(MID$(D$,K,2))
800NEXT I
810 'DATACNT NOW IS PRIMED WITH TRAVERSE COUNTERS.
820 'FILE IS LEFT OPEN, AS DATA WILL BE READ IN MAIN-LINE.
830RETURN'Back to mainline.
840   REMEND OF SUBROUTINE
850 '
860REM Subroutine for reading unedited data.
870 'FILE IS ALREADY OPEN.
880 '
890    FOR CH%=1 TO 8          'Loop through  channels 1,2,3,7,8
900     IF CH%>3 AND CH%<7 THEN 1040
910REC=(CH%+(T%-1)*8)
920     GET #1,REC
930     PRINT CH%;" READ.."
940FOR I=1 TO 96
950K=I*2-1
960DATAPTS(I,CH%)=CVI(MID$(A$,K,2))
970DATAPTS(I+96,CH%)=CVI(MID$(B$,K,2))
980DATAPTS(I+192,CH%)=CVI(MID$(C$,K,2))
990DATAPTS(I+288,CH%)=CVI(MID$(D$,K,2))
1000NEXT I
```

```
1010FOR I=1 TO 384
1020      IF DATAPTS(I,CH%) > 32767 THEN DATAPTS(I,CH%)=DATAPTS(I,CH%)-65536
!
1030NEXT I
1040NEXT CH%
1050PRINT "TRAVERSE ";T%;" READ"
1060RETURN'Back to mainline.
1070   REMEND OF SUBROUTINE
1080 '
1090   '******* END OF BLOCK 1 (EDITING SECTION) **************
1100   '****** START BLOCK 2 *********
1110  INPUT "DO YOU WANT TO SEE THE DATA (Y/N) ";SDA$
1120  IF SDA$="y" OR SDA$="Y" THEN GOSUB 1540
1130  PRINT "GIVE ME THE HEEL STRIKE TIME FOR RECORD " RN$(RN)
1140  INPUT HS%(RN)
1150  HSVSUM(RN)=DATAPTS(HS%(RN),1)+DATAPTS(HS%(RN),2)+DATAPTS(HS%(RN),3)
1160  'MENU FOLLOWS
1170  PRINT"SELECT ONE OF:    (1) GET NEW TRAVERSE  (2) GET NEW FILE
                                  (3) SCREEN PRINTOUT   (4) FIND STATS FOR
HSVSUMS"
1180  INPUT CD$
1190  IF CD$="1" THEN CLS:GOTO 150 :'NEW TRAVERSE
1200  IF CD$="2" THEN CLS:T%=0:CLOSE#1: GOTO 80  :'NEW FILE
1210  IF CD$="3" THEN GOTO 1250
1220  IF CD$="4" THEN GOTO 1320
1230  GOTO 1170: 'TRAP
1240 '
1250 ' SCREEN DUMP ROUTINE
1260 CLS
1270 PRINT"RECORD","HEELSTRIKE","HSVSUM"
1280 PRINT"------","----------","------"
1290 FOR Z=1 TO RN :PRINT RN$(Z),HS%(Z),HSVSUM(Z):NEXT Z
1300 PRINT "HIT ANY KEY TO CONTINUE":INPUT CRAP$
1310 CLS:GOTO 1170:'MENU
1320 'STATS ROUITINE
1330 'FIND MEAN HSVUM
1340 ZZ=0
1350 FOR Q=1 TO RN
1360  ZZ=ZZ+HSVSUM(Q)
1370 NEXT Q
1380 MEAN=ZZ/RN
1390 ' FIND SIGMA
1400 SMSQ=0
1410 FOR W=1 TO RN
1420  DEV=HSVSUM(W)-MEAN:DEV=DEV^2
1430  SMSQ=SMSQ+DEV
1440 NEXT W
1450 SIGMA=(SMSQ/RN)^.5
1460 PRINT "MEAN = " MEAN "SIGMA =" SIGMA
1470 INPUT "DO YOU WANT A PRINTOUT (Y/N) (make sure printer is on)";PO$
1480 IF PO$="y" OR PO$="Y" THEN 1490 ELSE CLS:GOTO 1170
1490 LPRINT"RECORD","HEELSTRIKE","HSVSUM"
1500 LPRINT"------","----------","------"
1510 FOR Z=1 TO RN :LPRINT RN$(Z),HS%(Z),HSVSUM(Z):NEXT Z
1520 LPRINT:LPRINT "MEAN = " MEAN "  SIGMA =" SIGMA
```

```
1530 CLS:GOTO 1170:'MENU
1540 ' SCREEN PRINT OF EDITED DATA
1550 CLS
1560 PRINT TAB(2)"PT"TAB(9)"L"TAB(15)"R"TAB(20)"VSUM"
1570 FOR Q%=1 TO DATACNT(T%)
1580 IF (Q%/20-INT(Q%/20))<>0 THEN 1620
1590 LOCATE 23,30:PRINT"HIT ANY KEY TO CONTINUE"
1600 Z$=INKEY$:IF Z$="" THEN 1600
1610 CLS
1620 PRINT TAB(2)Q%TAB(8)DATAPTS(Q%,7)TAB(14)DATAPTS(Q%,8);:PRINT USING"###
#.##"; DATAPTS(Q%,1)+DATAPTS(Q%,2)+DATAPTS(Q%,3)
1630 NEXT Q%
1640 LOCATE 23,30 :PRINT "HIT ANY KEY TO CONTINUE OR (R) TO REPEAT"
1650 Z$=INKEY$:IF Z$="" THEN 1650
1660 IF Z$="R" OR Z$="r" THEN 1550: 'REPEAT
1670 CLS
1680 RETURN
```

```
10    '!!!!!!!!!!!!! START OF EDIT BLOCK 1 !!!!!!!!!!!!!!!!!!!!
20    OPEN "COM2:2400,S,7,1,RS,CS,DS32000,CD"AS#3
30  COLOR 0,7 : 'REVERSES SCREEN
40  CLS
50  DIM DATAPTS(384,8),C(8),STRIKE(5),HIT(4,10),VSUM(384),CPX(384),CPY(384)
,        VCPX(384),VCPY(384),RVCP(384),VA(384),TYPE$(6)
60  DIM TEXTR(200),EXTR(25,200),H(10),AVEXTR(50),SDVSUM(384),HS(5),FDVSUM(3
84)          ,ZC(5),DATACNT(10),DCOFF(9),VERT(6),CALIB(9),STDEV(50),RDES$(25)
,TEMP(200)
70  GOSUB 520         '   subroutine to load in CALIB(I) factors
80  ' NOW HAVE CALIBRATORS
90  INPUT "Insert desired data disk (unedited) in drive b.",C%
100  PRINT:PRINT "Available files:  "
110  PRINT:FILES "B:*.*"
120  PRINT:INPUT "Desired data file  ",E$
130  F$="B:"+E$                   'Ensure drive B.
140  IF T%=0 THEN GOSUB 680 :'Get traverse counters ONLY ONCE PER FILE
150  INPUT"SELECT TRAVERSE ",T%
160  PRINT "Data points taken for traverse";T%;"=";DATACNT(T%)
170  GOSUB 820      '  get one traverse worth of data.
180  ' commence edit
190  FOR H%=1 TO 3 : 'Loop through 3 channels. 7 & 8 EDITED IN BLOCK2
200    SUM=0
210    FOR X=0 TO 2
220      ENDER% = DATACNT(T%)
230      SUM=DATAPTS(ENDER%-X,H%)+SUM
240    NEXT X
250  ' The above x driven loop gets the first or last 3 pts of traverse for
 D.C.
260  '   offset purposes.
270    SUM=SUM/3         'Average D.C. offset for this channel
280    FOR Q%=1 TO DATACNT(T%)-1
290      DAT=DATAPTS(Q%,H%)      'GET DATAPTS
300      DAT=DAT-SUM    'CORRECT DC OFFSET
310      IF H% >3 THEN GOTO 340
320      DAT=DAT/CALIB(H%)
330      GOTO 430   'LOOP AGAIN
340      REM entrance point fo M.L. correction
350      IF H%<>4 THEN 370
360      IF DAT<0 THEN DAT=DAT/CALIB(6) ELSE DAT=DAT/CALIB(4)
370      REM entrance point for M.L. correction
380      IF H%<> 5 THEN 400
390      IF DAT<0 THEN DAT=DAT/CALIB(7) ELSE DAT=DAT/CALIB(5)
400      REM entrance for P.A.
410      IF H%<>6 THEN 430 : 'NO EDIT OF CHS 7,8
420      IF DAT<0 THEN DAT=DAT/CALIB(9) ELSE DAT=DAT/CALIB(8)
430      DATAPTS(Q%,H%)=DAT
440   NEXT Q%
450   PRINT "CHANNEL  ";H%;" EDITED"
460  NEXT H%         'Next channel.
470  REM        At this point one complete traverse has been edited.
480  PRINT:PRINT "FILE " E$   " TRAVERSE " T% "HAS NOW BEEN EDITED"
490  PRINT "PROCEEDING TO BLOCK 2"
500  GOTO 1040: 'BLOCK 2
510 ' BLOCK 1 SUBROUTINES FOLLOW
```

```
520 '****SUBROUTINE TO READ IN CALIB(I) FACTORS FROM CALROSS ********
530    PRINT "Please enter calibration disk into drive B."
540    INPUT "Strike [ENTER] when ready...",CRAP%
550    FILES "B:*.*"
560    INPUT "Desired file (without filename extension):  ",CAL$
570    CAL$="B:"+CAL$+" RCAL": 'ENSURES DRIVE B:
580    OPEN CAL$ FOR INPUT AS #1
590    CLS:PRINT "CALIBRATION FACTORS READ FROM FILE":PRINT
600    FOR I=1 TO 9
610     INPUT#1,CALIB(I)
620     PRINT CALIB(I)
630    NEXT I
640    INPUT "CONTINUE WITH EDIT (Y/N)";ANS$
650    IF ANS$="N" OR   ANS$="n" THEN END
660    CLOSE#1
670    RETURN
680 'SUBROUTINE  IS USED TO GET THE TRAVERSE COUNTERS.
690OPEN F$ AS #1 LEN=768
700     FIELD #1,192 AS A$,192 AS B$,192 AS C$,192 AS D$
710     REC=49  'Last record contains desired data.
720     GET #1,REC       'Read data.
730     FOR I=1 TO 10 : ' ONLY 6 COUNTS USED
740       K=I*2-1
750       DATACNT(I)=CVI(MID$(A$,K,2))
760NEXT I
770 'DATACNT NOW IS PRIMED WITH TRAVERSE COUNTERS.
780 'FILE IS LEFT OPEN, AS DATA WILL BE READ IN MAIN-LINE.
790RETURN'Back to mainline.
800   REMEND OF SUBROUTINE
810 '
820REM Subroutine for reading unedited data.
830 'FILE IS ALREADY OPEN.
840    FOR CH%=1 TO 3          'Loop through 3 channels.
850REC=(CH%+(T%-1)*8)
860GET #1,REC
870    PRINT "CHANNEL  "CH%;" READ.."
880FOR I=1 TO 96
890       K=I*2-1
900       DATAPTS(I,CH%)=CVI(MID$(A$,K,2))
910       DATAPTS(I+96,CH%)=CVI(MID$(B$,K,2))
920       DATAPTS(I+192,CH%)=CVI(MID$(C$,K,2))
930       DATAPTS(I+288,CH%)=CVI(MID$(D$,K,2))
940    NEXT I
950    FOR I=1 TO 384
960    IF DATAPTS(I,CH%) > 32767 THEN DATAPTS(I,CH%)=DATAPTS(I,CH%)-65536!
970    NEXT I
980    NEXT CH%
990    PRINT
1000    RETURN  'Back to mainline.
1010   REMEND OF SUBROUTINE
1020 '!!!!!!!!!!!! END OF EDIT BLOCK 1!!!!!!!!!!!!!!!!!!!!
1030 '
1040 '@@@@@@@@@@@@ START BLOCK 2 @@@@@@@@@@@@@@@@@@@
1050  VTHRESH=10: '10 LBS. MIN. BEFORE CPX,ETC. CALCULATED
1060  TBASE=92.5:'BASE WIDTH INCHES
```

```
1070  HEIGHT=45.6:'PERPENDICULAR HEIGHT AT APEX
1080  SWITCH=200: ' THIS IS ABOUT 1 VOLTS ON A/D
1090  H(1)=-.97226:H(2)=.4462:H(3)=-.25667:H(4)=.15497:H(5)=-.092024
          :H(6)=.051667:H(7)=-.0268:H(8)=.01347:H(9)=-8.889998E-03: 'FOR D
IG DIFF.
1100  PRINT:PRINT "A FEW MOMENTS PLEASE WHILE I DETERMINE CPX,CPY,ETC."
1110  ' ROUTINE TO FIND CPX,ETC. FOLLOWS
1120  FOR Z=DATACNT(T%) TO 384:VSUM(Z)=0:CPX(Z)=0:CPY(Z)=0:NEXT:
                'CLEAR ARRAY  FROM LAST USE
1130  FOR Q%=1 TO DATACNT(T%)
1140    IF DATAPTS(Q%,7)<SWITCH THEN DATAPTS(Q%,7)=0 ELSE
            DATAPTS(Q%,7)=1: 'PUTS 1'S OR 0'S IN SWITCH CHANNEL 7
1150    IF DATAPTS(Q%,8)<SWITCH THEN DATAPTS(Q%,8)=0 ELSE
            DATAPTS(Q%,8)=1: 'PUTS 1'S OR 0'S IN SWITCH CHANNEL 8
1160    VSUM(Q%)=DATAPTS(Q%,1)+DATAPTS(Q%,2)+DATAPTS(Q%,3)
1170    ' following is later used for h.s. ident routine using sdvsum
1180    IF (Q%<20) OR (DATACNT(T%)-Q%) <20 THEN GOTO 1240 : 'IGNORE ENDS
1190    FVSUM=DATAPTS(Q%+1,1)+DATAPTS(Q%+1,2)+DATAPTS(Q%+1,3): 'FUTURE SUM
1200    PVSUM=DATAPTS(Q%-1,1)+DATAPTS(Q%-1,2)+DATAPTS(Q%-1,3): 'PAST SUM
1210    FDVSUM(Q%)=FVSUM-VSUM(Q%): 'PRESENT FIRST DIFFERENCE
1220    PFDVSUM=VSUM(Q%)-PVSUM: 'PAST FIRST DIFFERENCE
1230    SDVSUM(Q%)=FDVSUM(Q%)-PFDVSUM :' SECOND DIFFERENCE
1240    IF VSUM(Q%)<VTHRESH THEN CPX(Q%)=0:CPY(Q%)=0:VCPX(Q%)=0:
            VCPY(Q%)=0:RVCP(Q%)=0:VA(Q%)=0:GOTO 1340
1250    'BOARD MUST HAVE MIN LOAD BEFORE FINDING CPX,ETC.
1260    'following finds CPX
1270    NUM1=(DATAPTS(Q%,2)+2*DATAPTS(Q%,3))*TBASE/2
1280    CPX(Q%)=NUM1/VSUM(Q%) :'inches -largest value 92.5
1290    'cpx=0 at NE corner,increases going west
1300    'following finds CPY
1310    NUM2=DATAPTS(Q%,2)*HEIGHT
1320    CPY(Q%)=NUM2/VSUM(Q%) : 'inches-largest value is 45.6
1330    'cpy=0 at N side,increases going south
1340   NEXT Q%
1350 ' ROUTINE TO FIND VCPX,VCPY,RVCP,VA USING F.I.R.DIFF.
1360 ' TAKEN FROM  ONE-DIM.DIG.SIGNAL PROC.CHI-TSONG CHENG PAGE 204
1370   FOR Z=DATACNT(T%)-20 TO 384:VCPX(Z)=0:VCPY(Z)=0:RVCP(Z)=0: NEXT Z
           : 'CLEARS ARRAY FROM LAST USE
1380   FOR Q%=10 TO DATACNT(T%)-10
1390     VCPX(Q%)=0:VCPY(Q%)=0:RVCP(Q%)=0:'CLEARS ARRAYS (STARTING VALUES
             FOR THIS LOOP)
1400     IF VSUM(Q%-9)<VTHRESH OR VSUM(Q%+9)<VTHRESH THEN 1490
1410     FOR N=-9 TO 9
1420      IF N=0 THEN GOTO 1460
1430      IF N<0 THEN CC=+H(-N) ELSE CC=-H(N):' SIGNS OPPOSITE OF IN BOOK
1440      VCPX(Q%)=VCPX(Q%)+CC*CPX(Q%+N)
1450      VCPY(Q%)=VCPY(Q%)+CC*CPY(Q%+N)
1460     NEXT N
1470     VA(Q%)=180/3.14159*ATN(VCPY(Q%)/VCPX(Q%)): ' IN DEGREES
1480     RVCP(Q%)=SQR(VCPX(Q%)^2+VCPY(Q%)^2)
1490   NEXT Q%
1500 GOTO 1680: 'SKIP HS IDENT  ALGORITHMS IN THIS CASE
1510 ' ROUTINE TO FIND HEEL STRIKES BASED ON RVCP
1520   J=15
1530   FOR I=20 TO DATACNT(T%)-10
```

```
1540    DELTARVCP=RVCP(I+1)-RVCP(I)
1550    IF J>=15 AND DELTARVCP>=.7 AND (RVCP(I+1)+RVCP(I+2)+RVCP(I+3)+RVCP(
I+4)+        RVCP(I+5))/5>=1 THEN PRINT "(RVCP) HEEL STRIKE AT:";I:J=0
1560    J=J+1
1570   NEXT I
1580   PRINT
1590 ' ROUTINE TO FIND HEEL STRIKES BASED ON VERTICAL FORCE DERIVATIVES
1600   K=1:J=20: 'MIN OFFSET BETWEEN HEEL STRIKES
1610   FOR Z%=25 TO DATACNT(T%)-25: 'SKIP FIRST AND LAST PORTIONS
1620    J=J+1
1630    FOR XX%=0 TO 3:SUM=SUM+RVCP(Z%+XX%):NEXT XX%
1640     AVRVCP=SUM/4:SUM=0:IF AVRVCP < 1.5 THEN GOTO 1660
1650     IF SDVSUM(Z%)>1.5 AND SDVSUM(Z%+1)>1.5 AND((SDVSUM(Z%)+SDVSUM(Z%+1
))/2)          >5 AND J>=20 THEN PRINT"(FORCE DER.) HEEL STRIKE  AT " Z% :HS
(K)=Z%:            K=K+1:J=0
1660  NEXT Z%
1670  INPUT"FINISHED READING  HEELSTRIKES?",GARBAGE$
1680  PRINT" PROCEEDING TO BLOCK 3"
1690 '
1700 '*************** BLOCK 3 ****************************************
1710 ' ROUTINE TO EXTRACT ONE COMPLETE RECORD FROM A THREE STEP
1720 '********* FRACTIONAL SPLIT DETERMINATION OF VERT FORCES*****
1730 ' TO BEST UNDERSTAND THE FOLLOWING, SEE LOG NOTES AUG 5 ENTRY
1740 INPUT"GIVE ME H.S AND TOE OFF TIMES FOR DUAL SUPPORT PHASE 1 ",HS1,TOF
1
1750 INPUT"GIVE ME H.S AND TOE OFF TIMES FOR DUAL SUPPORT PHASE 2 ",HS2,TOF
2
1760 ' PHASE 1
1770 HSCPX1=CPX(HS1):TOFCPX1=CPX(TOF1):'CPX @ HS, AND @ TOF
1780 TFVCPX1=ABS(CPX(HS1-2)-CPX(HS1-12))/10: 'VEL OF CPX FOR TRAILING FOOT
1790 LFVCPX1=ABS(CPX(TOF1+12)-CPX(TOF1+2))/10: 'VEL OF CPX FOR LEAD FOOT
1800 ' PHASE 2
1810 HSCPX2=CPX(HS2):TOFCPX2=CPX(TOF2):'CPX @ HS, AND @ TOF
1820 TFVCPX2=ABS(CPX(HS2-2)-CPX(HS2-12))/10: 'VEL OF CPX FOR TRAILING FOOT
1830 LFVCPX2=ABS(CPX(TOF2+12)-CPX(TOF2+2))/10: 'VEL OF CPX FOR LEAD FOOT
1840 'ABOVE ASSUMES  BOTH VELOCITIES AS CONSTANT THRU THE DUAL SUPPORT PHAS
ES
1850 'ROUTINE TO FIND LEADING EDGE OF LEAD FOOT
1860 TEXTR(0)=0: 'EXTRACTED RECORD NUMBERED FROM ZERO
1870 FOR I=1 TO (TOF1-HS1-1)
1880    T=HS1+I
1890    NUM=ABS(CPX(T)-HSCPX1)-ABS(TFVCPX1*I): 'THIS IS 'D ' ON MY FIGURE
1900    DEN=NUM+ABS(TOFCPX1-CPX(T))+ABS(LFVCPX1*I): 'D +E ON MY FIG
1910    RLLF=NUM/DEN : 'RELATIVE LOADING OF LEAD FOOT
1920    TEXTR(I)=RLLF*VSUM(T): 'ABSOLUTE LOADING OF THE LEAD FOOT
1930 NEXT I
1940 ' SINGLE SUPPORT PHASE
1950 FOR I=(TOF1-HS1) TO (HS2-HS1)
1960 T=HS1 +I
1970   TEXTR(I)=VSUM(T)
1980 NEXT I
1990 ' FINDS TRAILING EDGE from sum moments about lead ft. local cpx
2000 FOR I=(HS2-HS1+1) TO (TOF2-HS1-1): 'ALL BUT LAST POINT
2010    T=HS1+I: Z=I-HS2 : 'Z IS NO. SAMPLES INTO D.S.PHASE 2
2020 '  NUM=ABS(TOFCPX2-CPX(T))+ABS(LFVCPX2*Z): 'THIS IS 'E ' ON MY FIGURE
```

```
2030 '  DEN=NUM+ABS(CPX(T)-HSCPX2)-ABS(TFVCPX2*Z): 'D +E ON MY FIG
2040 '  FOLLOWING IS THE  PEG-LEG APPROACH- IGNORES LOCAL CPX PROGRESSION
2050    NUM=ABS(TOFCPX2-CPX(T)): 'PEG-LEG
2060    DEN=NUM+ABS(CPX(T)-HSCPX2): 'PEG-LEG
2070    RLTF=NUM/DEN : 'RELATIVE LOADING OF TRAILING FOOT
2080    TEXTR(I)=VSUM(T)*RLTF:'ABSOLUTE LOAD ON TRAILING FOOT (EXTRACTED)
2090 NEXT I
2100 TEXTR (TOF2-HS1)=0:'LAST POINT
2110 ' ONE COMPLETE EXTRACTED RECORD NOW EXISTS
2120 '************************************
2130 ' SUB TO PLOT DATA ON SCREEN
2140 SCREEN 2,0,0:CLS
2150 KEY OFF
2160 LOCATE 9,1:PRINT;"V":LOCATE 11,1:PRINT;"S":LOCATE 13,1:PRINT;"U":LOCAT
E 15,     1:PRINT;"M":LOCATE 2,1:PRINT;"250"
2170 LOCATE 25,35:PRINT;"TIME"
2180 LINE (30,0)-(30,180):LINE -(639,180)
2190 SCRVSCALE=180/250:SCRHSCALE=(639-50)/DATACNT(T%)
2200 X=50:Y=180
2210 FOR I=1 TO DATACNT(T%): 'PLOTS VSUM
2220   XNEW=X+SCRHSCALE
2230   YNEW=180-VSUM(I)*SCRVSCALE
2240   LINE (X,Y)-(XNEW,YNEW)
2250   X=XNEW
2260   Y=YNEW
2270 NEXT I
2280 X=50+HS1*SCRHSCALE:Y=180: 'PLOTS EXTRACTED RECORD
2290 FOR I=1 TO (TOF2-HS1)
2300   XNEW=X+SCRHSCALE
2310   YNEW=180-TEXTR(I)*SCRVSCALE
2320   LINE (X,Y)-(XNEW,YNEW)
2330   X=XNEW
2340   Y=YNEW
2350 NEXT I
2360 LOCATE 22,53:PRINT" Press any key to continue"
2370 CRAPO$=INKEY$:IF CRAPO$="" THEN 2370
2380 SCREEN 0,0,0:CLS
2390  CLS: PRINT " NOW ENTERING BLOCK 4"
2400  ' @@@@@@@@@@@@@@@@ BLOCK 4 @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
2410  'THIS SECTION  NORMALIZES, AND COMPARES UP TO 25 RECORDS OF THE SAME
         EXTRACTED RECORD (MAX OF 200 DATAPTS PER RECORD)
2420  'THE FIRST RECORD CHOSEN ESTABLISHES THE NUMBER OF DATA POINTS
         IN THE INTERVAL USED FOR NORMALIZING
2430  RN=RN+1 :'RECORD NO.
2440  PRINT " MAKE SURE PLOTTER IS TURNED ON"
2450  RDES$(RN)=E$+"  TRAV "+STR$(T%) : 'GIVES RECORD A NAME
2460  CP=TOF2-HS1
2470  PRINT:PRINT "THERE ARE  " CP+1 " DATAPOINTS FOR THIS RECORD (BOTH END
 PTS INCLUDED)"
2480  IF RN=1 THEN PRINT:PRINT"THIS FIRST RECORD DETERMINES THE TIME BASE T
O WHICH ALL FURTHER RECORDS WILL BE NORMALIZED":MCP=CP
2490  INC=CP/MCP :  'INCREMENTAL TIME (INC=1 FOR FIRST RECORD)
2500  'ROUTINE TO FIND VALS AT EACH OF MCP EQUAL INTERVALS
2510  EXTR(RN,0)=TEXTR(0):EXTR(RN,MCP)=TEXTR(CP):'FIRST AND LAST PTS.
2520  FOR  I=1 TO MCP-1
```

```
2530    ET=I*INC: 'RATIONAL NO.
2540   'ROUTINE TO INTERPOLATE TO FIND EXTR(I) AT ET
2550    CRZ=TEXTR(INT(ET)+1)-TEXTR(INT(ET)):' MAY BE +OR-
2560    CRN=ABS(ET-INT(ET)) : '=0 FOR RN=1
2570    DELT=CRZ*CRN/1 : '=0 FOR RN=1
2580    EXTR(RN,I)=TEXTR(INT(ET))+DELT
2590  NEXT I
2600  PRINT:PRINT "FILE: "E$"  TRAVERSE:"T% " HAS NOW BEEN NORMALIZED": PRI
NT
2610  PRINT" SELECT  (1)GET NEW FILE   (2)GET NEW TRAVERSE    (3)PLOT PRESE
NT                        (4)LIST FILES     (5)FIND AVERAGE RECORD (6)SCR.D
MP"
2620   DS$=INKEY$:IF DS$="" THEN 2620
2630   IF DS$="1" THEN FLAG=0:CLOSE#1:CLS:T%=0:GOTO 90:' NEW FILE
2640   IF DS$="2" THEN CLS:FLAG=0:GOTO 150 :'NEW TRAVERSE
2650   IF DS$="3" THEN 2950 ELSE IF DS$="4" THEN 3550
2660   IF DS$="5" THEN 2680 ELSE IF DS$="6" THEN 3630
2670   GOTO 2620 : 'TRAP
2680   ' ROUTINE TO FIND AVERAGE (MASTER) RECORD
2690   CLS:LOCATE 12,28:PRINT"FINDING  AVERAGE (MASTER) RECORD"
2700   FOR Q%=0 TO MCP
2710     SEXTR=0
2720     FOR RECORD =1 TO RN
2730       SEXTR = SEXTR + EXTR(RECORD,Q%)
2740     NEXT RECORD
2750     AVEXTR(Q%)=SEXTR/RN
2760    NEXT Q%
2770  ' AVEXTR(Q%) IS THE AVERAGED RECORD
2780   CLS:PRINT "MASTER FILE NOW CREATED"
2790   PRINT:PRINT"FINDING VARIANCES, STANDARD DEVIATIONS  "
2800 ' ROUTINE TO FIND VARIANCES, STANDARD DEVIATIONS
2810   FOR Q%=0 TO MCP
2820     DELSQ=0:SUM=0
2830     FOR RECORD=1 TO RN
2840       DELSQ=(EXTR(RECORD,Q%)-AVEXTR(Q%))^2
2850       SUM=SUM+DELSQ
2860     NEXT RECORD
2870     VAR=SUM/RN
2880     STDEV(Q%)=SQR(VAR)
2890   NEXT Q%
2900   PRINT:PRINT"SELECT    (1)PLOT MASTER       (2)GO TO MAIN MENU
                                   (3)DUMP TO FILE      (4)SCREEN PRINT DATA
                                        (5)HARD COPY"
2910   DS$=INKEY$:IF DS$="" THEN 2910
2920   IF DS$="1" THEN MASFLAG=1 :GOTO 2950 ELSE IF DS$="2" THEN 2610
2930   IF DS$="3" THEN GOTO 3840 ELSE IF DS$="4" THEN 3960
2940   IF DS$="5" THEN GOTO 4060 ELSE 2910
2950 '***********PLOTING ROUTINE HERE************************
2960 'THIS PART OF THE ROUTINE SETS UP THE COORDINATE SYSTEM, DRAWS THE AXE
S
2970 'AND LABELS THE DIAGRAM.
2980 CLS
2990 LOCATE 12,28:PRINT"PLOTTER PREPARED ?"
3000 CRAP$=INKEY$:IF CRAP$="" THEN 3000
3010 LOCATE 12,28:PRINT"***** PLOTTING *****"
```

```
3020 LOCATE 13,28:IF MASFLAG=0 THEN PRINT"      RECORD "RDES$(RN)
        ELSE PRINT"      MASTER"
3030 IF LABELSPLOTTED = 1 THEN GOTO 3120 'THIS ASSURES LABELS ONLY DONE ONC
E
3040 PRINT #3,"IN;DT";+CHR$(1);";DI1,0;SP1;"
3050 PRINT #3,"PU 10300 1000;PD 1000 1000;PD 1000 7650;" 'DRAW AXES
3060 PRINT #3,"PU 500 7450;LB";STR$(250)+CHR$(1);";"
3070 PRINT #3,"PU 0 4500;LB VERTICAL";+CHR$(1);";PU 0 4300;LB FORCE";+CHR$(
1);";"
3080 PRINT #3,"PU 500 1000;LB";STR$(0)+CHR$(1);";"
3090 PRINT #3,"PU 5000 500;LB TIME";+CHR$(1);";"
3100 LABELSPLOTTED = 1
3110 '
3120 'AT THIS POINT ALL THE LABELS AND AXES HAVE BEEN PLOTTED ONCE ALREADY
3130 '
3140 UPSCALE = (7650 - 1000)/250
3150 SIDESCALE = (10000 - 1500)/MCP
3160 '
3170 IF MASFLAG=1 THEN GOTO 3290: 'IS THIS PLOT OF MASTER OR VSUM?
3180 'FOLLOWING FOR EXTR PLOT (NOT AVEXTR)
3190   X=1500:Y=EXTR(RN,0)*UPSCALE + 1000
3200 PRINT #3,"PU ";X;" ";Y;";PD;"
3210 FOR I=0 TO MCP
3220   Y=EXTR(RN,I)*UPSCALE +1000
3230   X=X+SIDESCALE
3240   PRINT #3,"PD ";X;" ";Y;";"
3250 NEXT I
3260 PRINT #3,"PU 0 0;"
3270 CLS:GOTO 2610: 'MAIN MENU
3280 '
3290 'ENTRY PT FOR MASTER PLOT
3300   X=1500:Y=AVEXTR(0)*UPSCALE + 1000
3310 PRINT #3,"PU ";X;" ";Y;";PD;"
3320 FOR I=0 TO MCP
3330   Y=AVEXTR(I)*UPSCALE +1000
3340   X=X+SIDESCALE
3350   PRINT #3,"PD ";X;" ";Y;";"
3360 NEXT I
3370 PRINT #3,"PU 0 0;"
3380 INPUT " PLOT STD. DEV ENVELOPE (Y/N) ";ENV$
3390 IF ENV$="N" OR ENV$="n" THEN 3520
3400 INPUT" NUMBER OF STD. DEVIATIONS DESIRED ";NSD
3410 FOR PASS=1 TO 2
3420   X=1500:Y=AVEXTR(0)*UPSCALE + 1000
3430   PRINT #3,"PU ";X;" ";Y;";PD;": 'PLOTS ENVELOPE
3440   FOR I=0 TO MCP: 'PLOTS + FIRST PASS, - ON SECOND
3450     Y=(AVEXTR(I)+NSD*STDEV(I))*UPSCALE +1000
3460     X=X+SIDESCALE
3470     PRINT #3,"PD ";X;" ";Y;";"
3480   NEXT I
3490   PRINT #3,"PU 0 0;"
3500   NSD = -NSD
3510 NEXT PASS
3520 MASFLAG=0
3530 CLS:GOTO 2610 :' MAIN NENU
```

```
3540 '*********** END PLOT ROUTINE**********************
3550 '******** ROUTINE TO LIST FILES SO FAR NORMALIZED **
3560 CLS:PRINT:PRINT"                FILES NOW NORMALIZED "
3570 PRINT"           -------------------- "
3580 FOR X=1 TO RN: PRINT TAB(8)X  SPC(2)RDES$(X):NEXT X
3590 LOCATE 20,8:PRINT "STRIKE ANY KEY TO CONTINUE  "
3600 POOP$=INKEY$:IF POOP$="" THEN 3600
3610 CLS
3620 GOTO 2610: 'BACK TO MAIN MENU
3630 '*************** SCREEN PLOT SUBROUTINE ************************
3640 SCREEN 2,0,0:CLS
3650 KEY OFF
3660 LOCATE 9,1:PRINT;"V":LOCATE 11,1:PRINT;"S":LOCATE 13,1:PRINT;"U":LOCAT
E 15,     1:PRINT;"M":LOCATE 2,1:PRINT;"250"
3670 LOCATE 25,35:PRINT;"TIME"
3680 LINE (30,0)-(30,180):LINE -(639,180)
3690 SCRVSCALE=180/250:SCRHSCALE=(639-50)/MCP
3700 FOR J=1 TO RN
3710   X=50:Y=180-EXTR(J,MCP)*SCRVSCALE
3720 FOR I=0 TO MCP
3730   XNEW=X+SCRHSCALE
3740   YNEW=180-EXTR(J,I)*SCRVSCALE
3750   LINE (X,Y)-(XNEW,YNEW)
3760   X=XNEW
3770   Y=YNEW
3780 NEXT I
3790 NEXT J
3800 LOCATE 22,53:PRINT" Press any key to continue"
3810 CRAPO$=INKEY$:IF CRAPO$="" THEN 3810
3820 SCREEN 0,0,0:CLS:GOTO 2610 'MAIN MENU
3830 '
3840 '**** ROUTINE TO DUMP MASTER INFO TO FILE *****
3850 'THE NUMBER OF POINTS(MCP), THE MASTER RECORD, AND STD. DEV. AT EACH D
ATA         POINT IS STORED
3860 INPUT" FILE NAME ";FN$
3870 FN$="B:"+FN$
3880 INPUT" PLEASE PLACE DISK INTO DRIVE B:  AND STRIKE WHEN ENTER ",POOP$
3890 OPEN FN$ FOR OUTPUT AS #2
3900 WRITE#2,MCP
3910 FOR Q%=0 TO MCP: WRITE#2,AVEXTR(Q%),STDEV(Q%):NEXT Q%
3920 CLOSE#2
3930 GOTO 2610: 'MAIN MENU
3940 '
3950 '
3960 'SCREEN PRINTOUT OF MASTER FILE DATA
3970 ' CLS:PRINT TAB(5)"PT"TAB(12)"AVEXTR(Q%)"TAB(25)"STDEV(Q%)"
3980 PRINT TAB(5)"--"TAB(12)"----------"TAB(25)"---------"
3990 FOR Q%=0 TO MCP
4000   PRINT TAB(5)Q% TAB(12)AVEXTR(Q%)TAB(25)STDEV(Q%)
4010 NEXT Q%
4020 PRINT" (R) REPEAT      (C) TO RETURN TO LAST MENU    "
4030 DIR$=INKEY$:IF DIR$="" THEN 4030
4040 IF DIR$="R" THEN 3960 ELSE IF DIR$="C" THEN 2900 ELSE 4030
4050 '
4060 ' PRINTOUT OF MASTER FILE DATA
```

```
4070 INPUT" PRINTER ON ";POOP$
4080 INPUT"FILE NAME ";FN$
4090 LPRINT FN$:LPRINT
4100   LPRINT TAB(5)"PT"TAB(12)"AVEXTR(Q%)"TAB(25)"STDEV(Q%)"
4110 LPRINT TAB(5)"--"TAB(12)"----------"TAB(25)"---------"
4120 FOR Q%=0 TO MCP
4130   LPRINT TAB(5)Q% TAB(12)AVEXTR(Q%)TAB(25)STDEV(Q%)
4140 NEXT Q%
4150 CLS:GOTO 2900: 'LAST MENU
```

```
10  'THIS PROGRAM  NORMALIZES, AND COMPARES UP TO 25 RECORDS OF THE SAME
        ONE STEP TRAVERSE (MAX OF 350 DATAPTS PER RECORD)
20  'THE FIRST RECORD CHOSEN ESTABLISHES THE NUMBER OF DATA POINTS
        IN THE INTERVAL USED FOR NORMALIZING
30  'IT ASSUMES THAT THE APPROACH PLATE HEEL STRIKE (FHS) HAS BEEN READ BY
        THE DATA ACQUISITION PROGRAM
40   RN=0:'RECORD NO.
190  CLS: PRINT " MAKE SURE PLOTTER IS TURNED ON"
200  OPEN "COM2:2400,S,7,1,RS,CS,DS32000,CD"AS#3
210  DIM DATAPTS(384,8),TYPE$(6)
220  DIM DATACNT(10),DCOFF(9),VERT(6),CALIB(9)
230  DIM VSUM(25,200),TEMP(384),SVSUM(384),AVSUM(384),RDES$(25),STDEV(384)
240  GOSUB 790: '  subroutine to load in CALIB(I) factors
250  ' NOW HAVE CALIBRATORS
260  PRINT:INPUT "Insert desired data disk (unedited) in drive b.",C%
270  PRINT:PRINT "Available files:  "
280  PRINT:FILES "B:*.*"
290  PRINT:INPUT "Select desired data file  ",E$
300  F$="B:"+E$                  'Ensure drive B.
320  IF T%=0 THEN GOSUB 970 :'Get traverse counters ONLY ONCE PER FILE
330  PRINT:INPUT"SELECT TRAVERSE ",T%
340  RN=RN+1
350  PRINT "Data points taken for traverse";T%;"=";DATACNT(T%)
360  GOSUB 1110: '  get one traverse worth of data.
370  ' commence edit
380  FOR H%=1 TO 3              'Loop through 3 channels only
390    SUM=0
400    IF DATACNT(T%)>381 THEN BEEP:PRINT "** WARNING - HIGH DATA COUNT FOR
 THIS TRAVERSE- NO D.C. OFFSET CORRECTION APPLIED ":GOTO 480
410    FOR X=0 TO 2:SUM=DATAPTS(X,H%)+SUM:NEXT X
450  ' The above x driven loop gets first 3 pts of traverse for D.C.
460  ' offset purposes.
470    SUM=SUM/3        'Average D.C. offset.
480    FOR Q%=1 TO DATACNT(T%)-1
490      DAT=DATAPTS(Q%,H%)       'GET DATAPTS
500      DAT=DAT-SUM     'CORRECT DC OFFSET
510      DAT=DAT/CALIB(H%): 'APPLY CALIBRATOR
520      DATAPTS(Q%,H%)=DAT
530    NEXT Q%
540    PRINT "CHANNEL ";H%;" EDITED"
550  NEXT H%          'Next channel.
560  REM        At this point one complete traverse has been edited.
570  PRINT:PRINT "FILE " E$   " TRAVERSE " T% "HAS NOW BEEN EDITED"
580 PRINT:PRINT"DO YOU WANT A SCREEN PRINTOUT OF DATA (Y/N) ?"
590 AA$=INKEY$:IF AA$="" THEN 590
600 IF AA$="N" OR AA$="n" THEN 760 ELSE IF AA$<>"Y" AND AA$<>"y" THEN 590
610 '***********SCREEN PRINT ROUTINE*******
620 CLS
630 PRINT TAB(2)"PT"TAB(9)"L"TAB(15)"R"TAB(20)"VSUM"
640 FOR Q%=1 TO DATACNT(T%)
650  IF Q%=1 OR ((Q%-1)/20 -INT((Q%-1)/20))<>0 THEN 690
       :'CHECKS FOR MULTIPLES OF 20
660  LOCATE 22,30:PRINT"HIT ANY KEY TO CONTINUE"
670  Z$=INKEY$:IF Z$="" THEN 670
680  CLS
```

```
690  PRINT TAB(2)Q%TAB(8)DATAPTS(Q%,7)TAB(14)DATAPTS(Q%,8);:
          PRINT USING"####.##  ";(DATAPTS(Q%,1)+DATAPTS(Q%,2)+DATAPTS(Q%,3)
)
700 NEXT Q%
710 LOCATE 22,30:PRINT"  HIT (R) TO REPEAT SCREEN DISPLAY "
720 LOCATE 23,30:PRINT"  HIT ANY OTHER KEY TO CONTINUE"
730 Z$=INKEY$:IF Z$="" THEN 730
740 IF Z$="R" OR Z$="r" THEN 610
760  CLS:PRINT "PROCEEDING TO BLOCK 2"
770  GOTO 1350: 'BLOCK 2
780 '
790 '****SUBROUTINE TO READ IN CALIB(I) FACTORS FROM CALROSS ********
800    REM
810    PRINT "Please enter calibration disk into drive B."
820    PRINT:INPUT "Strike [ENTER] when ready...",CRAP%
830    PRINT:FILES "B:*.*"
840    PRINT:INPUT "Desired file (without filename extension) OR 'R '  ",CA
L$
850    CAL$="B:"+CAL$+" RCAL"
860    OPEN CAL$ FOR INPUT AS #1
870    CLS:PRINT "CALIBRATION FACTORS READ FROM FILE":PRINT
880    FOR I=1 TO 9
890    INPUT#1,CALIB(I)
900    PRINT CALIB(I)
910    NEXT I
920    INPUT "CONTINUE WITH EDIT (Y/N)";ANS$
930    IF ANS$="N" OR  ANS$="n" THEN END
940    CLOSE#1
950    RETURN
960 '
970 'SUBROUTINE  IS USED TO GET THE TRAVERSE COUNTERS.
980OPEN F$ AS #1 LEN=768
990     FIELD #1,192 AS A$,192 AS B$,192 AS C$,192 AS D$
1000     REC=49  'Last record contains desired data.
1010     GET #1,REC       'Read data.
1020     FOR I=1 TO 10
1030K=I*2-1
1040DATACNT(I)=CVI(MID$(A$,K,2))
1050NEXT I
1060 'DATACNT NOW IS PRIMED WITH TRAVERSE COUNTERS.
1070 'FILE IS LEFT OPEN, AS DATA WILL BE READ IN MAIN-LINE.
1080RETURN'Back to mainline.
1090   REMEND OF SUBROUTINE
1100 '
1110REM Subroutine for reading unedited data.
1120 'FILE IS ALREADY OPEN.
1130    FOR CH%=1 TO 8          'Loop through 8 channels.
1140     IF CH%>3 AND CH%<7 THEN 1280 : '4,5,6 OF NO INTEREST
1150     REC=(CH%+(T%-1)*8)
1160     GET #1,REC
1170     PRINT "CHANNEL "CH%;" READ.."
1180     FOR I=1 TO 96
1190       K=I*2-1
1200       DATAPTS(I,CH%)=CVI(MID$(A$,K,2))
1210       DATAPTS(I+96,CH%)=CVI(MID$(B$,K,2))
```

```
1220        DATAPTS(I+192,CH%)=CVI(MID$(C$,K,2))
1230        DATAPTS(I+288,CH%)=CVI(MID$(D$,K,2))
1240      NEXT I
1250      FOR I=1 TO 384
1260       IF DATAPTS(I,CH%) > 32767 THEN DATAPTS(I,CH%)=DATAPTS(I,CH%)-6553
6!
1270      NEXT I
1280     NEXT CH%
1290     PRINT "TRAVERSE ";T%;" READ"
1300     RETURN  'Back to mainline.
1310     REM      END OF SUBROUTINE
1320 '!!!!!!!!!!!! END OF EDIT BLOCK 1!!!!!!!!!!!!!!!!!!!!
1340 '
1350 '@@@@@@@@@@@@START BLOCK 2@@@@@@@@@@@@@@@@@@@@
1360  RDES$(RN)=E$+"  TRAV "+STR$(T%) : 'GIVES RECORD A NAME
1370 'FOLLOWING FILLS TEMP ARRAY
1380  FOR Z=DATACNT(T%) TO 384:TEMP(Z)=0:NEXT Z: 'CLEARS DATA FROM LAST USE
1390  FOR Q%=1 TO DATACNT(T%)
1400    TEMP(Q%)=DATAPTS(Q%,1)+DATAPTS(Q%,2)+DATAPTS(Q%,3)
1410  NEXT Q%
1420  CLS:PRINT:PRINT"GIVE ME THE HEEL STRIKE AND TOE-OFF TIMES "
1430  INPUT HST,TOT
1440  CP=TOT-HST
1450  PRINT:PRINT "CYCLE PERIOD = " CP " DATAPOINTS FOR THIS RECORD"
1460  IF RN=1 THEN PRINT:PRINT"THIS FIRST RECORD DETERMINES THE TIME BASE T
O WHICH ALL FURTHER RECORDS WILL BE NORMALIZED":MCP=CP
1470  INC=CP/MCP :  'INCREMENTAL TIME (INC=1 FOR FIRST RECORD)
1480  'ROUTINE TO FIND VALS AT EACH OF MCP EQUAL INTERVALS
1490  VSUM(RN,0)=TEMP(HST):VSUM(RN,MCP)=TEMP(TOT):'FIRST AND LAST PTS.
1500  FOR  I=1 TO MCP-1
1510    ET=HST+I*INC: 'RATIONAL NO.
1520   'ROUTINE TO INTERPOLATE TO FIND VSUM AT ET
1530    CRZ=TEMP(INT(ET)+1)-TEMP(INT(ET)):' MAY BE +OR-
1540    CRN=ABS(ET-INT(ET)) : '=0 FOR RN=1
1550    DELT=CRZ*CRN/1 : '=0 FOR RN=1
1560    VSUM(RN,I)=TEMP(INT(ET))+DELT
1570  NEXT I
1580  PRINT:PRINT "FILE: "E$"  TRAVERSE:"T% " HAS NOW BEEN NORMALIZED": PRI
NT
1590  PRINT" SELECT  (1)GET NEW FILE   (2)GET NEW TRAVERSE    (3)PLOT PRESE
NT                          (4)LIST FILES     (5)FIND AVERAGE RECORD (6)QUIT
PROGRAM                         (7)SCR.DMP"
1600   DS$=INKEY$:IF DS$="" THEN 1600
1610   IF DS$="1" THEN CLOSE#1:CLS:T%=0:GOTO 260:' NEW FILE
1620   IF DS$="2" THEN CLS:GOTO 330 :'NEW TRAVERSE
1630   IF DS$="3" THEN 1920 ELSE IF DS$="4" THEN 2380
1640   IF DS$="5" THEN 1660 ELSE IF DS$="6" THEN STOP
1650   IF DS$="7" THEN GOTO 2460 ELSE GOTO 1600 : 'SCREEN DUMP
1660   ' ROUTINE TO FIND AVERAGE (MASTER) RECORD
1670   CLS:LOCATE 12,28:PRINT"FINDING  AVERAGE (MASTER) RECORD"
1680   FOR Q%=0 TO MCP
1690     SVSUM=0
1700     FOR RECORD =1 TO RN
1710       SVSUM = SVSUM + VSUM(RECORD,Q%)
1720     NEXT RECORD
```

```
1730     AVSUM(Q%)=SVSUM/RN
1740    NEXT Q%
1750  ' AVSUM(Q%) IS THE AVERAGED RECORD
1760   CLS:PRINT "MASTER FILE NOW CREATED"
1770   PRINT:PRINT"FINDING VARIANCES, STANDARD DEVIATIONS  "
1780 ' ROUTINE TO FIND VARIANCES, STANDARD DEVIATIONS
1790   FOR Q%=0 TO MCP
1800     DELSQ=0:SUM=0
1810     FOR RECORD=1 TO RN
1820       DELSQ=(VSUM(RECORD,Q%)-AVSUM(Q%))^2
1830       SUM=SUM+DELSQ
1840     NEXT RECORD
1850     VAR=SUM/RN
1860     STDEV(Q%)=SQR(VAR)
1870   NEXT Q%
1880   PRINT:PRINT"SELECT    (1)PLOT MASTER       (2)GO TO MAIN MENU
                                 (3)DUMP TO FILE       (4)SCREEN PRINT DATA
                                      (5)HARD COPY"
1890   DS$=INKEY$:IF DS$="" THEN 1890
1900   IF DS$="1" THEN MASFLAG=1 :GOTO 1920 ELSE IF DS$="2" THEN 1590
1910   IF DS$="3" THEN GOTO 3000 ELSE IF DS$="4" THEN 4000
1915   IF DS$="5" THEN GOTO 5000 ELSE 1890
1920 '***********PLOTING ROUTINE HERE************************
1930 'THIS PART OF THE ROUTINE SETS UP THE COORDINATE SYSTEM, DRAWS THE AXE
S
1940 'AND LABELS THE DIAGRAM.
1950 CLS
1960 LOCATE 12,28:PRINT"PLOTTER PREPARED ?"
1970 CRAP$=INKEY$:IF CRAP$="" THEN 1970
1980 LOCATE 12,28:PRINT"***** PLOTTING *****"
1990 LOCATE 13,28:IF MASFLAG=0 THEN PRINT"     RECORD "RDES$(RN)
        ELSE PRINT"     MASTER"
2000 IF LABELSPLOTTED = 1 THEN GOTO 2090 'THIS ASSURES LABELS ONLY DONE ONC
E
2010 PRINT #3,"IN;DT";+CHR$(1);";DI1,0;SP1;"
2020 PRINT #3,"PU 10300 1000;PD 1000 1000;PD 1000 7650;" 'DRAW AXES
2030 PRINT #3,"PU 500 7450;LB";STR$(250)+CHR$(1);";"
2040 PRINT #3,"PU 0 4500;LB VERTICAL";+CHR$(1);";PU 0 4300;LB FORCE";+CHR$(
1);";"
2050 PRINT #3,"PU 500 1000;LB";STR$(0)+CHR$(1);";"
2060 PRINT #3,"PU 5000 500;LB TIME";+CHR$(1);";"
2070 LABELSPLOTTED = 1
2080 '
2090 'AT THIS POINT ALL THE LABELS AND AXES HAVE BEEN PLOTTED ONCE ALREADY
2100 '
2110 UPSCALE = (7650 - 1000)/250
2120 SIDESCALE = (10000 - 1500)/MCP
2130 '
2140 IF MASFLAG=1 THEN GOTO 2260: 'IS THIS PLOT OF MASTER OR VSUM?
2150 'FOLLOWING FOR VSUM PLOT (NOT AVSUM)
2160 X=1500:Y=VSUM(RN,0)*UPSCALE + 1000
2170 PRINT #3,"PU ";X;" ";Y;";PD;"
2180 FOR I=0 TO MCP
2190   Y=VSUM(RN,I)*UPSCALE +1000
2200   X=X+SIDESCALE
```

```
2210   PRINT #3,"PD ";X;" ";Y;";"
2220 NEXT I
2230 PRINT #3,"PU 0 0;"
2240 CLS:GOTO 1590: 'MAIN MENU
2250 '
2260 'ENTRY PT FOR MASTER PLOT
2270 X=1500:Y=AVSUM(0)*UPSCALE + 1000
2280 PRINT #3,"PU ";X;" ";Y;";PD;"
2290 FOR I=0 TO MCP
2300   Y=AVSUM(I)*UPSCALE +1000
2310   X=X+SIDESCALE
2320   PRINT #3,"PD ";X;" ";Y;";"
2330 NEXT I
2340 PRINT #3,"PU 0 0;"
2350 MASFLAG=0
2360 CLS:GOTO 1590 :' MAIN NENU
2370 '*********** END PLOT ROUTINE**********************
2380 '******** ROUTINE TO LIST FILES SO FAR NORMALIZED **
2390 CLS:PRINT:PRINT"           FILES NOW NORMALIZED "
2400 PRINT"          -------------------- "
2410 FOR X=1 TO RN: PRINT TAB(8)X  SPC(2)RDES$(X):NEXT X
2420 LOCATE 20,8:PRINT "STRIKE ANY KEY TO CONTINUE  "
2430 POOP$=INKEY$:IF POOP$="" THEN 2430
2440 CLS
2450 GOTO 1590: 'BACK TO MAIN MENU
2460 '*************** SCREEN DMP SUBROUTINE ****************************
2470 SCREEN 2,0,0:CLS
2480 KEY OFF
2490 LOCATE 9,1:PRINT;"V":LOCATE 11,1:PRINT;"S":LOCATE 13,1:PRINT;"U":LOCAT
E 15,     1:PRINT;"M":LOCATE 2,1:PRINT;"250"
2500 LOCATE 25,35:PRINT;"TIME"
2510 LINE (30,0)-(30,180):LINE -(639,180)
2520 SCRVSCALE=180/250:SCRHSCALE=(639-50)/MCP
2530 FOR J=1 TO RN
2540 X=50:Y=180-VSUM(J,MCP)*SCRVSCALE
2550 FOR I=0 TO MCP
2560   XNEW=X+SCRHSCALE
2570   YNEW=180-VSUM(J,I)*SCRVSCALE
2580   LINE (X,Y)-(XNEW,YNEW)
2590   X=XNEW
2600   Y=YNEW
2610 NEXT I
2620 NEXT J
2630 LOCATE 22,53:PRINT" Press any key to continue"
2640 CRAPO$=INKEY$:IF CRAPO$="" THEN 2640
2650 SCREEN 0,0,0:CLS:GOTO 1590 'MAIN MENU
3000 '**** ROUTINE TO DUMP MASTER INFO TO FILE *****
3010 'THE NUMBER OF POINTS(MCP), THE MASTER RECORD, AND STD. DEV. AT EACH D
ATA        POINT IS STORED
3020 INPUT" FILE NAME ";FN$
3025 FN$="B:"+FN$
3027 INPUT" PLEASE PLACE DISK INTO DRIVE B:  AND STRIKE WHEN ENTER ",POOP$
3030 OPEN FN$ FOR OUTPUT AS #2
3035 WRITE#2,MCP
3040 FOR Q%=0 TO MCP: WRITE#2,AVSUM(Q%),STDEV(Q%):NEXT Q%
```

```
3060 CLOSE#2
3070 GOTO 1590: 'MAIN MENU
4000 'SCREEN PRINTOUT OF MASTER FILE DATA
4010 CLS:PRINT TAB(5)"PT"TAB(12)"AVSUM(Q%)"TAB(25)"STDEV(Q%)"
4020 PRINT TAB(5)"--"TAB(12)"---------"TAB(25)"---------"
4030 FOR Q%=0 TO MCP
4040 PRINT TAB(5)Q% TAB(12)AVSUM(Q%)TAB(25)STDEV(Q%)
4050 NEXT Q%
4060 PRINT" (R) REPEAT      (C) TO RETURN TO LAST MENU    "
4065 DIR$=INKEY$:IF DIR$="" THEN 4065
4070 IF DIR$="R" THEN 4000 ELSE IF DIR$="C" THEN 1880 ELSE 4065
4090 '
5000 ' PRINTOUT OF MASTER FILE DATA
5005 INPUT" PRINTER ON ";POOP$
5007 INPUT"FILE NAME ";FN$
5009 LPRINT FN$:LPRINT
5010 LPRINT TAB(5)"PT"TAB(12)"AVSUM(Q%)"TAB(25)"STDEV(Q%)"
5020 LPRINT TAB(5)"--"TAB(12)"---------"TAB(25)"---------"
5030 FOR Q%=0 TO MCP
5040 LPRINT TAB(5)Q% TAB(12)AVSUM(Q%)TAB(25)STDEV(Q%)
5050 NEXT Q%
5060 CLS:GOTO 1880: 'LAST MENU
```

```
99CLEAR'Program reentry point.
DIM DATACNT(384),DATAPTS(384,8),DCOFF(9),CALDATA(384,6)
DIM VERT(6),CALIB(9)
CLS
GOSUB 1'Subroutine to calculate scale factors.
INPUT "Insert desired data disk (unedited) in drive b.",C%
PRINT
PRINT
PRINT "Available files:  "
PRINT
FILES "B:*.*"
PRINT
INPUT "Desired data file  ",F$
F$="B:"+F$'Ensure drive B.
'
'
GOSUB 2'Get traverse counters.
FOR T% = 1 TO 6
PRINT "Data points taken for traverse";T%;"=";DATACNT(T%)
GOSUB 3'Get one traverse worth of data.
IF T%=6 THEN CLOSE #1'Close unedited file.
FOR H%=1 TO 8'Loop through channels.
SUM=0
IF DATACNT(T%)>381 GOTO 339
FOR X=0 TO 2
ENDER% = DATACNT(T%)
SUM=DATAPTS(ENDER%-X,H%)+SUM
NEXT X
'The above x driven loop gets last 3 pts of traverse for D.C.
'offset purposes.
SUM=SUM/3'Average D.C. offset.
339FOR Q%=1 TO DATACNT(T%)-1
DAT=DATAPTS(Q%,H%)'Get data point.
DAT=DAT-SUM'Correct D.C. Offset.
IF H% > 6 THEN GOTO 140 'DO NOT EDIT TRIGGER.
IF H% > 3 THEN GOTO 100
DAT=DAT/CALIB(H%)
DAT=DAT/200*30000
GOTO 140'Loop again.
'The above two lines of code convert the A-D number into
'a linear integer between 0 and 30000.   The corresponding
'scale in force is:0=0 lbs.
'30000=200 lbs.
'
100REM entrance point for M.L. correction.
IF H%<> 4 THEN 110
IF DAT<0 THEN DAT=DAT/CALIB(6) ELSE DAT=DAT/CALIB(4)
REM
110REM entrance point for M.L. correction.
IF H%<> 5 THEN 120
IF DAT<0 THEN DAT=DAT/CALIB(7) ELSE DAT=DAT/CALIB(5)
120 REM entrace for P.A.
IF H%<> 6 THEN 130
IF DAT<0 THEN DAT=DAT/CALIB(9) ELSE DAT=DAT/CALIB(8)
130REM
```

```
REMThe following line deconverts a force (M.L. or P.A.)
REMinto an integer between 0-30000.  This saves disk
REM space on a data write, as only 2 bytes are needed
REM for an integer.  Increasing the range of the integers
REM  to 30000 tends to improve the resolution of the system.
REM
DAT=DAT/70*30000
REM 0 LBS = 0      70 LBS=30000
IF ABS(DAT > 30000) THEN PRINT DAT;"**OVERFLOW IMPENDING**"
140REMDestination if channel is a vertical channel.
DATAPTS(Q%,H%)=DAT
NEXT Q%
PRINT "CHANNEL ";H%;" EDITED"
NEXT H%'Next channel.
REMAt this point one complete traverse has been edited.
GOSUB 4'Write edited data to disk.
NEXT T%'Increment to next traverse.
GOSUB 5'Add final record to edited file.
CLOSE #2'Close edited data file.
INPUT "Do you wish to edit another? [Y/N]",W$
IF W$="Y" OR W$="y" THEN GOTO 99
END
'
'
'***************SUBROUTINES FOLLOW********************
'
'
 'SUBROUTINE 1 TO READ IN CALIB(I) FROM CAL. FILE (CALROSS RECORDS)
 1      REM
        PRINT "PLEASE ENTER CALIBRATION DISK INTO DRIVE B."
        INPUT "STRIKE [ENTER] WHEN READY",CRAP%
 555    FILES"B:*.*"
        INPUT "DESIRED FILE (WITHOUT FILENAME EXTENSION):",CAL$
        CAL$="B:"+CAL$+" RCAL" 'ENSURES CALROSS FILES,DRIVE B
        OPEN CAL$ FOR INPUT AS #1
        CLS: PRINT"CALIBRATION FACTORS READ FROM FILE":PRINT
        FOR I=1 TO 9
        INPUT #1,CALIB(I)
        PRINT CALIB(I)
        NEXT I
        INPUT "CONTINUE WITH EDIT (Y/N) ";ANS$
        IF ANS$="N" OR ANS$="n" THEN GO TO 555
        CLOSE#1
        RETURN
2 REMSUBROUTINE ENTRY POINT.
'SUBROUTINE 2 IS USED TO GET THE TRAVERSE COUNTERS.
OPEN F$ AS #1 LEN=768
        FIELD #1,192 AS A$,192 AS B$,192 AS C$,192 AS D$
REC=49'Last record contains desired data.
GET #1,REC'Read data.
FOR I=1 TO 96
K=I*2-1
DATACNT(I)=CVI(MID$(A$,K,2))
DATACNT(I+96)=CVI(MID$(B$,K,2))
DATACNT(I+192)=CVI(MID$(C$,K,2))
```

```
DATACNT(I+288)=CVI(MID$(D$,K,2))
NEXT I
'DATACNT NOW IS PRIMED WITH TRAVERSE COUNTERS.
'FILE IS LEFT OPEN, AS DATA WILL BE READ IN MAIN-LINE.
RETURN'Back to mainline.
REMEND OF SUBROUTINE TWO.
'
3REM Subroutine for reading unedited data.
'FILE IS ALREADY OPEN.
'
FOR CH%=1 TO 8'Loop through 8 channels.
REC=(CH%+(T%-1)*8)
GET #1,REC
PRINT CH%;" READ.."
FOR I=1 TO 96
K=I*2-1
DATAPTS(I,CH%)=CVI(MID$(A$,K,2))
DATAPTS(I+96,CH%)=CVI(MID$(B$,K,2))
DATAPTS(I+192,CH%)=CVI(MID$(C$,K,2))
DATAPTS(I+288,CH%)=CVI(MID$(D$,K,2))
NEXT I
FOR I=1 TO 384
     IF DATAPTS(I,CH%) > 32767 THEN DATAPTS(I,CH%)=DATAPTS(I,CH%)-65536
NEXT I
NEXT CH%
PRINT "TRAVERSE ";T%;" READ"
RETURN'Back to mainline.
REMEND OF SUBROUTINE THREE.
'
4REMSUBROUTINE FOR WRITING EDITED DATA.
IF T%=1 THEN GOSUB 40
PRINT "MADE TRAVERSE *** ";T%;" ***"
FOR H%=1 TO 8'Loop through all channels.
FOR I=1 TO 96
Z$=Z$+MKI$(INT(DATAPTS(I,H%)))
Y$=Y$+MKI$(INT(DATAPTS(I+96,H%)))
Q$=Q$+MKI$(INT(DATAPTS(I+192,H%)))
X$=X$+MKI$(INT(DATAPTS(I+288,H%)))
NEXT I
LSET L$=Z$:LSET M$=Y$:LSET N$=Q$:LSET O$=X$
REC=H%+(T%-1)*8
PUT #2,REC
Z$="":Y$="":Q$="":X$=""
PRINT " CHANNEL ";H%;" WRITTEN"
NEXT H%
RETURN'Back to mainline.
'
40REMSUBROUTINE TO OPEN EDITED DATA FILE.
FE$=F$+"ED"'Add file extension.
OPEN FE$ AS #2 LEN=768
FIELD #2,192 AS L$,192 AS M$,192 AS N$,192 AS O$
RETURN
REMEND OF SUBROUTINE 40 (CALLED BY SUBROUTINE 4).
'
5REM SUBROUTINE TO TACK ON RECORD 49 (DATACNT) TO EDITED
```

```
'DATA FILE.
FOR I=1 TO 96
K=I*2-1
Z$=Z$+MKI$(INT(DATACNT(I)))
Y$=Y$+MKI$(INT(DATACNT(I+96)))
Q$=Q$+MKI$(INT(DATACNT(I+192)))
X$=X$+MKI$(INT(DATACNT(I+288)))
NEXT I
LSET L$=Z$:LSET M$=Y$:LSET N$=Q$:LSET O$=X$
REC=49'DATACNT RECORD.
PUT #2,REC
Z$="":Y$="":Q$="":X$=""
RETURN
REMEND OF SUBROUTINE 5.
```

```
5 ' THIS PROGRAM IS PAUG9 RENAMED. IT WAS USED ON THE FIRST DATA BLOCK REGA
RDING    MY THESIS. THIS IS THE LAST VERSION OF THE PROGRAM WHICH DID NOT I
NCLUDE AN     EDITING SECTION.
10 DIM DATAPTS(400,8),C(8),STRIKE(5),HIT(4,10),VSUM(400),CPX(400),CPY(400),
 VCPX (400),VCPY(400),RVCP(400),VA(400),TYPE$(6),AXES(32),AXES2(32),H(10),S
DVSUM(400)   ,HS(5),FDVSUM(400),ZC(5)
20 CLS
30 ' FOLLOWING READS IN DATA USED FOR DRAWING AXES
40   FOR I=1 TO 32: ' FOR CPX,ETC.PLOT
50    READ AXES(I)
60   NEXT I
70   DATA 200,1700,2650,1700,2650,9900,2650,1700 :'vert.forces
80   DATA 3140,1700,5100,1700,4120,9900,4120,1700 :'CPX,CPY
90  DATA 5590,1700,7550,1700,6570,9900,6570,1700 :'VCPX,VCPY
100  DATA 8039,1700,9999,1700,9019,9900,9019,1700 :'RVCP,VA
110  FOR I=1 TO 24 : 'FOR FORCE PLOT
120    READ AXES2(I)
130  NEXT I
140  DATA 700,1700,3300,1700,3300,9900,3300,1700
150  DATA 3700,1700,6300,1700,5000,9900,5000,1700
160  DATA 6700,1700,9300,1700,8000,9900,8000,1700
170  VTHRESH=10: '10 LBS. MIN. BEFORE CPX,ETC. CALCULATED
180  TBASE=92.5:'BASE WIDTH INCHES
190  HEIGHT=45.6:'PERPENDICULAR HEIGHT AT APEX
200  SWITCH=200: ' THIS IS ABOUT 1 VOLTS ON A/D
210  C(1)=30000/250:C(2)=C(1):C(3)=C(1):C(4)=30000/70:C(5)=C(4):C(6)=C(4):C
(7)=1:    C(8)=1: 'CONSTANTS FOR VALUE DECODING
220  H(1)=-.97226:H(2)=.4462:H(3)=-.25667:H(4)=.15497:H(5)=-.092024
        :H(6)=.051667:H(7)=-.0268:H(8)=.01347:H(9)=-8.889998E-03: 'FOR DIG
 DIFF.
230  PRINT "PAPER PLOT PROGRAM COMPATIBLE WITH HP7470A PLOTTER "
240  PRINT "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ":PRINT:PRIN
T
250  PRINT "  THIS PROGRAM LOADS IN EDITED GAIT DATA, CALCULATES CENTRE OF
PRESSURES AND RELATED PARAMETERS,  AND  PROVIDES THE  OPTION OF A DATA PRIN
TOUT, AND SEVERAL FORMS OF DATA PLOTS"
260  PRINT
270  PRINT "BEFORE PROCEEDING FURTHER, PLEASE ENSURE THAT THE PLOTTER IS CO
NNECTED THROUGH  COM 2 AND THAT BOTH THE PRINTER AND PLOTTER ARE ON"
280  OPEN "COM2:2400,S,7,1,RS,CS,DS32000,CD"AS#3
290  PRINT:INPUT "Please enter desired data disk in drive B, strike [enter]
 when ready.",CRAP%
300  PRINT:PRINT"Now select one of these file reference numbers.";:FILES "B
:*.?ED"
310  PRINT:INPUT "FILE NUMBER [with out the ED suffix]: ",N$:B$="B:":SUFX$=
"ED":M$=B$+N$:E$=N$+SUFX$:F$=B$+E$:CLS
320  CLS:INPUT "SELECT A TRAVERSE [1,2,3,4,5,or 6] ? ",T%
330  PRINT:PRINT"PREPARE THE PLOTTER AND/OR THE PRINTER WHILE DATA IS BEING
 READ IN"
340  GOSUB 1580 'DATA GRAB SUBROUTINE
350  ' AT THIS POINT, ONE TRAVERSE OF DATA HAS BEEN LOADED
360  PRINT:PRINT "TRAVERSE "T%" HAS A DATA COUNT OF "COUNT
370  PRINT:INPUT " GIVE TRAVERSE DESCRIPTOR TYPE (e.g.WRLR )   ",TYPE$
380  PRINT:PRINT "ONE ETERNITY WHILE I DETERMINE CPX,CPY,ETC."
390  ' ROUTINE TO FIND CPX,ETC. FOLLOWS
```

```
400       FOR Z=COUNT TO 400:VSUM(Z)=0:CPX(Z)=0:CPY(Z)=0:NEXT:
               'CLEAR ARRAY  FROM LAST USE
410       FOR Q%=1 TO COUNT
420            IF DATAPTS(Q%,7)<SWITCH THEN DATAPTS(Q%,7)=0 ELSE
                    DATAPTS(Q%,7)=1: 'PUTS 1'S OR 0'S IN SWITCH CHANNEL 7
430            IF DATAPTS(Q%,8)<SWITCH THEN DATAPTS(Q%,8)=0 ELSE
                    DATAPTS(Q%,8)=1: 'PUTS 1'S OR 0'S IN SWITCH CHANNEL 8
440            VSUM(Q%)=DATAPTS(Q%,1)+DATAPTS(Q%,2)+DATAPTS(Q%,3)
450            ' following used for h.s. ident routine
460            IF Q%<20 THEN GOTO 520
470            FVSUM=DATAPTS(Q%+1,1)+DATAPTS(Q%+1,2)+DATAPTS(Q%+1,3): 'FUTURE
 SUM
480            PVSUM=DATAPTS(Q%-1,1)+DATAPTS(Q%-1,2)+DATAPTS(Q%-1,3): 'PAST S
UM
490            FDVSUM(Q%)=FVSUM-VSUM(Q%): 'PRESENT FIRST DIFFERENCE
500            PFDVSUM=VSUM(Q%)-PVSUM: 'PAST FIRST DIFFERENCE
510            SDVSUM(Q%)=FDVSUM(Q%)-PFDVSUM :' SECOND DIFFERENCE
520            IF VSUM(Q%)<VTHRESH THEN CPX(Q%)=0:CPY(Q%)=0:VCPX(Q%)=0:
                    VCPY(Q%)=0:RVCP(Q%)=0:VA(Q%)=0:GOTO 620
530            'BOARD MUST HAVE LOAD
540            'following finds CPX
550            NUM1=(DATAPTS(Q%,2)+2*DATAPTS(Q%,3))*TBASE/2
560            CPX(Q%)=NUM1/VSUM(Q%) :'inches -largest value 92.5
570            'cpx=0 at NE corner,increases going west
580            'following finds CPY,FCPY
590            NUM2=DATAPTS(Q%,2)*HEIGHT
600            CPY(Q%)=NUM2/VSUM(Q%) : 'inches-largest value is 45.6
610            'cpy=0 at N side,increases going south
620       NEXT Q%
630 ' ROUTINE TO FIND VCPX,VCPY,RVCP,VA USING F.I.R.DIFF.
640 ' TAKEN FROM  ONE-DIM.DIG.SIGNAL PROC.CHI-TSONG CHENG PAGE 204
650   FOR Z=COUNT-20 TO 400:VCPX(Z)=0:VCPY(Z)=0:RVCP(Z)=0: NEXT Z
            : 'CLEARS ARRAY FROM LAST USE
660   FOR Q%=10 TO (COUNT-10)
670     VCPX(Q%)=0:VCPY(Q%)=0:RVCP(Q%)=0:'CLEARS ARRAYS (STARTING VALUES
             FOR THIS LOOP)
680     IF VSUM(Q%-9)<VTHRESH OR VSUM(Q%+9)<VTHRESH THEN 760
690     FOR N=-9 TO 9
700      IF N=0 THEN GOTO 740
710      IF N<0 THEN CC=+H(-N) ELSE CC=-H(N):' SIGNS OPPOSITE OF IN BOOK
720      VCPX(Q%)=VCPX(Q%)+CC*CPX(Q%+N)
730      VCPY(Q%)=VCPY(Q%)+CC*CPY(Q%+N)
740     NEXT N
750     RVCP(Q%)=SQR(VCPX(Q%)^2+VCPY(Q%)^2)
760   NEXT Q%
770 ' ROUTINE TO FIND HEEL STRIKES BASED ON RVCP
780 J=15
790 FOR I=20 TO COUNT-10
800   DELTARVCP=RVCP(I+1)-RVCP(I)
810   IF J>=15 AND DELTARVCP>=.7 AND (RVCP(I+1)+RVCP(I+2)+RVCP(I+3)+RVCP(I+
4)+        RVCP(I+5))/5>=1 THEN PRINT "(RVCP) HEEL STRIKE AT:";I:J=0
820   J=J+1
830 NEXT I
840 PRINT
850 ' ROUTINE TO FIND HEEL STRIKES BASED ON VERTICAL FORCE DERIVATIVES
```

```
860 K=1:J=20: 'MIN OFFSET BETWEEN HEEL STRIKES
870 FOR Z%=25 TO (COUNT-25): 'SKIP FIRST AND LAST PORTIONS
880   J=J+1
890   FOR XX%=0 TO 3:SUM=SUM+RVCP(Z%+XX%):NEXT XX%
900   AVRVCP=SUM/4:SUM=0:IF AVRVCP < 1.5 THEN GOTO 920
910   IF SDVSUM(Z%)>1.5 AND SDVSUM(Z%+1)>1.5 AND((SDVSUM(Z%)+SDVSUM(Z%+1))/
2)>5         AND J>=20 THEN PRINT"(FORCE DER.) HEEL STRIKE  AT " Z% :HS(K)=Z
%:K=K+1:J=0
920  NEXT Z%
930 INPUT"FINISHED READING  HEELSTRIKES?",GARBAGE$
940  ' MAIN MENU FOLLOWS
950  CLS:PRINT "WORKING WITH DATA FOR TRAVERSE " T%:PRINT
960  PRINT"SELECT ONE OF:    (1) DATA PRINTOUT     (2) FORCES PLOT (V,M/L,P/A
)                                    (3) PLOT CPX,ETC.     (4) PECK
                                           (5) GET NEW TRAVERSE (6) GET NEW FILE
"
970  INPUT CD$
980  IF CD$="1" THEN GOTO 1260: ' PRINT ROUTINE
990  IF CD$="5" THEN FLAG=0:GOTO 320 :'NEW TRAVERSE
1000  IF CD$="6" THEN FLAG=0:GOTO 290 :'NEW FILE
1010  IF FLAG=1 GOTO 1210
1020  FLAG=1
1030  ' SOME PLOT ROUTINE MUST HAVE BEEN SELECTED OR ELSE AN INVALID ANSWER
1040  PRINT:PRINT:INPUT "DO YOU WANT TO GIVE ME THE TEMPORAL EVENTS? (Y/N)
",TE$       :PRINT
1050  IF TE$="N" OR TE$="n" THEN GOTO 1210 :'SKIP FOLLOWING IF N/A
1060  IF TE$<>"Y" AND TE$<>"y" THEN GOTO 1040:'TRAP
1070 'ROUTINE TO INPUT TEMPORAL EVENTS FOLLOWS
1080  PRINT:INPUT;"NUMBER OF RIGHT HEEL STRIKES=";STRIKE(1)
1090  PRINT:INPUT;"NUMBER OF LEFT HEEL STRIKES=";STRIKE(2)
1100  PRINT:INPUT;"NUMBER OF RIGHT TOE OFFS=";STRIKE(3)
1110  PRINT:INPUT;"NUMBER OF LEFT TOE OFFS=";STRIKE(4)
1120  FOR S=1 TO 4
1130   IF STRIKE(S)=0 THEN GOTO 1200
1140   FOR P=1 TO STRIKE(S)
1150     IF S=1 THEN PRINT:PRINT,"PT. OF RIGHT HEEL STRIKE NUMBER: ";P;" ="
          :INPUT;HIT(1,P)
1160     IF S=2 THEN PRINT:PRINT,"PT. OF LEFT HEEL STIKE NUMBER: ";P;" =":
          INPUT;HIT(2,P)
1170     IF S=3 THEN PRINT:PRINT,"PT. OF RIGHT TOE OFF NUMBER: ";P;" =":
          INPUT;HIT(3,P)
1180     IF S=4 THEN PRINT:PRINT,"PT OF LEFT TOE OFF NUMBER: ";P;" =":
          INPUT;HIT(4,P)
1190    NEXT P
1200  NEXT S
1210  PRINT:PRINT
1220  IF CD$="2" THEN GOTO 3580 :'FORCES PLOT
1230  IF CD$="3" THEN GOTO 1510 :'CPX,ETC.PLOT
1240  IF CD$="4" THEN GOTO 3190 :'PECK PLOT
1250  GOTO 950: 'TRAP
1260 ' PRINT ROUTINE FOLLOWS*************************************
1270 BEEP:INPUT" PRESS RETURN IF PRINTER IS READY",POOP$
1280 CLS:PRINT:INPUT "SELECT ONE   (1)PRINT ALL DATA F/L   (2)PRINT SECTION
S ONLY  "      ,PD$:IF PD$="1"THEN 1370 ELSE IF PD$="2"THEN 1300 ELSE 1280
1290 ' ABBREVIATED PRINT OUT ROUTINE
```

```
1300 PRINT:INPUT"GIVE ME THE NO.OF ZONES AND THE ZONE RANGE (NO.PTS EITHER
SIDE OF CENTER PT.)          ",NZ,ZR
1310 ON NZ GOTO 1320,1330,1340,1350,1360
1320 INPUT "GIVE ME THE ZONE CENTER        ",ZC(1):GOTO 1380
1330 PRINT "GIVE ME THE "NZ;:INPUT " ZONE CENTERS      ",ZC(1),ZC(2):GOTO 13
80
1340 PRINT "GIVE ME THE "NZ;:INPUT " ZONE CENTERS      ",ZC(1),ZC(2),ZC(3)
        :GOTO 1380
1350 PRINT "GIVE ME THE "NZ;:INPUT " ZONE CENTERS      ",ZC(1),ZC(2),ZC(3),Z
C(4)        :GOTO 1380
1360 PRINT "GIVE ME THE "NZ;:INPUT " ZONE CENTERS      ",ZC(1),ZC(2),ZC(3),Z
C(4),       ZC(5):GOTO 1380
1370 NZ=1:FIR=1:LAS=COUNT
1380 LPRINT "FILENAME: "N$" TRAVERSE NO: "T%" TYPE: "TYPE$
        :LPRINT:LPRINT
1390 LPRINT"PT.    L    R     VSUM  M/L   P/A     FDVSUM  SDVSUM  VCPX    VCP
Y    RVCP  ":LPRINT
1400 FOR T=1 TO NZ
1410 IF PD$="1" THEN 1420 ELSE FIR=ZC(T)-ZR:LAS=ZC(T)+ZR
1420 FOR Q%= FIR TO LAS
1430    LPRINT Q% TAB(6)DATAPTS(Q%,7)TAB(11)DATAPTS(Q%,8)TAB(16)INT(VSUM(Q%
))TAB(22)INT(DATAPTS(Q%,4)+DATAPTS(Q%,5))TAB(28)INT(DATAPTS(Q%,6))TAB(33);:
LPRINT USING"  ###.##";FDVSUM(Q%),SDVSUM(Q%),VCPX(Q%),VCPY(Q%),RVCP(Q%)
1440 NEXT Q%
1450 LPRINT
1460 NEXT T
1470 GOTO 940: 'MAIN MENU
1480 ' END OF PRINTOUT SECTION
1490 '**********************************************************
1500 '
1510 ' CPX,ETC. PLOT ROUTINE FOLLOWS ****************************
1520 BEEP:INPUT"PRESS RETURN IF PLOTTER IS READY",POOP$
1530 GOSUB 2760:'AXES AND LABELS SUBROUTINE
1540 GOSUB 1780 :'PLOTTING ROUTINE
1550 GOTO 950 : 'MAIN MENU
1560 '
1570 '************************************************************
1580 ' DATAGRAB SUBROUTINE
1590 ' THIS ROUTINE IS MODIFIED TO DECODE VALUES ON READ-IN
1600 OPEN F$ AS #1 LEN=768
1610 FIELD #1,192 AS A$,192 AS B$,192 AS C$,192 AS D$
1620 GET #1,49
1630 K=T%*2-1
1640 COUNT= CVI(MID$(A$,K,2))
1650 FOR CH%= 1 TO 8
1660    ' ARRAY C(CH%) HOLDS THE DECODING CONSTANTS
1670   GET #1,(CH% + (T%-1)*8)
1680   FOR I=1 TO 96
1690      K=I*2-1
1700      DATAPTS(I,CH%)=CVI(MID$(A$,K,2)):DATAPTS(I,CH%)=DATAPTS(I,CH%)/C(
CH%)
1710      DATAPTS(I+96,CH%)=CVI(MID$(B$,K,2)):DATAPTS(I+96,CH%)=
              DATAPTS(I+96,CH%)/C(CH%)
1720      DATAPTS(I+192,CH%)=CVI(MID$(C$,K,2)):DATAPTS(I+192,CH%)=
              DATAPTS(I+192,CH%)/C(CH%)
```

```
1730       DATAPTS(I+288,CH%)=CVI(MID$(D$,K,2)):DATAPTS(I+288,CH%)=
                DATAPTS(I+288,CH%)/C(CH%)
1740   NEXT I
1750 NEXT CH%
1760 CLOSE 1
1770 RETURN
1780 '****************************************************************
1790 ' CPX.ETC. PLOTTING SUBROUTINE
1800 'PLOTS VSUM
1810 Y=1700
1820 PRINT #3,"PU;SP2;PU 2650 ";Y;";PD;"
1830 INCR=8200/COUNT: 'time interval depends on count
1840 VSCALE1=(2650-200)/250
1850 FOR I= 1 TO COUNT
1860    Y=Y+INCR
1870    X=2650-VSUM(I)*VSCALE1
1880    PRINT #3,"PA ";X;" ";Y;";";
1890 NEXT I
1900 'PLOTS CPX
1910 Y=1700
1920 PRINT #3,"PU;SP1;PU 4120 ";Y;";PD;"
1930 INCR=8200/COUNT
1940 SCALEX=(4120-3140)/92.5
1950 FOR I= 1 TO COUNT
1960    Y=Y+INCR
1970    X=4120-CPX(I)*SCALEX
1980    PRINT #3,"PA ";X;" ";Y;";";
1990 NEXT I
2000 '
2010 'PLOTS CPY
2020  POSI = 6570
2030  Y=1700
2040  SCALEY=(6570-5590)/45.6
2050  PRINT #3,"PU ";POSI;" ";Y;";PD;"
2060  FOR I=1 TO COUNT
2070     Y=Y+8200/COUNT
2080     X=POSI-CPY(I)*SCALEY
2090     PRINT #3,"PA ";X;" ";Y;";";
2100 '
2110  NEXT I
2120 '
2130 'PLOTS RVCP
2140  POSI = 9019
2150  Y=1700
2160  SCRVCP=2*SCALEX
2170  PRINT #3,"PU ";POSI;" ";Y;";PD;"
2180  FOR I=1 TO COUNT
2190    Y=Y+8200/COUNT
2200    X=POSI-RVCP(I)*SCRVCP :' 2 x's value for cpx
2210    PRINT #3,"PA ";X;" ";Y;";";
2220  NEXT I
2230 '
2240 PRINT #3,"SP 2;"  'CHANGE TO PEN TWO
2250 '
2260 'PLOTS VA, VCPY, AND SDVUM
```

```
2270 POSI = 4120
2280 SCVX=SCALEX*2
2290 SCVY=SCALEY*2
2300 FOR GR =1 TO 3
2310   Y=1700
2320   PRINT #3,"PU ";POSI;" ";Y;";PD;"
2330   FOR I=1 TO COUNT
2340     Y=Y+8200/COUNT
2350     IF GR = 1 THEN X=POSI-VCPX(I)*SCVX: '2 x's value for cpx
2360     IF GR = 2 THEN X=POSI-VCPY(I)*SCVY:'2 x's value for cpy
2370     IF GR = 3 THEN X=POSI-SDVSUM(I)*(9019-8039)/90
2380     PRINT #3,"PA ";X;" ";Y;";";
2390    NEXT I
2400  POSI=POSI+2449.5
2410  NEXT GR
2420 '
2430 '
2440 'FOLLOWING DRAWS TEMPORAL LINES (IF GIVEN ) ON CPX,ETC. PLOT
2450 '
2460 PRINT #3,"PU;SP1;PU 0,0;"
2470 IF TE$="n" OR TE$="N" THEN GOTO 2620
2480 PRINT:INPUT"DRAW TEMPORAL EVENTS (Y/N) "; D3$:IF D3$<>"Y" AND D3$<>"y"
 AND D3$<>     "N" AND D3$<>"n" THEN GOTO 2480 :'TRAP
2490 IF D3$="N" OR D3$="n" THEN GOTO 2620
2500 PRINT #3,"SP 2;DI 0,1;SI .15 .2;DT";+CHR$(1);";"
2510 FOR S=1 TO 4
2520   IF STRIKE(S)=0 THEN GOTO 2590' NEXT STRIKE
2530   FOR P=1 TO STRIKE(S)
2540     IF S=1 THEN X=2650-VSCALE1*VSUM(HIT(1,P)):Y=1700+HIT(1,P)*8200/COU
NT             :PRINT #3,"PU ";2775;" ";Y-190;";PD;LB RH";+CHR$(1);";PU ";X;
" ";Y;";PD 9999 ";Y;";PU;"
2550     IF S=2 THEN X=2650-VSCALE1*VSUM(HIT(2,P)):Y=1700+HIT(2,P)*8200/COU
NT             :PRINT #3,"PU ";2775;" ";Y-190;";PD;LB LH";+CHR$(1);";PU ";X;
" ";Y;";PD 9999 ";Y;";PU
2560     IF S=3 THEN X=2650-VSCALE1*VSUM(HIT(3,P)):Y=1700+HIT(3,P)*8200/COU
NT             :PRINT #3,"PU ";2900;" ";Y-250;";PD;LB RTO";+CHR$(1);";PU ";X
;" ";Y;";PD 9999 ";Y;";PU;"
2570     IF S=4 THEN X=2650-VSCALE1*VSUM(HIT(4,P)):Y=1700+HIT(4,P)*8200/COU
NT             :PRINT #3,"PU ";2900;" ";Y-250;";PD;LB LTO";+CHR$(1);";PU ";X
;" ";Y;";PD 9999 ";Y;";PU;"
2580   NEXT P
2590 NEXT S
2600 PRINT #3,"PU 0 0;"
2610 '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ BEGINNING OF VSUM SHIFT
2620 PRINT:INPUT"DO YOU WANT VSUM SHIFT? ",V$:IF V$="N" OR V$="n" THEN GOTO
 2740 ELSE IF V$<>"Y" AND V$<>"y" THEN GOTO 2610
2630 PRINT:INPUT"SOURCE POINT: ",SOURCE:PRINT:INPUT"DESTINATION POINT: ",DE
ST
2640 INCR=8200/COUNT:VSCALE1=(2650-200)/250
2650 Y=1700+DEST*INCR:X=2650-VSUM(SOURCE)*VSCALE1
2660 PRINT #3,"SP1;LT 6;PU ";X;" ";Y;"; PD;"
2670 FOR I=SOURCE+1 TO COUNT
2680    Y=Y+INCR
2690    X=2650-VSUM(I)*VSCALE1
2700    PRINT #3,"PD ";X;" ";Y;";"
```

```
2710 NEXT I
2720 '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@   END OF VSUM SHIFT
2730 PRINT #3,"PU;SP1;PU 0,0;"
2740 RETURN
2750 ' **************************************************************
2760 ' AXES AND LABELS SUBROUTINE FOR CPX,ETC.
2770 PRINT #3,"IN;SC0,9999,0,9999;SP1;"
2780 PRINT #3,"DT";+CHR$(1);";DI 0,1;PU 0,2000;LB FILENAME:";N$+"  TRAVERSE
:";STR$(T%)+"   TYPE: "+TYPE$+CHR$(1);";"
2790 PRINT #3,"DT";+CHR$(1);";DI 0,1;PU 0,8000;TYPE";TYPE$+CHR$(1);";"
2800 FOR I= 1 TO 32 STEP 4' DRAWS AXES
2810    X=INT(AXES(I))
2820    Y=INT(AXES(I+1))
2830    PRINT #3,"LT;PU",X,",",Y,";PD;"
2840    X=INT(AXES(I+2))
2850    Y=INT(AXES(I+3))
2860    PRINT #3,"PA",X,",",Y,";"
2870 NEXT I
2880 '
2890 'DRAWS LABELS
2900  PRINT #3,"DI 0,1;PU 700,7000;LB VERTICAL FORCES";+CHR$(1)
2910 PRINT #3,";PU 3180,7000;LB C.P.X";+CHR$(1)
2920 PRINT #3,";PU 5600,7000;LB C.P.Y.";+CHR$(1)
2930 PRINT #3,";PU 8100,7000;LB RESULTANT VEL.(C.P.)";+CHR$(1)
2940 PRINT #3,"SP 2;PU 3380,7000;LB VELOCITY C.P.X.";+CHR$(1)
2950 PRINT #3,";PU 5800,7000;LB VELOCITY C.P.Y.";+CHR$(1)
2960 PRINT #3,"SP 1;"
2970 '
2980 ' FOLLOWING MARKS SCALE VALUES ON AXES
2990 '*VERTICAL FORCES *
3000 X=2650
3010 FOR I=0 TO 250 STEP 50
3020    PRINT #3,"PU ";X;" 1000;LB";STR$(I)+CHR$(1);";"
3030    IF I>225 THEN GOTO 3090
3040       FOR J=1 TO 5
3050           PRINT #3,"PU ";X;" 1700;PD;XT;"
3060          X=X-2450/25
3070       NEXT J
3080 NEXT I
3090 PRINT #3,"PU 200 1700;PD;XT;"
3100  '*CPX *
3110 PRINT #3,"PU 5100  850;LB";STR$(-92.5)+CHR$(1);";PU 3140 850;LB";STR$(
92.5)+CHR$(1);";"
3120 PRINT #3,"PU 7550 850;LB";STR$(-45.6)+CHR$(1);";PU 5590 850;LB";STR$(4
5.6)+CHR$(1);";"
3130 PRINT #3,"PU 9999 850;LB";STR$(-92.5)+CHR$(1);";PU 8039 850;LB";STR$(9
2.5)+CHR$(1);";"
3140 PRINT #3,"PU 7550 200;LB /3";+CHR$(1);";PU 5590 200;LB /3";+CHR$(1);";
"
3150 PRINT #3,"PU 5100 200;LB /3";+CHR$(1);";PU 3140 200;LB /3";+CHR$(1);";
"
3160 RETURN
3170 '
3180  '**********************************************
3190 'PECK ROUTINE
```

```
3200 BEEP:INPUT"PRESS RETURN IF PLOTTER IS READY",POOP$
3210 PRINT #3,"IN;SC0,9999,0,9999;SP1;"
3220 PRINT #3,"DT";+CHR$(1);";DI 0,1;PU 0,2000;LB FILENAME:";N$+"  TRAVERSE
:";STR$(T%)+"   TYPE: "+TYPE$+CHR$(1);";"
3230 PRINT #3,"DT";+CHR$(1);";DI 0,1;PU 0,8000;TYPE";TYPE$+CHR$(1);";"
3240 PRINT #3,"PU 6000 10000;PD 4000 10000;"
3250 IF LEFT$(TYPE$,1)="W" OR LEFT$(TYPE$,1)="w" THEN
          PRINT #3,"PU 4120 9880;PD 4000 10000;PD 4120 10120;" ELSE
               PRINT #3,"PU 5880 9880;PD 6000 10000;PA 5880 10120;":' DRAWS
 ARROWHEADS
3260 PRINT #3,"DI 0 1;"
3270 PRINT #3,"IN;IP 1150 1825 9150 5825;SC 0.000 8.000 0.000 4.000;SP 1;PA
 0 0;PD 8 0;PA 4 4 0 0;PU 0.5 3;PD 1.5 3;PU 1 3.5;PD 1 2.5;" :' DRAWS TRIAN
GLE
3280 PRINT #3,"DT";+CHR$(1);";"
3290 PRINT #3,"DI 0,1;PU 1 2.25;LB N";+CHR$(1);";"
3300 PRINT #3,"DI 0,1;PU 1 3.5;LB S";+CHR$(1);";"
3310 PRINT #3,"DI 0,1;PU 0.4 2.875;LB E";+CHR$(1);";"
3320 PRINT #3,"DI 0,1;PU 1.65 2.875;LB W";+CHR$(1);";"
3330 PRINT #3,"SC 0.000 92.50 0.000 45.60;"
3340 FOR J=1 TO COUNT
3350   X=CPX(J):Y=CPY(J)
3360 IF (Y=0) OR (X=0) THEN GOTO 3380: 'DON'T PLOT ZEROS
3370 PRINT #3,"PU;PA ";X;" ";Y;";PD;"
3380 NEXT J
3390 PRINT #3,"DF;DT";+CHR$(1);";DI 0,1;PU 1030 1825;LB 0,0";+CHR$(1);";PU
9270 1825;LB 92.5,0";+CHR$(1);";PU 5150 5945;LB 46.25,45.6";+CHR$(1);";PU;"
3400 PRINT #3,"PU;SP1;PU 0,0;"
3410 ' PLOTS HEEL STRIKES & TOE OFFS ON PECK
3420 IF TE$="n" OR TE$="N" THEN GOTO 3570
3430 PRINT:INPUT"DRAW TEMPORAL EVENTS ON PECK (Y/N) "; D$:IF D$<>"Y" AND D$
<>"y" AND D$<>"N"       AND D$<>"n" THEN GOTO 3430
3440 IF (D$="N" OR D$="n") OR (TE$<>"Y" AND TE$<>"y") THEN GOTO 3570
3450  PRINT #3,"IN;IP 1150 1825 9150 5825;SC 0.000 92.50 0.000 45.60;"
3460 PRINT #3,"SP 2;DI 0,1;DT";+CHR$(1);";"
3470 FOR S=1 TO 4
3480   IF STRIKE(S)=0 THEN GOTO 3550
3490   FOR P=1 TO STRIKE(S)
3500     IF S=1 THEN X=CPX(HIT(1,P)):Y=CPY(HIT(1,P)):PRINT #3,"PU ";X;" ";Y
+1;";PD ";X;" ";Y+6;";LB RH";+CHR$(1);";"
3510     IF S=2 THEN X=CPX(HIT(2,P)):Y=CPY(HIT(2,P)):PRINT #3,"PU ";X;" ";Y
-1;";PD ";X;" ";Y-6;";PU ";X;" ";Y-10;";PD;LB LH";+CHR$(1);";"
3520     IF S=3 THEN X=CPX(HIT(3,P)):Y=CPY(HIT(3,P)):PRINT #3,"PU ";X;" ";Y
+1;";PD ";X;" ";Y+3;";LB RTO";+CHR$(1);";"
3530     IF S=4 THEN X=CPX(HIT(4,P)):Y=CPY(HIT(4,P)):PRINT #3,"PU ";X;" ";Y
-1;";PD ";X;" ";Y-3;";PU ";X;" ";Y-8;";LB LTO";+CHR$(1);";"
3540   NEXT P
3550 NEXT S
3560 PRINT #3,"PU;SP1;PU 0,0;"
3570 GOTO 960: 'MAIN MENU
3580 ' FORCES PLOT FOLLOWS ******************************************
3590 INPUT"SELECT  (1) ALL FORCES PLOT   (2) VERTICAL ONLY PLOT  ";PFLAG
3600 IF PFLAG<>1 AND PFLAG<>2 THEN 3590: 'TRAP
3610 BEEP:INPUT"If the plotter is ready press [ENTER] key ";Z$
3620 PRINT #3,"IN;SC0,9999,0,9999;SP1;"
```

```
3630 PRINT #3,"DT";+CHR$(1);";DI 0,1;PU 300,5000;LB TRAVERSE ";STR$(T%)+CHR
$(1);";"
3640 PRINT #3,"DI 0,1;PU 0,5000;LB FILENAME ";N$+CHR$(1);";PU 700,6000;LB V
ERTICAL FORCES";+CHR$(1)
3650 PRINT #3,";PU 3700,6000;LB MEDIO LATERAL FORCES";+CHR$(1);";PU 6700,56
00;LB ANTERIOR POSTERIOR FORCES";+CHR$(1);";"
3660 IF PFLAG=2 THEN QQ=8 ELSE QQ=24
3670 FOR I= 1 TO QQ STEP 4 'DRAWS AXES FOR FORCE PLOT
3680   X=INT(AXES2(I))
3690   Y=INT(AXES2(I+1))
3700   PRINT #3,"LT;PU",X,",",Y,";PD;"
3710   X=INT(AXES2(I+2))
3720   Y=INT(AXES2(I+3))
3730   PRINT #3,"PA ";X;" ";Y;";";
3740 NEXT I
3750 '*VERTICAL FORCES *******************  MAKES TICK MARKS
3760 X=3300
3770 FOR I=0 TO 250 STEP 50
3780   PRINT #3,"PU ";X;" 1000;LB";STR$(I)+CHR$(1);";"
3790   IF I>225 THEN GOTO 3850
3800   FOR J=1 TO 5
3810      PRINT #3,"PU ";X;" 1700;PD;XT;"
3820      X=X-(3300-700)/25
3830   NEXT J
3840 NEXT I
3850 PRINT #3,"PU 700 1700;PD;XT;"
3860 IF PFLAG=2 THEN 4020
3870  '* MEDIO LATERAL *
3880 X=6300
3890 PRINT #3,"PU 6300 1000;LB";STR$(-40)+CHR$(1);";PU 3700 1000;LB";STR$(4
0)+CHR$(1);";"
3900 FOR I=0 TO 16
3910     PRINT #3,"PU ";X;" 1700;PD;XT;"
3920     X=X-(6300-3700)/16 : 'MARKS EVERY 10 LBS.
3930 NEXT I
3940 '* ANTERIOR POSTERIOR *
3950 X=9300
3960 PRINT #3,"PU 9300 1000;LB";STR$(-40)+CHR$(1);";PU 6700 1000;LB";STR$(4
0)+CHR$(1);";"
3970 FOR I=0 TO 16
3980     PRINT #3,"PU ";X;" 1700;PD;XT;"
3990     X=X-(9300-6700)/16 : 'MARKS EVERY 10 LBS.
4000 NEXT I
4010 '
4020 '$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ PLOTTING ROUTINE FOR FORCE
4030 Y=1700
4040 PRINT #3,"PU;SP2;PU 3300,",Y,";PD;"
4050 INCR=8200/COUNT
4060 VSCALE2=(3300-700)/250
4070 FOR I= 1 TO COUNT
4080     Y=Y+INCR
4090     X=3300-VSUM(I)*VSCALE2
4100     PRINT #3,"PA ";X;" ";Y;";";
4110 NEXT I
4120 IF PFLAG=2 THEN 4270
```

```
4130 '
4140 POSI = 5000
4150 FOR GR =2 TO 3
4160    Y=1700
4170 SCMLPA=(5000-3700)/40
4180 PRINT #3,"PU",POSI,",",Y,";PD;"
4190 FOR I=1 TO COUNT
4200   Y=Y+INCR
4210   IF GR = 2 THEN X=POSI-(DATAPTS(I,4)+DATAPTS(I,5))*SCMLPA
4220   IF GR = 3 THEN X=POSI-DATAPTS(I,6)*SCMLPA
4230   PRINT #3,"PA ";X;" ";Y;";";
4240   NEXT I
4250   POSI=POSI+3000
4260 NEXT GR
4270 PRINT #3,"PU;SP1;PU 0,0;"
4280 'FOLLOWING DRAWS TEMPORAL LINES (IF GIVEN ) ON FORCE PLOT
4290 IF TE$="n" OR TE$="N" THEN GOTO 4430
4300 PRINT:INPUT"DRAW TEMPORAL EVENTS (Y/N) "; D3$:IF D3$<>"Y" AND D3$<>"y"
 AND       D3$<> "N" AND D3$<>"n" THEN GOTO 4300 :'TRAP
4310 IF D3$="N" OR D3$="n" THEN GOTO 4430
4320 PRINT #3,"DI 0,1;SI .15 .2;DT";+CHR$(1);";"
4330 FOR S=1 TO 4
4340   IF STRIKE(S)=0 THEN GOTO 4410' NEXT STRIKE
4350   FOR P=1 TO STRIKE(S)
4360     IF S=1 THEN X=3300-VSCALE2*VSUM(HIT(1,P)):Y=1700+HIT(1,P)*8200/COU
NT:PRINT #3,"PU ";3425;" ";Y-190;";PD;LB RH";+CHR$(1);";PU ";X;" ";Y;";PD 9
999 ";Y;";PU;"
4370     IF S=2 THEN X=3300-VSCALE2*VSUM(HIT(2,P)):Y=1700+HIT(2,P)*8200/COU
NT:PRINT #3,"PU ";3425;" ";Y-190;";PD;LB LH";+CHR$(1);";PU ";X;" ";Y;";PD 9
999 ";Y;";PU;"
4380     IF S=3 THEN X=3300-VSCALE2*VSUM(HIT(3,P)):Y=1700+HIT(3,P)*8200/COU
NT:PRINT #3,"PU ";3550;" ";Y-250;";PD;LB RTO";+CHR$(1);";PU ";X;" ";Y;";PD
9999 ";Y;";PU;"
4390     IF S=4 THEN X=3300-VSCALE2*VSUM(HIT(4,P)):Y=1700+HIT(4,P)*8200/COU
NT:PRINT #3,"PU ";3550;" ";Y-250;";PD;LB LTO";+CHR$(1);";PU ";X;" ";Y;";PD
9999 ";Y;";PU;"
4400   NEXT P
4410 NEXT S
4420 PRINT #3,"PU 0 0;"
4430 CLS:INPUT "SELECT (T)TEMPORAL SHIFT   (F)FRACTIONAL SPLIT  (M) MAIN ME
NU ",        DI$
4440 IF DI$="T" OR DI$="t" THEN 4450 ELSE IF DI$="F" OR DI$="f" THEN 4960
        ELSE IF DI$="M" OR DI$="m" THEN 940 ELSE 4430
4450 '@@@@@@@@@@@@@@@@@@@ BEGINNING OF TEMPORAL VSUM SHIFT
4460 'ROUTINE TO SHIFT OVER TRAILING EDGE
4470 PRINT:INPUT"SOURCE POINT: ",SOURCE:PRINT:INPUT"DESTINATION POINT: ",DE
ST
4480 INCR=8200/COUNT:VSCALE2=(3300-700)/250
4490 Y=1700+DEST*INCR:X=3300-VSUM(SOURCE)*VSCALE2
4500 PRINT #3,"SP1;LT 6;PU ";X;" ";Y;";PD;"
4510 FOR I=SOURCE+1 TO COUNT
4520    Y=Y+INCR
4530    X=3300-VSUM(I)*VSCALE2
4540    PRINT #3,"PD ";X;" ";Y;";"
4550 NEXT I
```

```
4560 IF PFLAG=2 THEN 4720
4570 ' FOLLOWING FOR TEMP SHIFT ON M/L AND P/A
4580 Y=1700+DEST*INCR:X=5000-(DATAPTS(SOURCE,4)+DATAPTS(SOURCE,5))*SCMLPA
4590 PRINT #3,"PU ";X;" ";Y;";PD;"
4600 FOR I=SOURCE+1 TO COUNT
4610    Y=Y+INCR
4620    X=5000-(DATAPTS(I,4)+DATAPTS(I,5))*SCMLPA
4630    PRINT #3,"PD ";X;" ";Y;";"
4640 NEXT I
4650 Y=1700+DEST*INCR:X=8000-DATAPTS(SOURCE,6)*SCMLPA
4660 PRINT #3,"PU ";X;" ";Y;";PD;"
4670 FOR I=SOURCE+1 TO COUNT
4680    Y=Y+INCR
4690    X=8000-DATAPTS(I,6)*SCMLPA
4700    PRINT #3,"PD ";X;" ";Y;";"
4710 NEXT I
4720 'ROUTINE TO SUBTRACTIVELY DETERMINE AND PLOT LEADING EDGE OTHER FOOT
4730 INCR=8200/COUNT:VSCALE2=(3300-700)/250
4740 Y=1700+DEST*INCR:X=3300-(VSUM(DEST)*VSCALE2-VSUM(SOURCE)*VSCALE2)
4750 PRINT #3,"SP1;LT 6;PU ";X;" ";Y;";PD;"
4760 FOR I=1 TO 10
4770    Y=Y+INCR
4780    X=3300-(VSUM(DEST+I)*VSCALE2-VSUM(SOURCE+I)*VSCALE2)
4790    PRINT #3,"PD ";X;" ";Y;";"
4800 NEXT I
4810 PRINT #3,"PU;SP1;PU 0,0;LT;"
4820 PRINT:INPUT"DO YOU WANT T/E OF NEXT STEP PLOTTED ? ",V$:IF V$="N" OR V
$="n" THEN GOTO 4930 ELSE IF V$<>"Y" AND V$<>"y" THEN GOTO 4820
4830 INPUT" GIVE ME  MID , START ",MID,START
4840 INCR=8200/COUNT:VSCALE2=(3300-700)/250
4850 Y=1700+MID*INCR:X=3300-VSUM(MID)*VSCALE2
4860 PRINT #3,"SP1;LT 6;PU ";X;" ";Y;";PD;"
4870 FOR I=1 TO 10
4880    Y=Y+INCR
4890    X=3300-(VSUM(MID+I)*VSCALE2-VSUM(START+I)*VSCALE2)
4900    PRINT #3,"PD ";X;" ";Y;";"
4910 NEXT I
4920 PRINT #3,"PU;SP1;PU 0,0;LT;"
4930 GOTO 940 :'MAIN MENU
4940 '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@   END OF VSUM SHIFT
4950 '
4960 ' FRACTIONAL DETERMINATION OF VERT FORCES
4970 INPUT"GIVE ME H.S AND TOE OFF TIMES ",HS,TOF
4980 HSCPX=CPX(HS):TOCPX=CPX(TOF):SL=ABS(TOCPX-HSCPX)
4990 INCR=8200/COUNT:VSCALE2=(3300-700)/250
5000 'ROUTINE TO PLOT LEADING EDGE
5010 Y=1700+HS*INCR:X=3300
5020 PRINT #3,"SP1;LT 6;PU ";X;" ";Y;";PD;"
5030 FOR I=HS+1 TO TOF
5040    Y=Y+INCR
5050    RHL=ABS(CPX(I)-HSCPX)/SL:RTL=1-RHL
5060    AHL=RHL*VSUM(I)
5070    X=3300- AHL*VSCALE2
5080    PRINT #3,"PD ";X;" ";Y;";"
5090 NEXT I
```

```
5100 'ROUTINE TO PLOT TRAILING EDGE
5110 Y=1700+HS*INCR:X=3300-VSUM(HS)*VSCALE2
5120 PRINT #3,"SP1;LT 6;PU ";X;" ";Y;";PD;"
5130 FOR I=HS+1 TO TOF
5140    Y=Y+INCR
5150    RHL=ABS(CPX(I)-HSCPX)/SL:RTL=1-RHL
5160    ATL=RTL*VSUM(I)
5170    X=3300- ATL*VSCALE2
5180    PRINT #3,"PD ";X;" ";Y;";"
5190 NEXT I
5200 PRINT #3,"PU;SP1;PU 0,0;LT;"
5210 GOTO 950: 'MAIN MENU
```