Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle
Pavement Crack Detection With Lasers

5. Report Date
January 1990

6. Performing Organization Code

7. Author(s)
Roger S. Walker, Chuying Kuo

8. Performing Organization Report No.
1141-2F

9. Performing Organization Name and Address
The University of Texas at Arlington
Arlington Texas 76019

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
Study # 8-18-89-1141

12. Sponsoring Agency Name and Address
Texas State Department of Highways and Public Transportation, D-10 Research
P. O. Box 5051, Austin, Texas 78763

13. Type of Report and Period Covered
Final

14. Sponsoring Agency Code

15. Supplementary Notes
Study done in cooperation with US Dept. of Transportation, Federal Highway Administration

16. Abstract

This report provides the final details on Research Study 8-18-88-1141. the research was initiated to investigate the capability of using lasers for crack detection in pavements. If such a capability could be developed it would be used to aid in obtaining and evaluating pavement distress and cracking information for the State's pavement management procedures, or PES.

The research effort has involved three stages. The first two stages were to determine the crack detection capabilities of the laser probes, used on the Surface Dynamics Profilometer (SDP). The SDP is owned by the State and used for road profile measurements. After experiments indicated that these probes could be used for such detection, a system was developed to further study this capability and to determine how it could be used to implement an automated high speed crack identification system. The results of the first two stages, and a complete description of the research effort is provided in Research Report 1141-1, 'The Use of Lasers For Pavement Crack Detection.' The third stage has been the development of a system which could be used for implementation of such a system so it's usefulness for P.E.S. data collection activities can be determined. This research report describes the final research effort.

17. Key Words	18. Distribution Statement
Surface Dynamics Profilometer (SDP) Lasers, Pavement Distress Measurements, Pavement Crack Identification and Recording.	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified		

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

PAVEMENT CRACK DETECTION WITH LASERS

by

Roger S. Walker
Chuying Kuo

The University of Texas at Arlington

Research Report 1141-2F

Crack Identification Using Lasers

Research Project 8-18-88-1141

conducted for

Texas State Department of Highways and Public Transportation

in cooperation with the
U.S. Department of Transportation
Federal Highway Administration

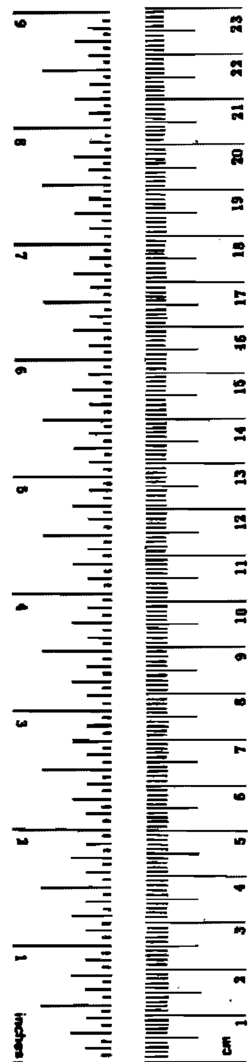
January 1990

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

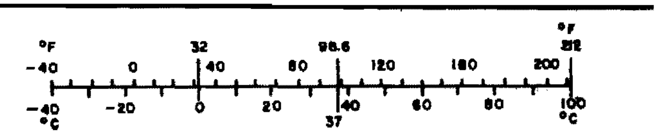
Symbol	When You Know	Multiply by	To Find	Symbol
	LENGTH			
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
	AREA			
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
	MASS (weight)			
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
	VOLUME			
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
	TEMPERATURE (exact)			
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
	LENGTH			
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
	AREA			
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
	MASS (weight)			
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
	VOLUME			
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
	TEMPERATURE (exact)			
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



The contents of this report reflect the views of the authors, who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

There was no invention or discovery conceived or first actually reduced in the course of or under this contract, including any art, method, process, machine, manufacture, design or composition of matter, or any new and useful improvement thereof, or any variety of plant which is or may be patentable under the patent laws of the United States of America or any foreign country.

PREFACE

This project report presents final results from Project 8-18-87-1141. The Project was initiated to determine the feasibility of using lasers for automating pavement crack measurements. and if found feasible, to develop a system which could be used for crack measurements for the Texas pavement management system or PES. An earlier report, "The Use of Lasers For Pavement Crack Detection", 1141-1, provided the details on the first phases of the project which investigated the feasibility of the use of lasers for this purpose. This report provides details on the third stage which describes a real-time system for crack detection and reporting that has been developed.

Special recognition is due Mr. Robert Harris of D-18, for his support in initiating the project and his many contributions to this research efforts.

Roger S. Walker
Chuying Kuo

January 1990

ABSTRACT

This report provides the final details on Research Study 8-18-88-1141. The research was initiated to investigate the capability of using lasers for crack detection in pavements. If such a capability could be developed it would be used to aid in obtaining and evaluating pavement distress and cracking information for the State's pavement management procedures, or PES.

The research effort has involved three stages. The first two stages were to determine the crack detection capabilities of the laser probes, used on the Surface Dynamics Profilometer (SDP). The SDP is owned by the State and used for road profile measurements. After experiments indicated that these probes could be used for such detection, a system was developed to further study this capability and to determine how it could be used to implement an automated high speed crack identification system. The results of the first two stages, and a complete description of the research effort is provided in Research Report 1141-1, "The Use of Lasers For Pavement Crack Detection". The third stage has been the development of a system which could be used for implementation of such a system so it's usefulness for P.E.S. data collection activities can be determined. This research report describes the final research effort.

KEY WORDS: Surface Dynamics Profilometer (SDP), Lasers, Pavement Distress Measurements, Pavement Crack Identification and Recording.

SUMMARY

This report provides final details on The Texas State Department of Highways and Public Transportation (SDHPT) Project 8-10-88-1141, Crack Detection Using Lasers. The project was initiated to first determine the feasibility of using the laser probes on the Surface Dynamics Profilometer (SDP) owned by the SDHPT, for crack detection and identification. As reported in Research Report 1141-1, 'The Use of Lasers For Pavement Crack Detection', it was determined that the laser could be used to provide a limited crack measurements capability, the principal limitations being in the number of lasers needed for extensive crack measurements and the necessary computing required for the crack detection and reporting algorithms. The SDP was selected for the initial testing and evaluation as it had existing on-board laser equipment.

In the initial study, two detection algorithms were found to perform well for crack detection, however, they could not be used in real-time with the computing hardware for the initial investigation. During this last phase of the project and reported herein, additional hardware was obtained and software developed which could be used for implementing a real-time crack identification and reporting system. The crack measurement hardware includes the Selcom Laser probes, the Motorola opened ended VME architecture, and the COMPAQ portable PC.. The software implements two crack detection algorithms, and crack reporting procedures. The system is capable of real-time measurements and reporting at normal highway driving speeds.

The system is currently being installed in a van for field implementation. The usefulness of the system for providing crack measurements for the State's pavement management system or PES, can only be determined from extensive field usage.

IMPLEMENTATION STATEMENT

An automated and objective procedure for crack measurements and recording would provide a significant savings to the State during P.E.S. data collection procedures. It could be used in many other areas where statistical information regarding pavement cracking is desired.

TABLE OF CONTENTS

APPENDIX

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Project and Report Scope

This project was initiated to determine the feasibility of using the laser probes on the Surface Dynamics Profilometer (SDP) owned by the State Department of Highways and Public Transportation (SDHPT), for crack detection and identification. If found feasible a system was then to be developed for use on the ARAN measurement vehicle, also owned by the State. The SDP was selected for the initial testing and evaluation as it had existing on-board laser equipment.

Initial evaluations proved that the lasers on the SDP could be used for crack detection. Additionally, successful uses of similar lasers for this purpose had been reported (Ref 13). Based on these results, the study proceeded in obtaining the necessary parallel processing equipment and developing additional real-time software so that an automated crack measuring system could be implemented for the State's pavement evaluation system, PES.

Research Report 1141-1, "The Use Of Lasers For Pavement Crack Detection", (Ref 8), provides details on the first phases of the research effort. For these initial phases, two lasers, one in each wheel path, were used to obtain crack data which was processed on a Motorola 68000 based data acquisition board and the COMPAQ Portable III. The data was sampled at 16 times per inch and filtered to remove the DC component and long wavelengths before processing. The data was analyzed using several different statistical techniques which are discussed in Ref 8. Two analysis techniques were found to provide good results. These were the running mean/slope threshold and the autocorrelation difference methods. However, it was determined that these algorithms could not provide crack detection in real-time with the hardware developed in this initial study.

The study has preceded to the last phase, where improvements have been made in the crack detection and reporting algorithms, and the hardware so that real-time processing at speeds up to 60 MPH is possible. In particular, the autocorrelation difference method was improved for faster processing. A new algorithm, using a running mean and slope analysis (running-mean downup method) has also been developed.

The Motorola open architecture VMEsystem has been used to replace the MC68000 based wire wrap board. This system interfaces with the other system components, the COMPAQ computer and Selcom lasers. The VMEsystem architecture allows the user to configure a system for specific applications.

The necessary background and other system requirements for the project are provided in the first project report (Ref 8). The report, herein, describes the improved and new detection algorithms, Chapter 2, and information on the real-time parallel processing hardware and software used, Chapter three. The Appendix provides details on the interface wiring between the Motorola VME system, general operating instructions and the algorithm software. The operator interface software was written in the C language. The real-time software in the Motorola system was written in MC68020 Assembler.

It was originally planned to have a working system for implementation during the middle of the last year of this project. However, delays in obtaining the necessary hardware and in mounting the laser equipment on the ARAN prevented this from occurring. Currently, the necessary mounting containers are nearing completion. Also, there is a possibility that the system will be mounted in the upgraded SDP van, rather than the ARAN, using the lasers for both profile and crack measurements.

It is still questionable on how well the two algorithms implemented will be able to provide suitable crack detection for PES. As will be shown in Chapter 2, they have been found to provide good crack detection and reporting on the small sample of pavements selected. The running mean downup method appears to work best. A much larger sample, however is needed for testing the algorithms. Using the original methods of recording the raw data for later processing has not been practical because of the large amounts of data that must be collected (16 samples per inch). Thus, the capabilities of these two algorithms and their appropriate parameters can only be determined during wide scale usage of the system.

CHAPTER II

CRACK DETECTION PROCEDURES

The various methods initially investigated to identify pavement cracking are described in Research Report 1141-1, (ref 8). The two methods which consistently gave the best results were the running mean/slope threshold technique and the autocorrelation difference method (Codiff). However, the computing requirements to implement these algorithms could not be achieved in real-time. The methods were applied on the laser data after it had been collected and stored on disk.

During this last phase of the project, a slight change has been made in the autocorrelation difference or Codiff filtering method. A new algorithm, referred to as the running-mean downup method (Downup), has been developed which provides detection improvements over the running mean/slope threshold technique and permits real-time computation. These methods are discussed in this chapter. The procedures and software used for crack reporting are discussed in Chapter 3.

2.1 Autocorrelation Difference Method (Codiff)

The autocorrelation difference method is discussed in detail in Research Report 1141-1 (ref 8), thus only a brief description will be provided.

Autocorrelation is a statistic which measures the correlation of data at different time increments apart. Assuming ergodicity, the autocorrelation lag m, denoted r(m), tells if data points m time increments apart over a length of data are related. The autocorrelation value will be approximately zero if the data is uncorrelated. Data with sharp cracks will show large correlation for a lag or two but the autocorrelation value decreases rapidly as the number of lags increases. Data with longer wavelength components, such as bumps, show high autocorrelation values for longer lag times.

The autocorrelation difference method is an enhancement of using the variance of the pavement crack data, (Variance method in Ref 8). It involves determining the spread between r(0) and r(m) calculated for every one inch (16 points) block of data. This difference is then compared with a threshold value. As discussed previously, r(0), an estimate of the variance for zero mean data, is large for data with cracking. r(m) is the autocorrelation for data points in the 16 point block which are m time lags apart. r(m), m is typically 4, will decrease more rapidly if variance in the data is higher frequency, that is, sharp cracks.

Using the property r(0) $\geq$ r(m) and examining the four cases for relative values of r(0) and r(m) provides justification for this technique.

CASE I: r(0) small and r(m) small implies a small difference and no cracking.

CASE II: r(0) small and r(m) large is not possible by property r(0) $\geq$ r(m).

CASE III: r(0) large and r(m) small implies a large difference and cracking present.

CASE IV: r(0) large and r(m) large implies a small difference and no cracking.

Each of these four cases are illustrated on both filtered and unfiltered data in Reference 8.

The data can first be filtered with a highpass filter to remove the DC component and much of the variability caused by hills, tire bounce, and vehicle suspension effects. Initially, it was thought that this filtering needed to be performed. A modification to the filtering procedure was done to help speed up the computation. However, upon running the algorithm on different pavement types, it was observed that similar result were obtained without filtering. Since one case might work better than the other, the current system has been designed to allow either (operator selectable).

There are at least three shortcomings of the Codiff algorithm. Because a crack is indicated by the relative difference between two lag values, the method is unable to estimate crack width or depth. A third disadvantage or short coming of this method is that if in the middle of a crack, should one 16 point block end and another begin, the crack may not be detected.

2.2 Running-Mean Downup Method (Downup)

The Downup method provides an improvement over both the autocorrelation difference method and the running mean/slope threshold technique. Unlike the Codiff method, it can estimate both crack width and depth. It also provides the ability to adjust for elevation changes which was often a problem in the running mean/slope threshold technique. Most importantly, it can be executed faster than either the Codiff or the running mean/slope threshold technique and is thus more suited for real-time.

The basic idea behind the Running-Mean Downup Method is that a crack is usually identified by a sharp negative (down) slope followed by a positive (up) slope. A running mean is used to filter the noise of the laser data and also help establish a reference plane. Figures 2.1 and 2.2 illustrates this method on sample laser data. The sample data, the running mean of the data, and the slope of the running mean of the data are illustrated in the Figure 2.1. A four point running average is used. The number of points used for the running average is denoted as the 'mbar' parameter.

The slope of the data is computed by taking the difference between each averaged point for a base length ('sbar'), or for this case, seven. The slope or difference is illustrated by the third line in Figure 2.1. This difference is denoted by the variable 'diff'. Note that diff indicates a vertical depth (the spacing between adjacent points are equal and thus a division is not performed in computing the slope in order to save processing time).

The slope of the crack may consist of several points. Since the pavement surface might be changing because of an elevation change (sharp drop, pot hole, etc.), there needs to be a maximum value which is used to determine if a change is because of a crack or because of some other characteristic or elevation change. The parameter 'slope' is used for this purpose. This parameter is used by the algorithm when examining two elevation changes in the same direction. If both are greater than slope, an elevation change is assumed and not a crack. If a change is less than the parameter slope, or the first change is greater but the second is not, a crack beginning is assumed. This allows the detection of the crack about a reference surface.

A typical crack requires both a downward slope followed by an upward slope. Thus, the algorithm searches for an acceptable downward slope which is followed by an upward slope. If a second downward slope occurs, then the search restarts. Other wise the crack depth continues. This allows for slight variations in the crack characteristics. Two other parameters are used. The parameter 'width' is used to provide a maximum acceptable crack width, and 'tc' for the maximum acceptable crack depth. The Downup algo-

Fig 2.1

Running Mean and Slope - Test Data

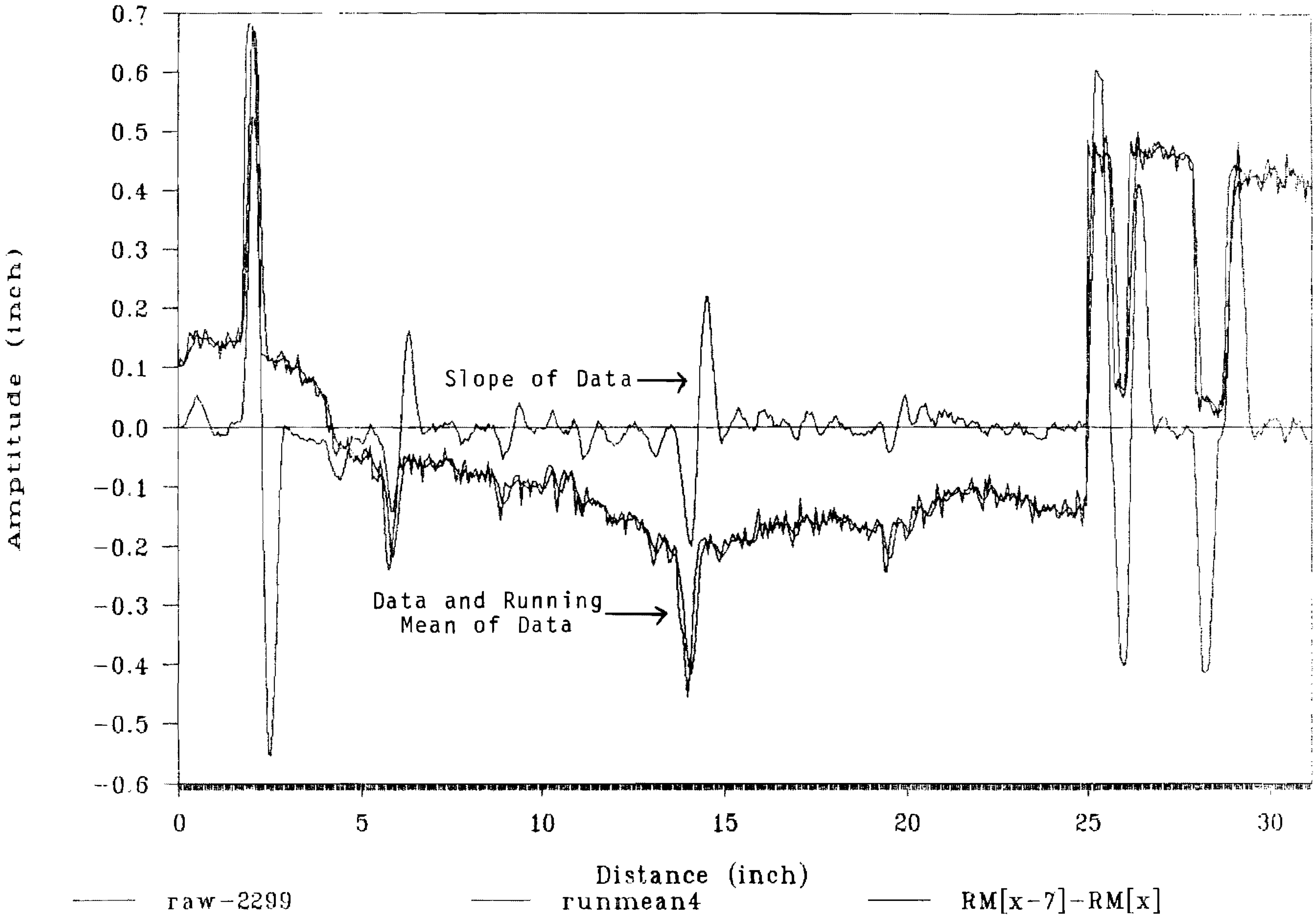
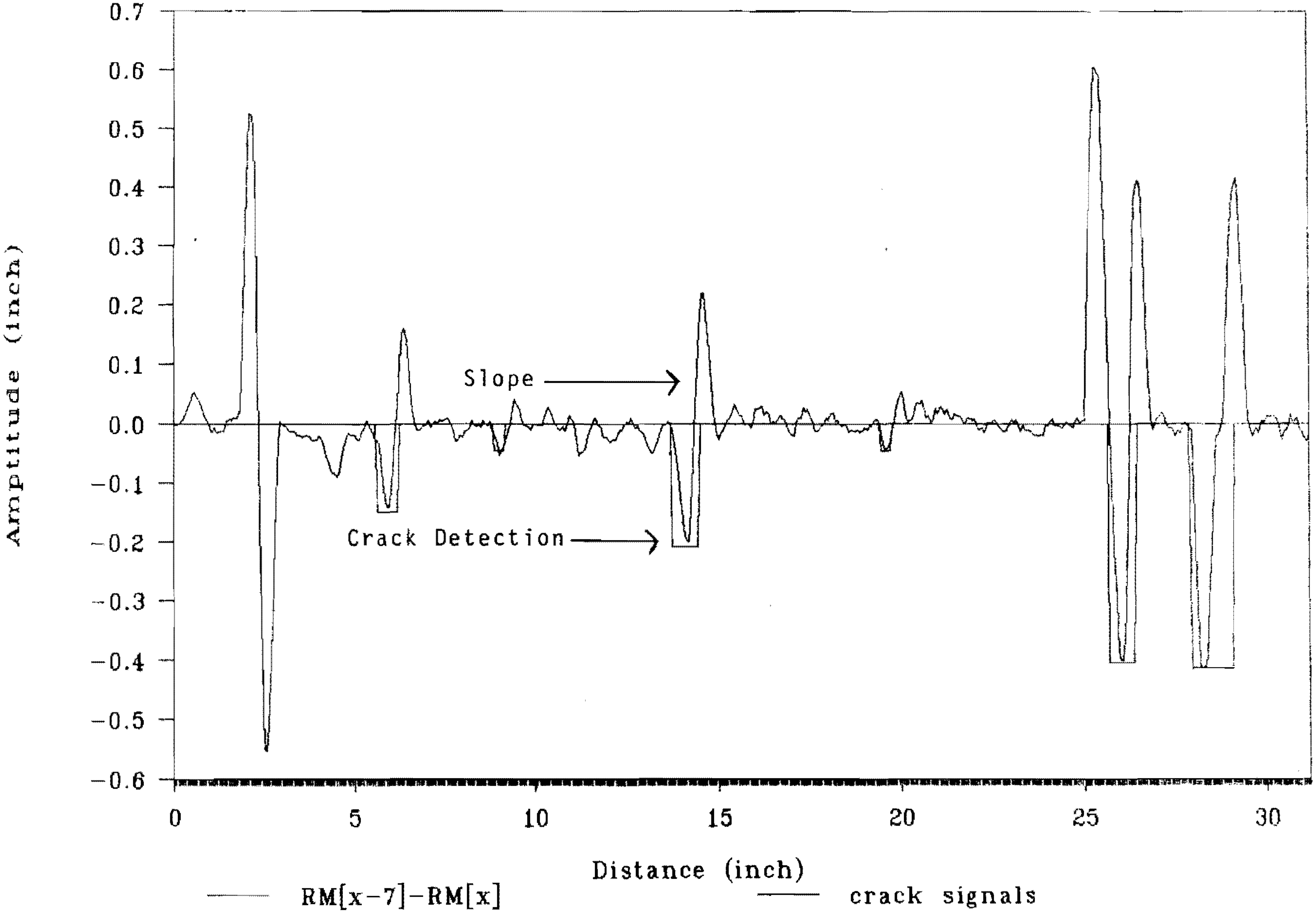


Fig 2.2

Downup Crack Detection - Test Data



rithm, of course can't detect all crack characteristics, but has been found to perform well on the pavements sampled. The best suited set of algorithm parameters may be changed by the operator as experience is gained in the use of the method during various field measurements.

The following code illustrates the algorithm.

```
{       mi=0;
        si = 0;
        diff = 0;
        depth =0;       }
/*   Compute the running mean and slope */
{
      value[(mi+mbar)]=data;
      rtotal = rtotal+data-value[mi];
      rmean = rtotal/mbar;
      mean[(si+sbar)] = rmean;
      diff = rmean-mean[si];
      mi++;                 }
/* Find a possible crack  */
{
     if (depth == 0)
         if (diff > slope || diff < -slope)
             {
                 depth = diff;
                 start = pos-1;
             }
         else;  /* normal profile frequency value, ignored*/
     else
     if (depth > 0)
          if (diff < slope)
              {
                  end = pos-sbar;
                  if (negative[1] !=0)
                         {
      /*                   if (negative[1]< -depth)
                                depth = negative[1];
                           else
                                depth = -depth;
      */                   /* write start,(depth),end */
                           depth = (-negative[1] + depth )/2;
                           fprintf(repfile, "%ld %ld %ld\n ",
                                negative[0], depth , end-negative[0]
                         }
                      else;    /* unmatched positive ignore*/
                  negative[1] = 0; /* delete negative record */
                  depth = 0;
              }
          else  /* diff still > slope */
              if (depth < diff)
                  depth = diff;
```

```
              else ;  /* depth not increased */

     else   /* depth < 0 */
          if (diff > -slope)
              {
                   negative[0]= start;
                   negative[1] = depth;
                   depth = 0;
              }
          else  /* diff still < -slope */
              if (depth > diff)
                  depth = diff;
              else ;  /* depth not decreased */
}
```

Then, the crack may be accepted or rejected by the following statement.

```
{
           if (width < max && depth >= tc)
{
```

Figure 2.2 provides the a plot of the computed slope or vertical difference between successive points and the results of the Downup algorithm of the data in Figure 2.1. As can be noted, this algorithm provides both an estimate of the crack width and depth. Notice that the upward changes in the pavement characteristics, are not detected as a crack. The elevation change shown at about 25 inches and the corresponding cracks were detected. The Codiff and running mean/slope threshold methods have problems with this test case. The Codiff method would record each positive and negative change as a crack, and the running mean/slope threshold method would have a problem adjusting to the abrupt elevation change.

The parameters defined above adjust the detection method for the various pavement types. The values used in Figures 2.1 and 2.2 are:

tc=15 slope=11, mbar=4, sbar=7, and width=32.

These numbers represent consecutive points in the vertical direction (10.8/4096 or 0.002634 inches) for the first two parameters, and horizontal direction (1/16 or 0.0625 inches) for the other four parameters. This particular set of values have been found to perform well for the pavement samples considered.

Figures 2.3 to 2.8 provide example results for asphalt pavements with severe, mild, and light cracking. The odd number figures, (2.3, 2.5, 2.7, 2.9) depict the result of

Fig 2.3

Severe Cracking - Codiff Method

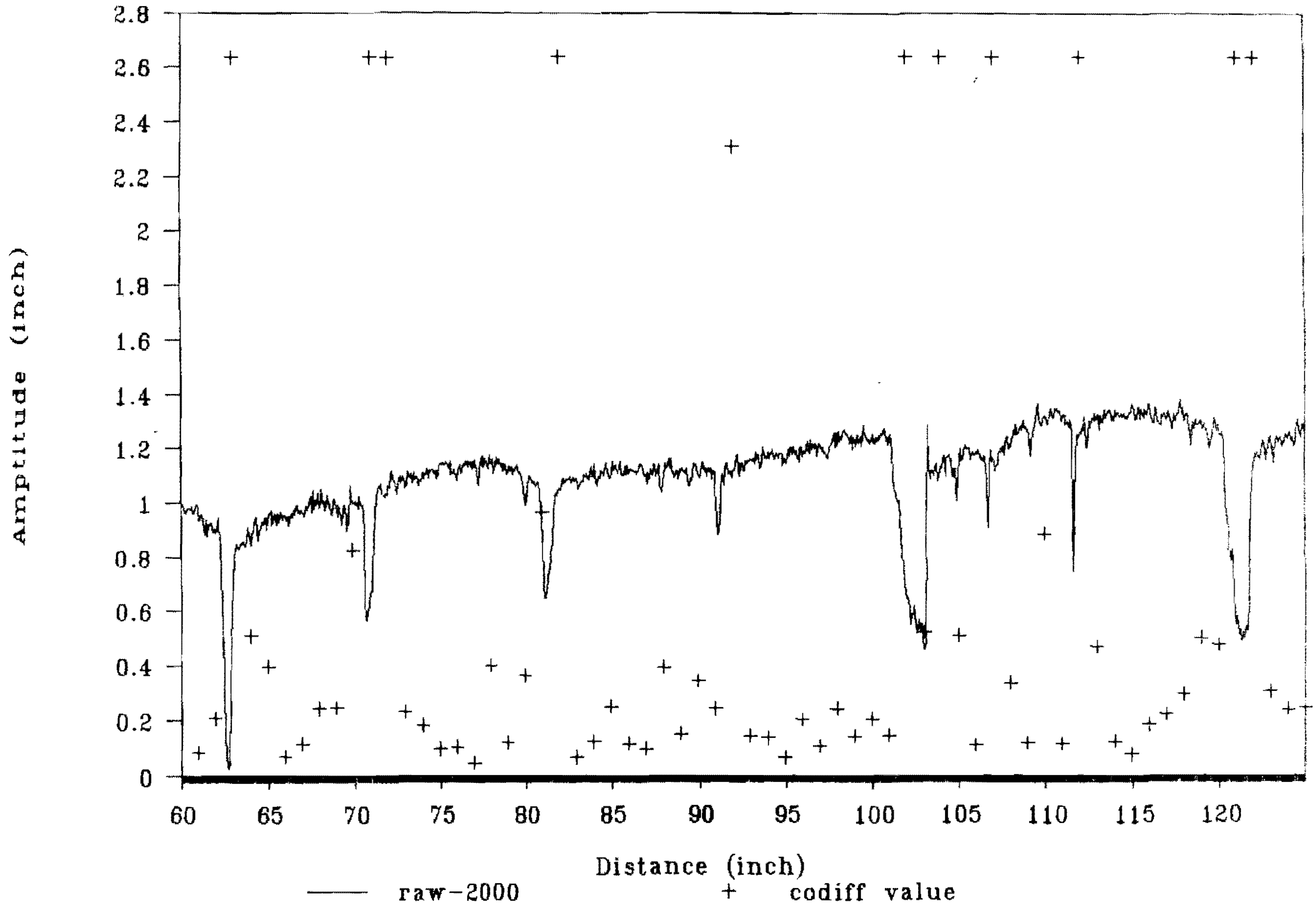


Fig 2.4

Severe Cracking - Downup Method

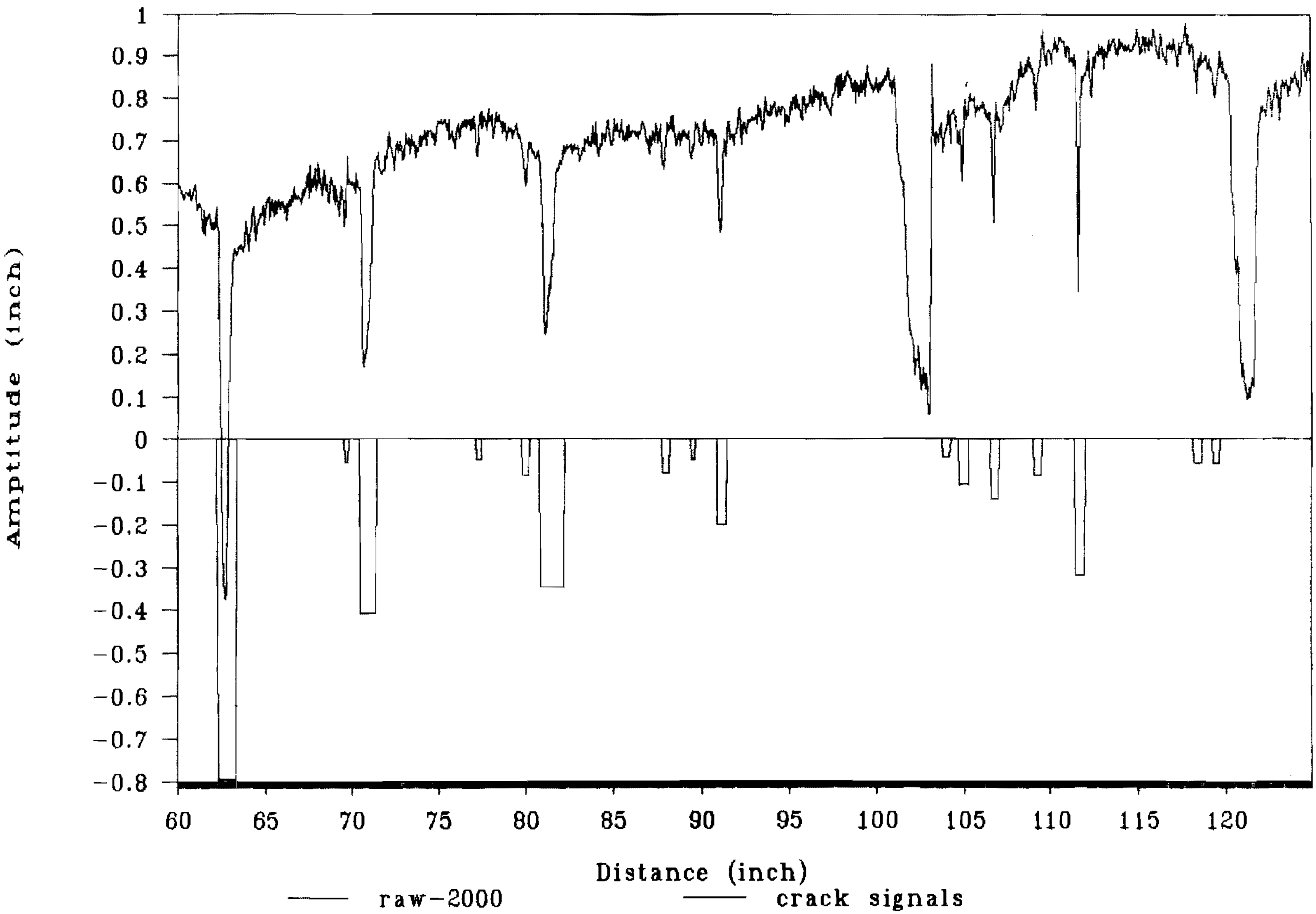


Fig 2.5

Mild Cracking - Codiff Method

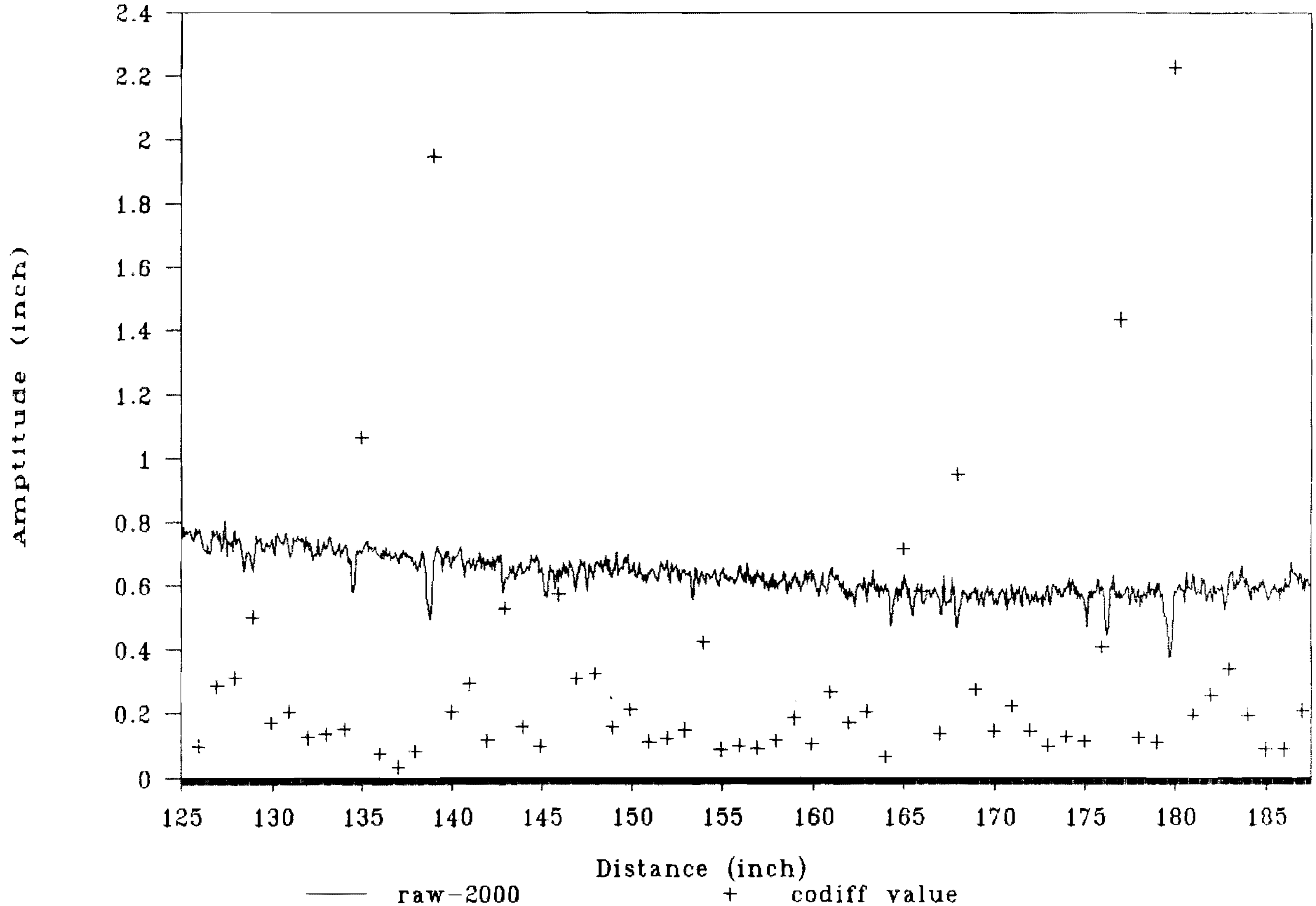


Fig 2.6

Mild Cracking - Downup Method

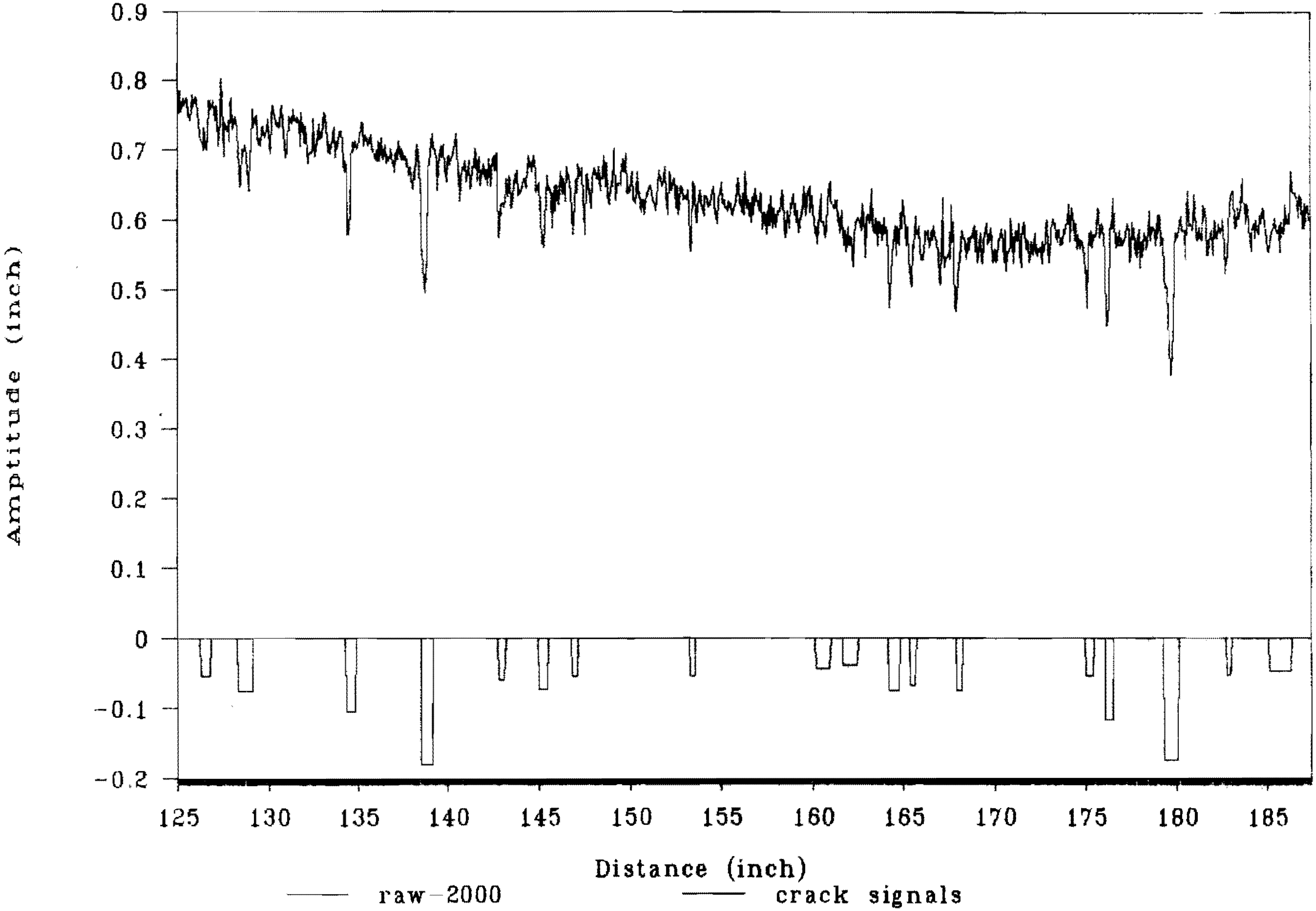


Fig 2.7

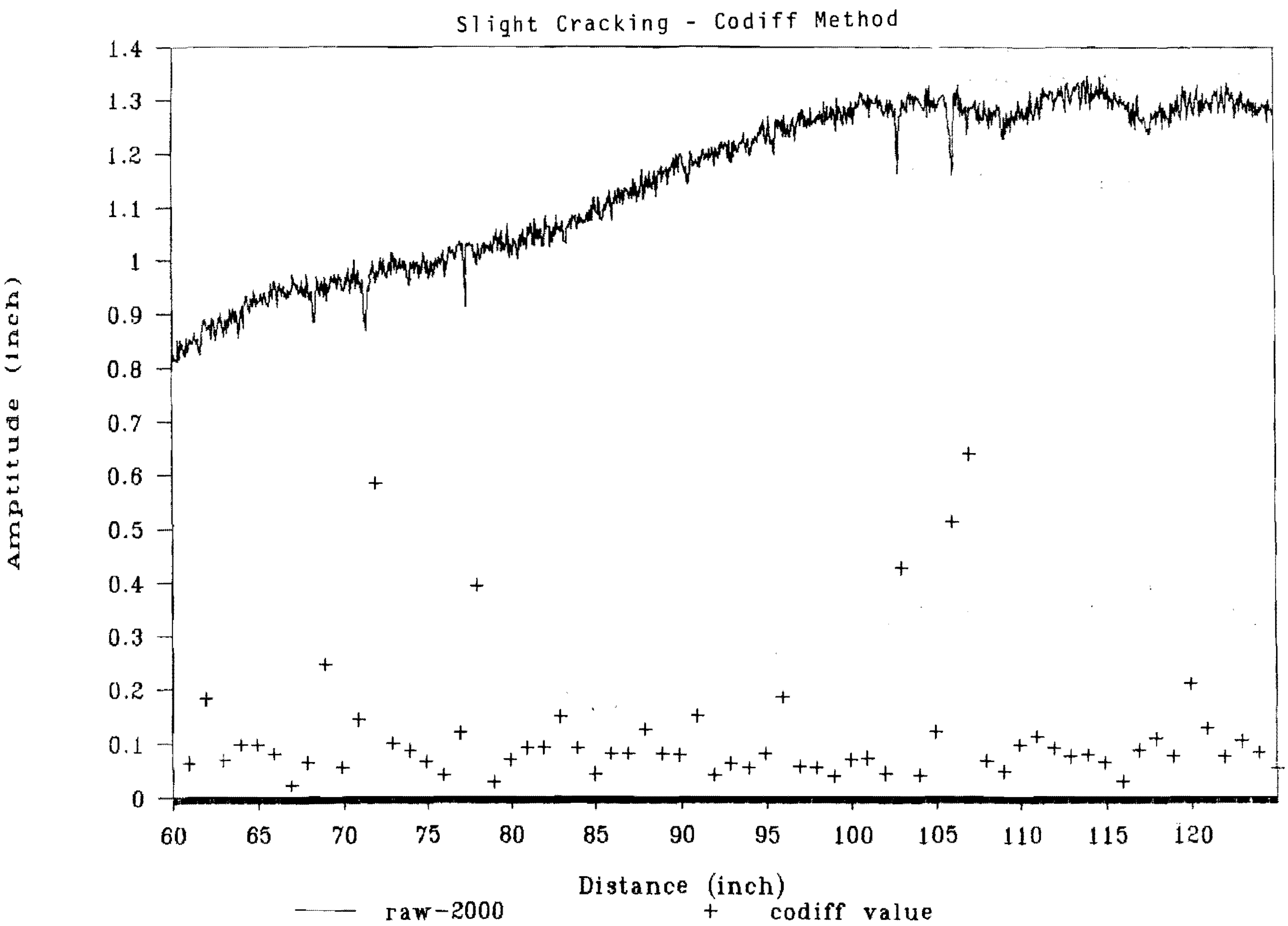
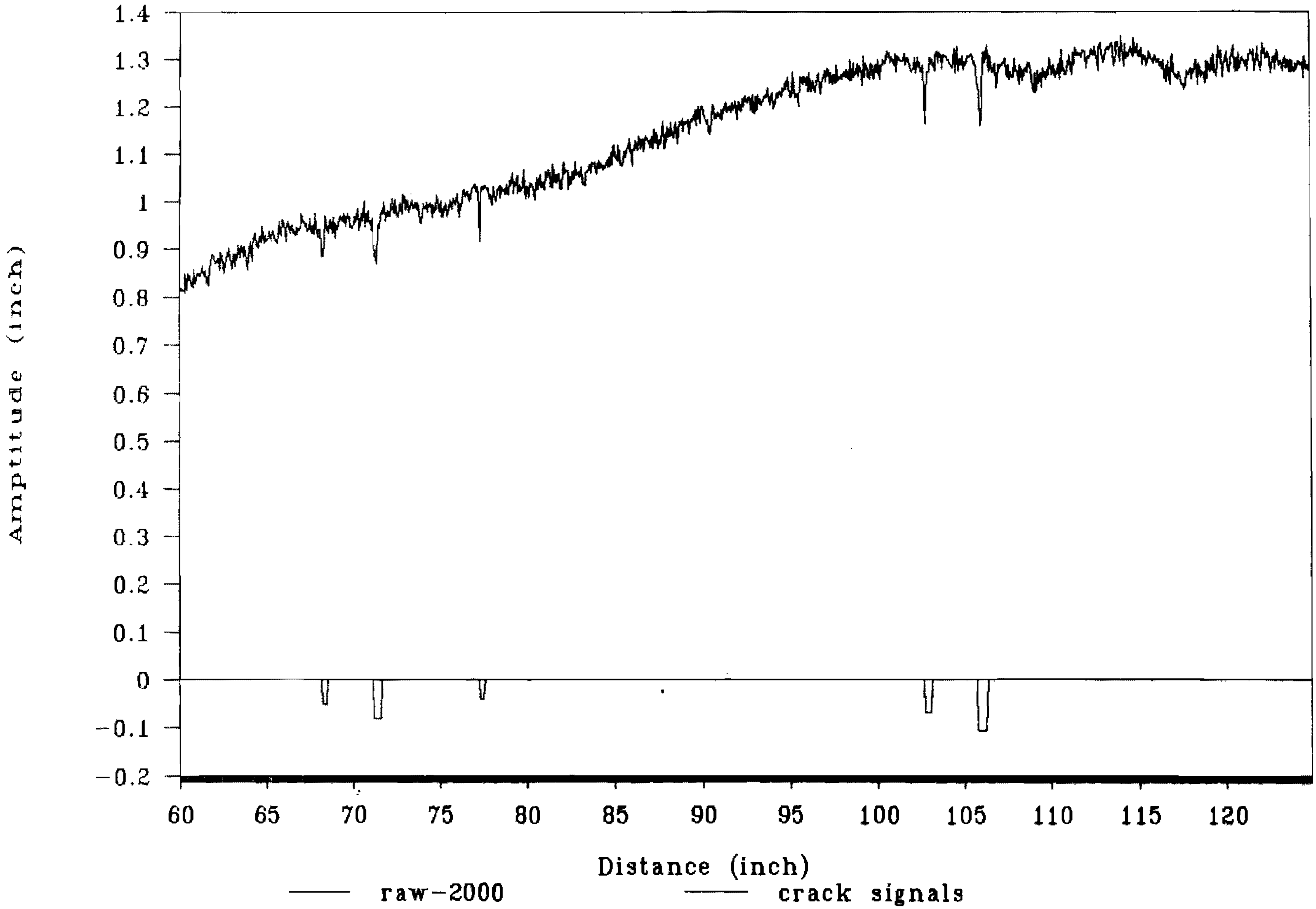

Slight Cracking - Codiff Method
Amptitude (inch)
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
60
65
70
75
80
85
90
95
100
105
110
115
120
Distance (inch)
raw-2000
codiff value

Fig 2.8

Slight Cracking - Downup Method



the Codiff method, while the even numbered figures (2.4, 2.6, 2.8, 2.10) depict the Downup method. These plots illustrate that both algorithms detect cracks. The Codiff method does so by the greater lag difference values as explained by Case III, discussed earlier. The Downup method indicates cracks by the width and depth associated with each crack, assuming it is within the prespecified intervals.

All figures indicate the relative displacements of the pavement surface profile so that the results of the two algorithms can be included on the same plots. That is a constant magnitude was subtracted from the pavement surface magnitudes. The magnitudes were than scaled to inches by multiplying each value by the ratio of the laser measurement range to the full scale resolution or 10.8 inches divided by 4096. The magnitudes of the difference in lag values for the Codiff method were not subtracted by any scale factor, thus to convert them back to their unscaled value, they would need to be multiplied by the inverse of the above relation or 379.259. The unscaled values are those used and displayed when in the real-time measurement mode, and which are used by the operator for selecting the appropriate threshold. Values exceeding this threshold are used to indicate a crack. For the Downup algorithm, (even number figures) the detected cracks widths and depths may be read directly from the plots.

The figures illustrate a major advantage of the Downup method over the Codiff method. Notice that the threshold values needed to detect cracks for the Codiff method is sensitive to the severity of cracking. That is a threshold around 0.3 * (4096/10.8) should be used for slight cracking (Figure 2.7). However, the use of this same value for severe cracking, (Figure 2.3), would incorrectly indicate to many cracks. On the other hand, For the Downup method, the same set of parameters provided good results for these three levels of cracking.

Figures 2.9 and 2.10 illustrate the use of the two methods on a different pavement texture. The pavement for this case had a seal coat which was a little over a year old. Mild 1/6" to 1/8" alligator cracks, however, had begun to reflect through the seal coat. The use of the Codiff algorithm is illustrated in Figure 2.9 and the Downup method in Figure 2.10. For this pavement an appropriate threshold would be difficult to select for the Codiff algorithm. The Downup method, however, properly estimated the correct amount of cracking which was verified by visual examination.

There, of course, are some pavement surfaces where it is doubtful any algorithm would work well. For instance, newly resurfaced pavements with a heavy seal coat, can

Fig 2.9

Seal - Coat With Alligator Cracking - Codiff Method

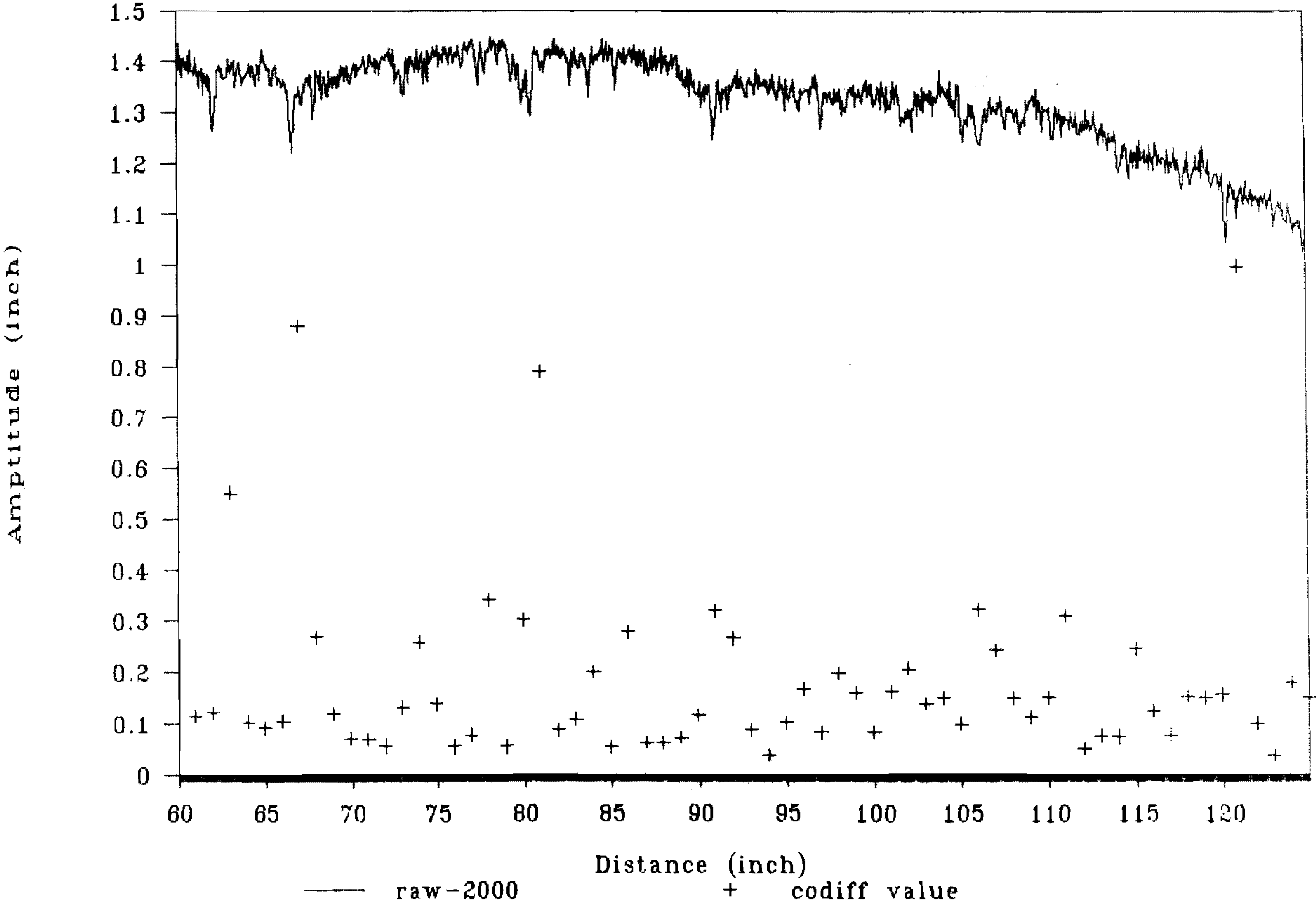
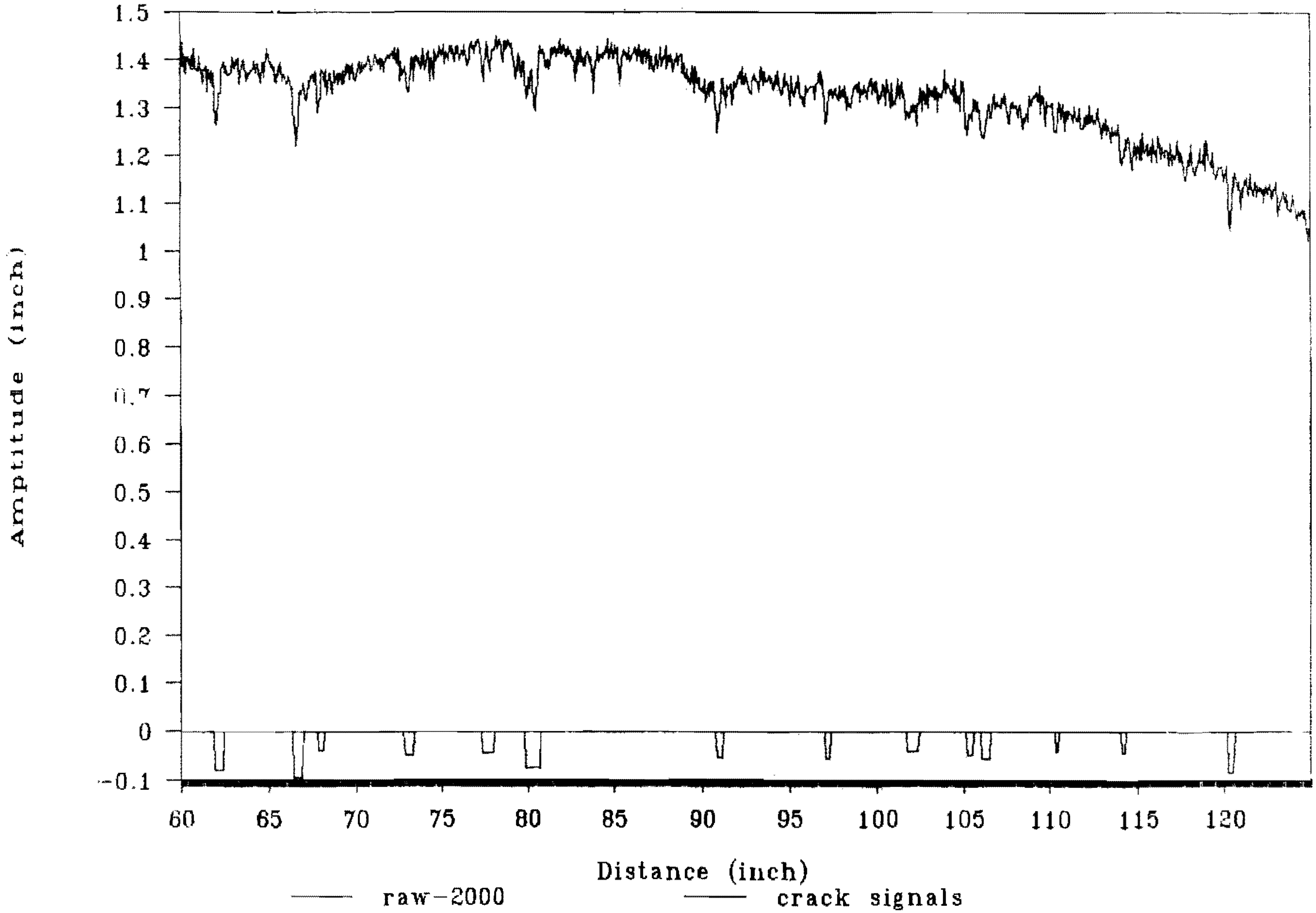


Fig 2.10

Seal - Coat With Alligator Cracking - Downup Method



result in surface textures with the same characteristics as smooth texture surfaces with cracks. An additional improvement to the Downup algorithm is currently being investigated which will help to minimized the false crack detection for such cases. Of course, such a pavements wouldn't be suspected of cracking to begin with, however, this stresses the need for a trained operator in the use of the crack measurement system during the measurement process. A better determination of the performance of the algorithms on various pavement types and proper parameter selection can be determined by extensive field use.

CHAPTER III

SYSTEM ARCHITECTURE AND MEASUREMENT PROCEDURES

This chapter describes the parallel processing architecture and instrumentation used for real-time crack detection and recording. This system updates the processing capability of the initial data acquisition system for real-time crack measurements by replacing the 68000 PC based data acquisition board with the Motorola open architecture VME-system. The VMEsystem with its various processing and I/O modules configured for this crack measurement application will be referred to as the Real-Time Crack Processing Unit (RTCPU). The RTCPU interfaces with the other system components, or the COMPAQ computer and Selcom lasers. The RTCPU is structured to support multiple lasers. Details of the laser and COMPAQ system components are discussed on the first project report, Ref 8. A brief description of these subsystem components is given next, followed by a description of the overall parallel processing system architecture of the RTCPU. The crack measurement system illustrating the system components is shown in Figure 3.1

3.1 The Selcom Optocator

The Selcom optocator is an opto-electronic measurement system which measures the distance to an object. The basic components of the optocator are the non-contact laser probes, the probe processing units (PPU), and the CPU sub-rack which contains the power supply and the receiver-averaging boards, and which receive and process data from the laser probes. An optocator interface module which plugs directly into the VME bus can also be used in place of this last component.

The laser probe contains a pulsed, modulated (32KHz) and intensity-controlled gallium-arsenide (GaAs)laser diode, a position sensitive photodetector and an appropriate lens system. The GaAs laser probe gives off pulsed, modulated invisible infrared light. The bursts occur at a frequency of 32 KHz which accounts for the 32 KHz data rate of the serial data passed to the receiver-averaging board. The light from the laser beam passes through a lens which focuses the light in the center of the measurement range. The spot size which strikes the ground surface is approximately 1/4 inch by 1/16 inch.

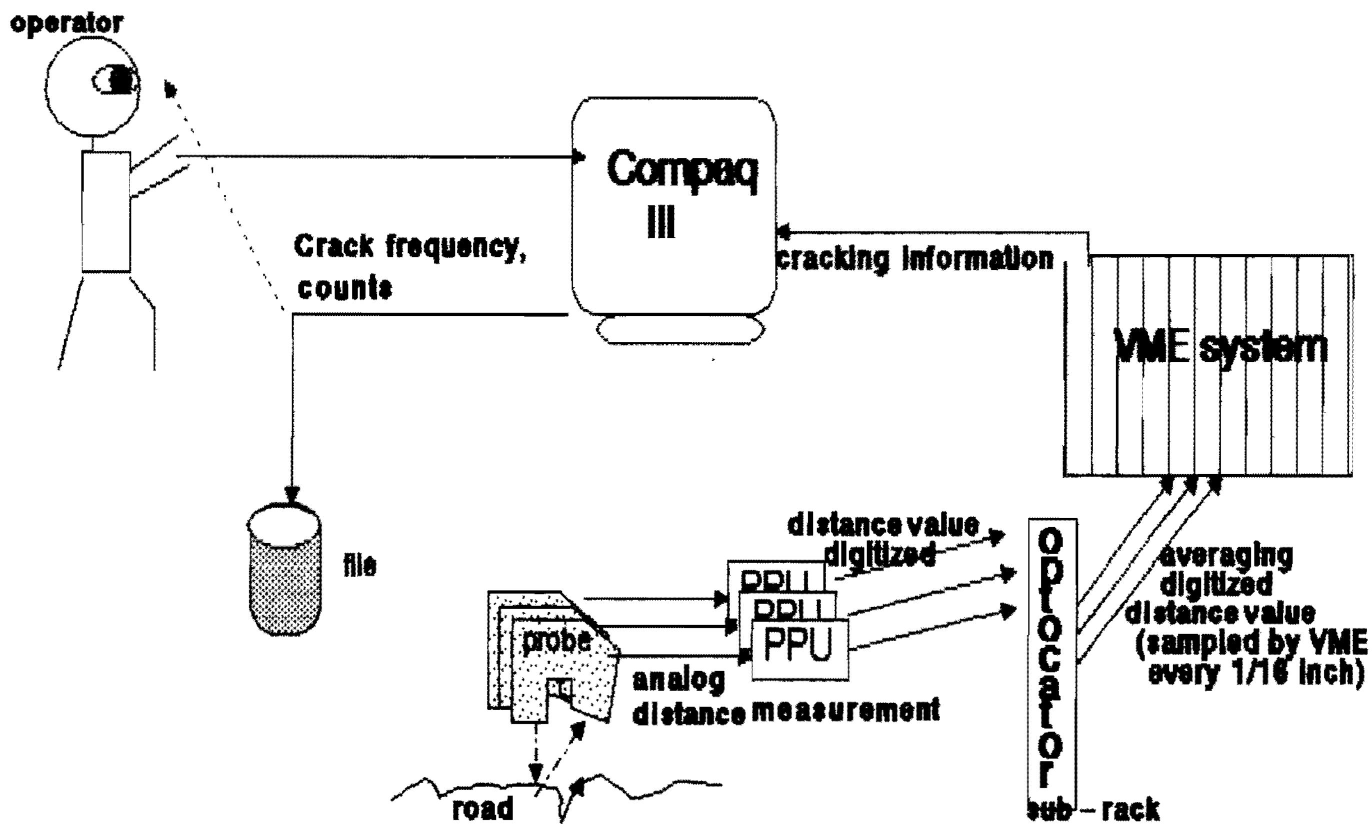


Fig 3.1

Crack Measurement System

The optocator measures the distance to an object by use of the triangulation principle, as illustrated in Figure 3.2 From a light source, L, a concentrated light beam is directed onto the surface of the measured object, O1. The light beam will strike the surface at point A and the scattered light reflection is focused through a lens to a point A' on a position sensitive detector. If the distance of the measured object is changed by X, the laser beam will hit point B on surface O2 and be focused at point B' on the detector. Since the relative position of the light source, the lens and the detector are fixed, the relation between X and X' is known and distance measurements can be obtained.

The maximum measurement range, O1-O2, as well as the standoff distance must be considered when mounting the laser probes. Selcom's gauge probe type 2008 requires a standoff distance of 355mm (13.98 inches) and has a measurement range of 256mm (10.08 inches) [5]. Therefore, to obtain correct measurements, the laser probes should be mounted such that the distance from the bottom of the probe to the ground surface (middle of the measurement range) is approximately 14 inches. When correctly mounted, distances plus or minus 128mm (5.04 inches) from the calibrated ground level can be accurately measured. Refer to Figure 3.3. Measured surfaces which do not fall within the measurement range will result in invalid readings.

The PPU processes the analog signal from the laser probe and sends the signal (in digital form) to the receiver-averaging boards located in the CPU sub-rack or optocator interface module. The serial digital output includes the 12 bit value from the analog to digital converter as well as 3 invalid data bits. The probe processing unit determines invalid data if the reflected laser beam is not correctly detected by the position sensitive detector in the probe.

The receiver-averaging board receives serial data from the PPU at a rate of 32 KHz and is capable of reducing the data rate by forming the average of a number of measurements. The data rate, also referred to as updating frequency, is set by jumpers on the board. The update frequency ranges from a maximum of 32 KHz (no averaging) down to 62.5 Hz in powers of two.

Output from the receiver-averaging boards is the measured distance value represented as 12 bit parallel data plus a data invalid bit and a data ready flag. This 12 bit parallel data value is input to the RTCPU.

3.2 The Real-Time Processing Unit (RTCPU)

As discussed, the data acquisition board initially used

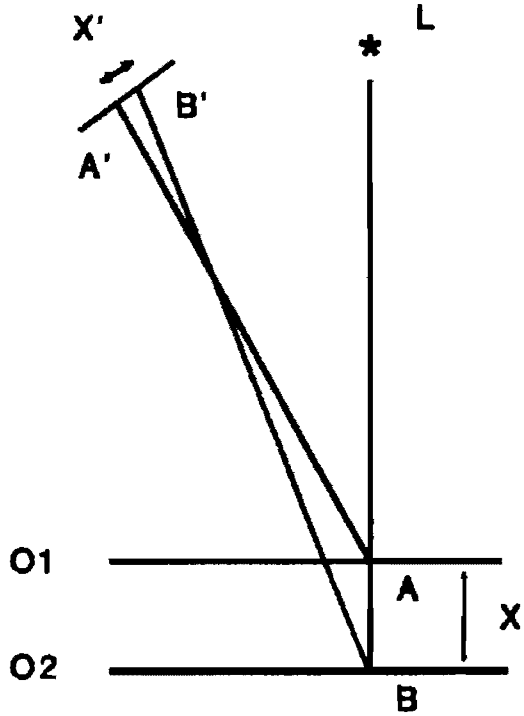


Figure 3.2 Triangulation Principle

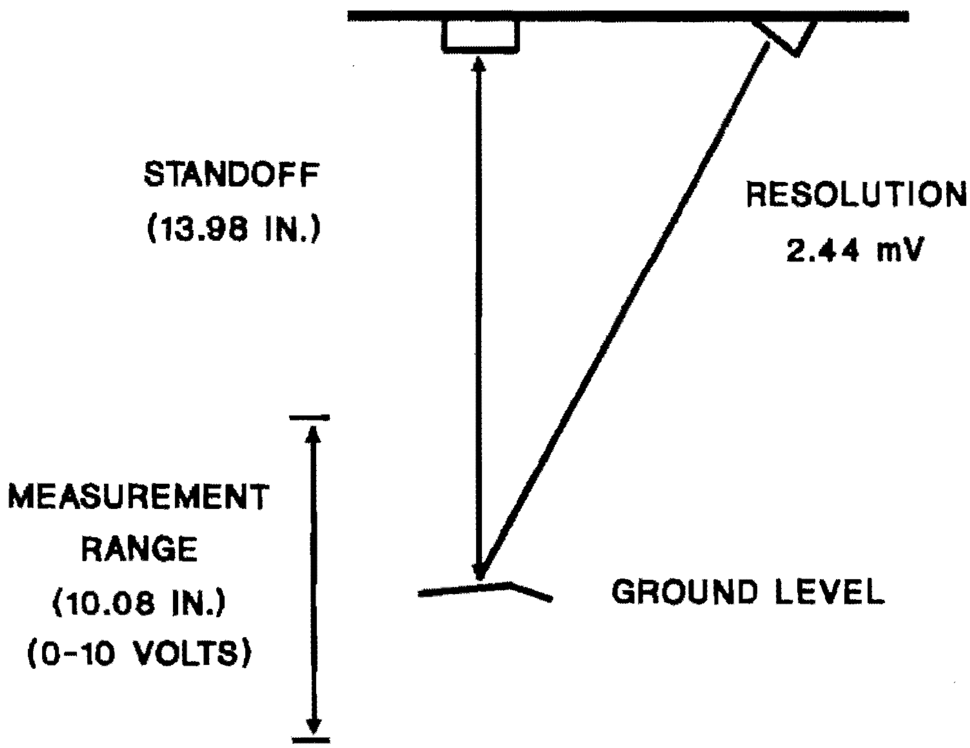


Figure 3.3 Laser Measurement Range

to determine the measuring characteristics and capabilities for the project is a specially designed board which uses the Motorola 68000 processor and plugs into one of the system expansion slots in the COMPAQ Portable III expansion module. Its function was to receive the laser data from the optocator and perform some preliminary processing of the crack data before passing it on to the COMPAQ Portable III for final crack identification and section analysis.

The RTCPU is used to replace this board. In addition to receiving the laser data from the optocator; however, it also performs the crack detection function by the real-time processing of the crack detection algorithms.

The VMEsystem architecture of the RTCPU allows the capability for designing various general purpose computing systems by purchasing a basic cardcage which has the VMEbus interconnect standard. The various desired systems can then be configured by purchasing one of the many individual VMEmodules (Ref 10) which simply plug into the VMEbus with the widely accepted eurocard connector. Typical VMEmodules are microprocessor boards, memory boards, various controller boards, and I/O boards. The VMEsystem architecture allows the user to configure a multiprocessor system with both local and shared memory.

For this system VMEmodules with the 68020 microprocessor (VME 133) interface to the PC via a XVME parallel I/O module (Ref 12) and a Data Translation DT2817 module (Ref 9). Each of these VME 133 boards are dedicated to processing the data from a single laser, and where one of the modules also has a master control function.

3.3 The COMPAQ PC Subsystem

The COMPAQ Portable III is the user's interface to the entire system. From the COMPAQ's keyboard the user can run the real-time crack detection and recording activities, perform various system diagnostics, or collect raw laser data. The programs which provide crack reporting and recording is run on the COMPAQ. The real-time crack count provided by the RTCPU provides a rough estimate of the number of cracks seen as the vehicle containing this equipment moves at highway speeds. Software in the COMPAQ summarizes, displays and records this information in the form of frequency counts for specified intervals.

3.4 System Data Flow:

The system data flow is illustrated in Figure 3.4. In this figure, it is noted that the operator interfaces with the COMPAQ computer initializing the real-time operations.

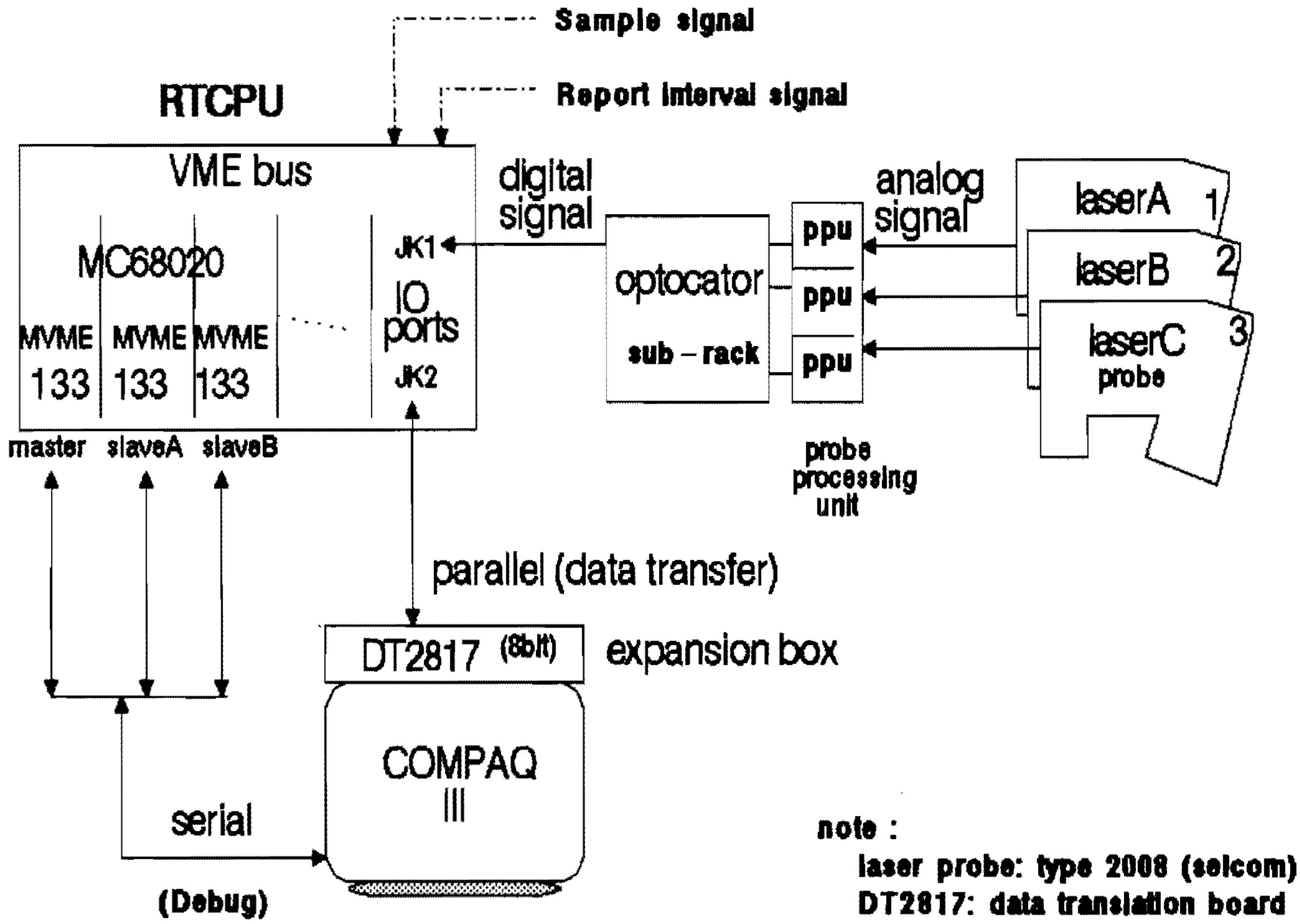


Fig 3.4

System Data Flow

The laser probes provide the detailed pavement profile information at 16 samples per inch which is passed to one of the crack processing boards (VME133) in the RTCPU. As indicated, the RTCPU is currently configured for three lasers, each one processed by a VME133, although with the modular approach used additional lasers can easily be supported. Each laser requires a separate processing board if both the Codiff and Downup algorithms are to used. (If only the Downup algorithm is used, the VME 133 board can support two lasers at normal highway speeds.) One of the processing boards is denoted as the master, which additionally performs the duties of determining when each of the other (slave) boards should sample the crack or profile data and also sends the processed data to the COMPAQ. The VME 133 boards interface to the Optocator and COMPAQ via a VME parallel I/O module. Communications between the VME 133 modules and the Parallel I/O port (XVME-240) is provided by the VME bus structure. The COMPAQ interfaces to the parallel I/O board via a PC based parallel port I/O module (DT 2816). The COMPAQ expansion box is used for containing this board.

Figure 3.5 provides the communication media for the parallel processing activities. That is the COMPAQ interfaces to the master VME 133 via ports 0 and 1 of the parallel I/O board. The two slaves communicate with each other and the master via flags in global shared memory, contained on each board. Each slave processor communicates with the optocator and laser probes via ports 4 to 7 on the parallel I/O board. This second figure only depicts interface to two lasers. This is because the current parallel I/O board being used (XVME-240) only provides for four (8 bit) ports. An additional board is needed for the third laser, which would be used by the master. The software is designed for this third laser.

3.5 System Software Modules:

The system software functions are illustrated in Figure 3.6. There are three basic processing components, the PC processing, master processing and slave processing. The PC processing software is performed by the COMPAQ and is written primarily in C. The Master and Slave processing software is done by the VME 133 boards and is written in MC68020 assembler. The PC based software is used to interface with the operator, and is thus the main control for the RTCPU.

The PC program provides the user with four modes. The first, the serial communications mode, allows the user to interface with the VME 133 debugging monitor(Bug 133 Ref 10), contained in the EPROM of each VME 133 boards. Thus various system checks can be made in the event of hardware

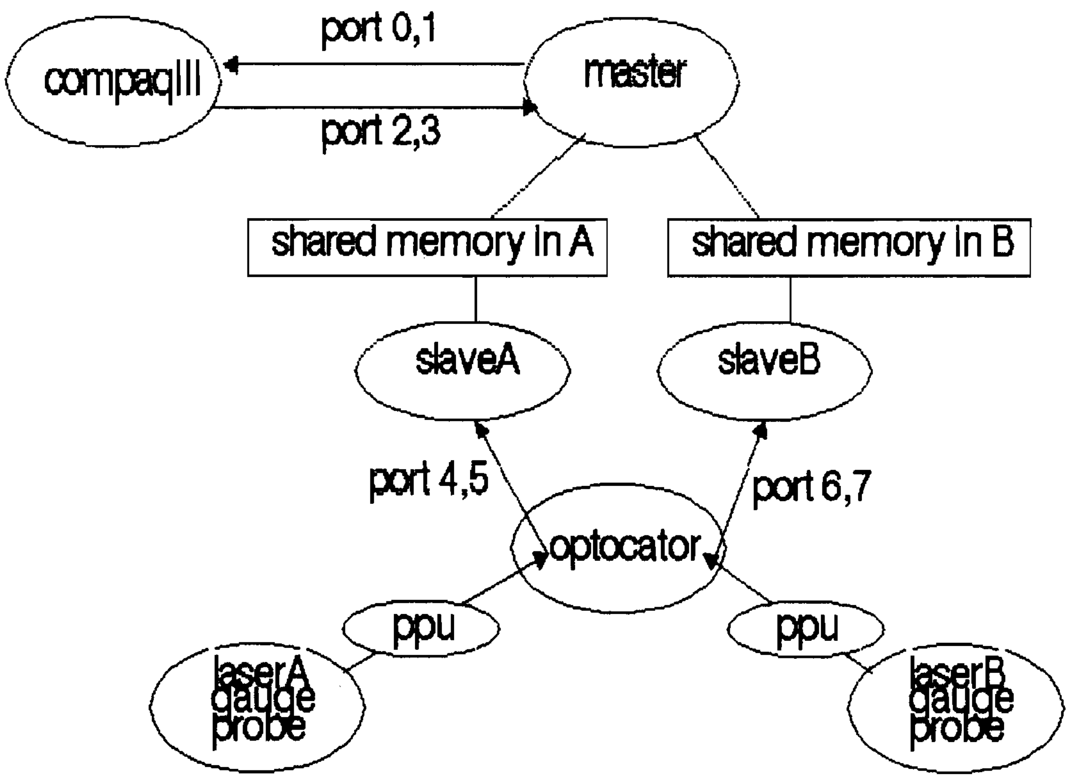
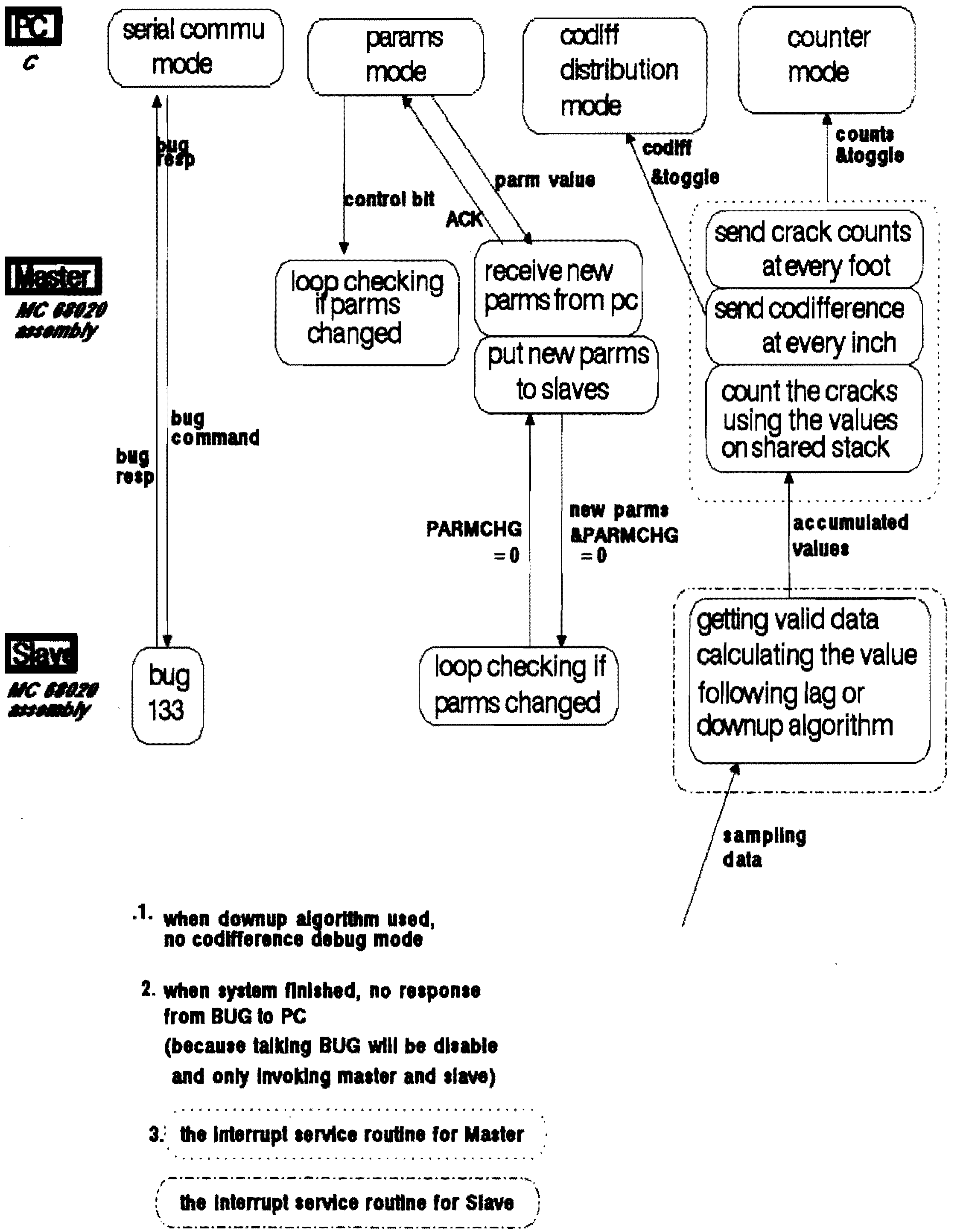


Fig 3.5

Parallel Processing Communications

Fig 3.6

System Software Functions



problems. The other three modes are used for normal operations. The parameter mode is used for entering the operating parameters, the Downup, Codiff mode for crack measurements and reporting, and the counter mode for investigating the characteristics of a particular pavement section for setting the measurement parameters.

For crack measurements, the system is first placed in the parameter mode where the various parameters, for the two algorithms (Codiff,or Downup) are entered. These parameters can be changed following a given run as required. Once crack measurements are initiated, the PC receives the crack counts from the VME 133 modules. The memory map used by the VME 133 boards for communications with one another is illustrated in Figure 3.7.

As discussed in Chapter 2, the two algorithms implemented by the slaves (master) indicate if a crack was detected in one inch. When the distribution mode is selected this crack count is reported to the master which (in addition to also performing the crack measurements for one of the lasers) than keeps track of the number of cracks reported in a foot. This count is sent to the PC. For each foot, the master also keeps information on the number of times two adjacent lasers both indicated a crack for the same inch. This also is sent to the PC and is displayed to the operator (and/or recorded). The master uses the 16 bits of the two ports 0 and 1 of the Parallel I/O Module for communicating this information to the PC.

When the PC is in the counter mode, the slaves provide the actual autocovariance difference magnitude for the Codiff algorithm and the crack depth measurement for the Downup algorithm. This information is reported to the master for every inch, which is then sent to the PC. The PC accumulates this information providing a frequency distribution indication of the number of times cracks occurred within interval of these variables. This information is useful for setting the crack threshold and depth parameters as discussed in Chapter 2. This information is provided to the PC, once again using ports 0 and 1 of the parallel I/O module.

Communications from the PC to the master is implemented by using ports 2 and 3 of the parallel I/O module. The various parameter and control functions are decoded into the 16 bits of these two ports. Ports 4 through 7 of the parallel I/O board are used for sending the laser measurement information from the optocator to the slave (master) VME 133 boards. Since the parallel I/O port currently used only has eight of these eight bit ports, another board needs to be added for three or more lasers.

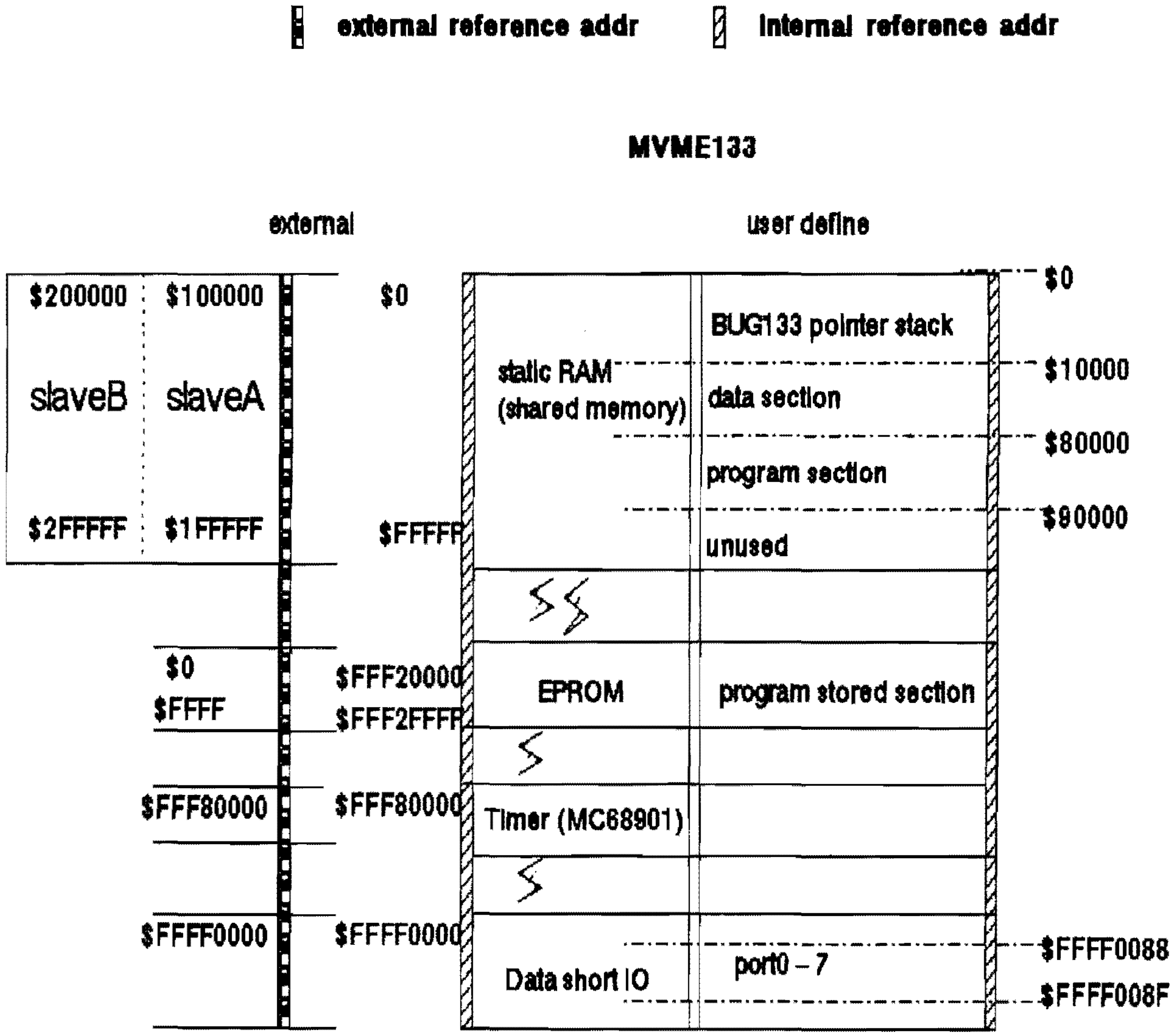


Fig 3.7

Memory Map

3.6 Crack Measurements and Recording.

As discussed, the laser crack measurement system is currently configured to implement the Codiff and Downup algorithms. Both primarily indicate if a crack was detected and its magnitude. The Codiff method only specifies the lag difference between autocovariance values indicating a possible crack, whereas the Downup method can provide an estimate of both the crack depth and width. Both of these methods are used to provide the cracking within an inch for the distribution mode. When in the magnitude frequency count mode the Codiff method still provides the lag difference magnitudes for any crack for a one inch resolution, whereas, the Downup method provides a summary of the crack depth. Although measurable using the Downup method, the two modes currently implemented do not provide the crack width.

During the distribution mode the PC keeps track of the individual and common laser crack counts as reported from the master. This information is then displayed on the CRT and recorded to disk. Figure 3.8 illustrates the screen used for displaying this information, as well as, the system parameter information. The PC keeps track and displays (records) for the operator, the number of times cracks were found within each of three count intervals for each foot for a specified distance interval. Thus the three intervals selected provide the number of times there were no cracks detected in a foot, the number of times one to three cracks were detected in a foot, and the number of times greater than three cracks were detected in a foot for a distance indicated by a distance reporting signal (can be manually selected). This information is displayed for each laser (up to three) and for the two adjacent lasers (lasers one and two and, two and three).

These statistics were selected so they could be used to help estimate the number of times alligator and block cracking was detected in the measurement interval. As noted, room is available on the screen to indicate the amount of alligator and block cracking, although, it is not currently implemented.

The screen shown in Figure 3.9 is used when operating in the crack count mode. As discussed, this mode allows the user to determine what the various crack detection thresholds or depths should be used for a given pavement type. The PC will accumulate the number of times the cracking magnitudes fall within 12 different user selected intervals. Both the distribution and the count information is also recorded on disk for later off-line evaluation. An example of the information recorded for the crack count mode is illustrated in Figure 3.10. A similar file is written for

Fig 3.8

Crack Reporting and Status Screen

Header : a
Active Lasers on = 0,1,
Lag = 4
Filter(smp/ft) = 192, (ft/cycle)=1
Speed = 30.0 mph Timer Constant = 118
Threshold for Crack is 1000

	LaserA	LaserB	LaserC	Comm A,B	Comm B,C
None					
1-3					
>3					
Block Alleq					
Counts Errors					

F1-Change Parameter F2-Quit System F5-Reset Counter

userlag.sc1

HEADER : a

A: Active Lasers on = A,B,
L: Lag = 4
F: Filter (smp/ft) = 192, (ft/cycle)=1
S: Speed = 30.0 mph
T: Threshold for Crack is 1000
H: Change the Header a

F2-Quit System F3-Count Mode F8-Debug Values F10-Serial Talk

userlag.sc2

Fig 3.9

Crack Count Distribution Screen

Enter the detected Laser number (0-2): 0

Enter the scale(0.5,1,2) : 1

Frequency for Recent inches

Class	100	200	300	400	500	600	700	800	900	1000	1100	--
Freq												

value=

F1-Rtn Change Mode F2-Exit System F4-Start Acc

userlag.sc3

```
 Header = u31
 Active Lasers bits = 3
 Lag = 4
 Filter(smp/ft) = 192, (ft/cycle)=1
 Car Speed = 30
 Crack Threshold = 1000
cnt =0 err=0,  cnt =0 err=0,  cnt =0 err=0,
=645,  =645,  =645,  =645,  =645,
=0,  =0,  =0,  =0,  =0,
=0,  =0,  =0,  =0,  =0,

for 2749 inches1898,664,107,49,18,4,5,1,2,0,0,1,0,0,0,



 Header = u32
for 3062 inches2312,554,112,41,18,7,3,5,4,1,0,5,0,0,0,
```

userlag.out

Fig 3.10

Crack Count Distribution File

the normal crack measurement and recording modes.

CHAPTER IV

CONCLUSIONS AND FURTHER RESEARCH

This report provides final details on The Texas State Department of Highways and Public Transportation (SDHPT) Project 8-10-88-1141, Crack Detection Using Lasers. The project was initiated to first determine the feasibility of using the laser probes on the Surface Dynamics Profilometer (SDP) owned by the SDHPT, for crack detection and identification. As reported in Research Report 1141-1, "The Use of Lasers For Pavement Crack Detection" (Ref 8), it was determined that the laser could be used to provide a limited crack measurements capability, the principal limitations being in the number of lasers needed for extensive crack measurements and the necessary computing required for the crack detection and reporting algorithms. The Surface Dynamics Profilometer was selected for the initial testing and evaluation as it had existing on-board laser equipment.

In the initial study, two detection algorithms were found to perform well for crack detection, however, they could not be used in real-time with the computing hardware for the initial investigation. During this last phase of the project and reported herein, additional hardware was obtained and software developed which could be used for implementing a real-time crack identification and reporting system. The crack measurement hardware includes the Selcom Laser probes, the Motorola opened ended VME architecture, and the COMPAQ portable PC.. The software implements two crack detection algorithms, and crack reporting procedures. The system is capable of real-time measurements and reporting at highway driving speeds up to 60 MPH.

As was discussed in the first report on this project, several important conclusions can be made as a result of this study. First, alligator and block pavement cracking can be detected using the Selcom lasers mounted in the wheel paths. Transverse cracking is more difficult to measure. The three laser configuration with the common cracking measurements might be able to provide an indication of such cracking. This configuration provides a way of measuring cracking across the lane. Multiple lasers also allows rutting to be detected.

Any user of this system must understand the limitations imposed by trying to detect cracking using only two or three narrow beams of laser light. Obviously, massive amounts of information across the lane is not available. Video systems provide much more detail, but this extra detail presents problems in processing the unwanted information. It is still believed that a system with a small cluster of lasers along and in between each wheel path could provide the best choice, although, such a system is not practical at the time, A less expensive laser or other similar sensor system would be needed. The combined use of video and the laser system described herein might provide a useful system. This has been one of the original intentions in the project by placing the laser system in the ARAN. The laser system could invoke the video system when cracking is detected. The video could then be later processed for more detailed analysis.

The system is currently being installed in a van for field implementation. The usefulness of the system for providing crack measurements for the State's future pavement management system or the current PES, can only be determined from extensive field usage.

REFERENCES

[1] Spangler, E. B., and Kelly, W. J. "GMR Road Profilom eter - A Method for Measuring Road Profiles," Research Publication GMR-452, Engineering Mechanics Department, General Motors Corpora tion, December 1964.

[2] Claros, German J., W. Ronald Hudson, and Clyde E. Lee, "Performance of the Analog and the Digital Profilometer with Wheels and with Non-Contact Transducers", Research Report 251-3F, Center for Highway Research, Bureau of Engineering Research, The University of Texas at Austin, April 1985.

[3] Walker, Roger S., and John Stephen Schuchman, "Upgrade of 690D Surface Dynamics Profilometer for Non-Contact Meas urements", Research Report 494-1F, The University of Texas at Arlington, December 1986.

[4] "Selcom Technical Manual", Selective Electronic Co., 1982.

[5] "Selcom User's Manual", Manual no. 04/3713/2, Selective Electronic Co., 1982.

[6] "Pavement Evaluation System Rater's Manual", Texas State Department of Highways and Public Transportation, 1987.

[7] "MC68000 : 16-/32-Bit Microprocessor", Motorola Semi conductors, Austin, Texas, 1985.

[8] Payne, Donnell Payne, Roger S. Walker, The Use Of Lasers For Pavement Crack Detection", Texas State Department of Highways and Public Transportation Research Report 1141-1, The University of Texas at Arlington, December 1988.

[9] "DT2817 Users Manual, IBM PC/XT/AT Bus Compatible Digital I/O Board", Data Translation, Inc., Marlboro, Ma., 1988

[10] "MVME133ABUG, 133ABug Debugging Package User's Manual", Motorola, Inc., 1987.

[11] "MVME133A-20, VMEmodule 32-Bit Monoboark Microcomputere User's Manual", First Edition, Motorola, Inc., 1987.

[12] "XVME-240 Digital I/O Module", XYCOM Co., 1984.

[13] Stone, Jerry, "Evaluation of the Laser Road Surface Tester For Measuring Measuring Pavement Roughness and Rut Depth, FHWA-DP-88-072-008, NTIS, May 1988.

APPENDIX A

OPERATION INSTRUCTIONS FOR CRACK IDENTIFICATION SYSTEM

-------- Codif Method --------

I. Preparations before starting system --

1) Create the following DOS directories:

```
X: /crack/program -- Store output file.
X: /crack/report  -- Store executable files.
                         userlag.exe.
                     Store screen file. screen1,
                                        screen3.lag
```

II. Procedures of operation --

1) Make sure that userlag.exe is accessible under current directory. Then type userlag and press ENTER.

2) Before entering the main menu, four questions must need to be answered. This system can support support three lasers running simultaneously.

```
Do you want to run Slave A ? (y/n)
Do you want to run Slave B ? (y/n)
Do you want to run Slave c ? (y/n)
Enter the header (output file names):
```

Note: When the error message that "parameters transmission time-out" is shown, check if master board in VME bus is active, or if a hardware transmission error occurred, or the acknowledge from VME bus is lost.

3) The main menu is shown as follows :

```
HEADER : ***

       A:  Active Laser bit  : A,B,
       L:  Lag = 4
       C:  Calculation using filtered data
       F:  Filter((smp/ft) = 192. (ft/cycle)=1
       S:  Speed = 30.0 mph
       T:  Threshold for Crack is 1000
       H:  Change the Header cof
      last sending word is 0

  F2-Quit System             F3-Count Mode
  F8-Debug Values            F10-Serial Talk
```

Option A : Change active laser bit according to the binary form of number given.
For example,
7(111) means A, B and C laser are active.

6(110) means B and C laser are active.
5(101) means A and C laser are active.
.... etc.

Option S : Change initial car speed (15 <==> 70). Car speed would be updated along with interrupt from external ARAM

Option L, C, F, T : Parameters related to downup method
Option H : Change the header name.

F2 simply exits system and return to DOS.
F3 enters Count Mode. See step 4.
F8 enters Debug Values Mode. See step 5.
F10 enters Serial Communication Mode. See step 6.

4) The screen of Count Mode is shown as follows :

Header : cof Active lasers on = 0,1,
Lag = 4 raw data used Filter(smp/ft)=192, (ft/cycle)=1
Speed = 30.0 mph Timer Constant = 118
Threshold for Crack is 1000

	LaserA	LaserB	LaserC	Comm A,B	Comm B,C
None					
1-3					
> 3					
Block					
Alleq					
Counts					
Errors					

F1-Change Parameter F2-Quit System F5-Reset Counter

F1 makes screen return to main menu and change parameters of lag algorithm.
F2 simply exits system and returns to DOS.
F5 resets all the counters to zero.

ps.1: The rows, None, 1-3 and >3, show the number of feet that contains cracks, none, 1-3 and >3.
2: Example --
For certain foot, if laser A detects 2 cracks and laser B detects 5 cracks, then there are 2 cracks for common A,B.
Laser A would be added 1 on 1-3 and added 2 on counts
Laser B would be added 1 on >3 and added 5 on counts
Comm A,B would be added 1 on 1-3 and added 2 on counts
3. The row, Errors, shows the number of feet that contains either invalid data or no data received at

all. Invalid data means the data is out of range. No data received means error with laserA or laserB.

5) Before entering Debug Values Mode, we have to tell system the detected laser number (laserA is 0, laserB is 1 , laserC is 2) and the scale number.

```
Enter the detected Laser number (0-2)  :
Enter the scale (50,100,200)  :
```

Then the screen of Show Slope Freq Mode is shown as follows :

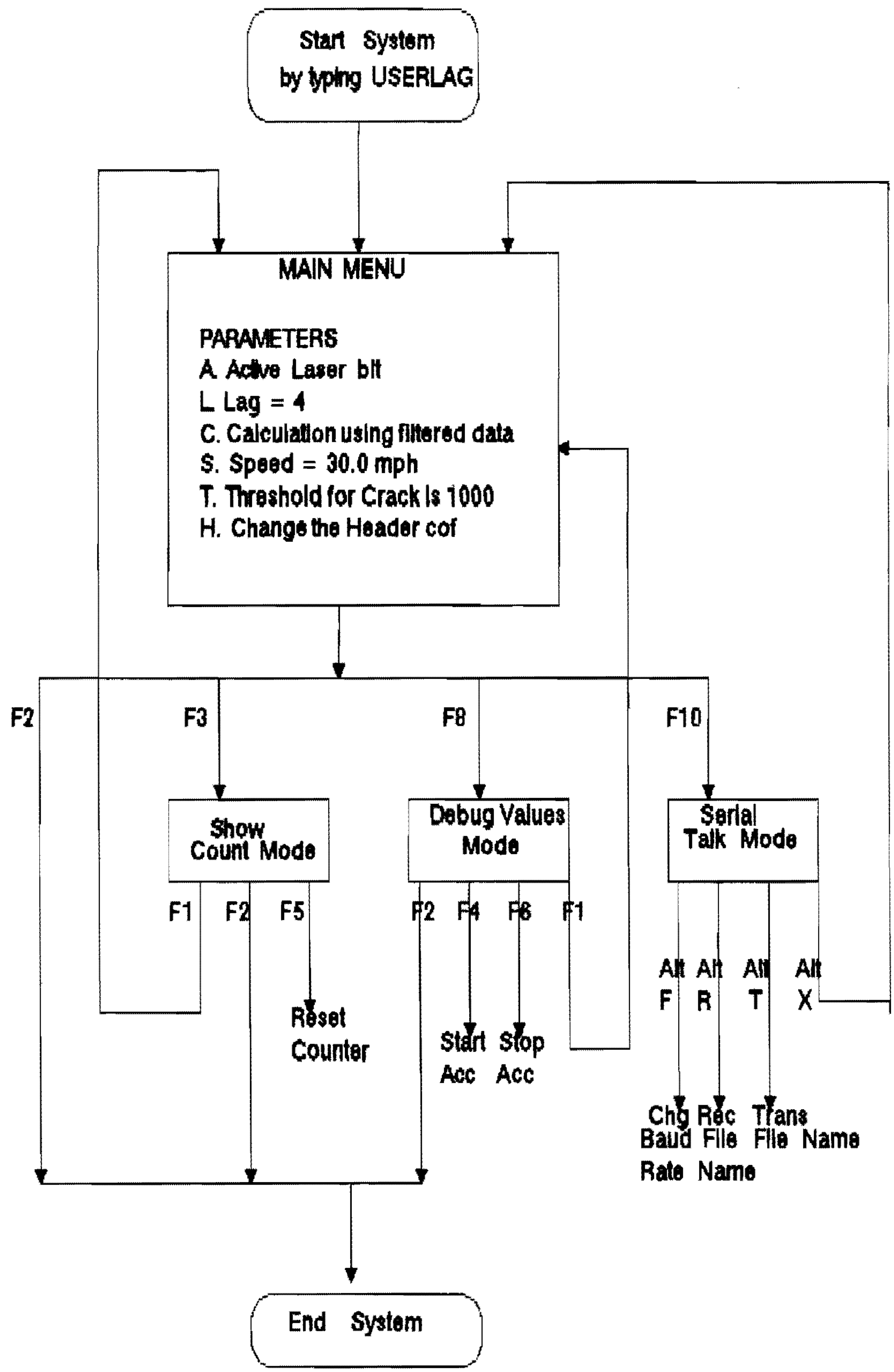
```
Frequency for Recent          inches
  Class    50      100     150     200      250    ---

 |______|_______|______|_______|_______|______|

F1-Rtn Change Mode          F2-Exit System
F4-Start Acc                F6-Stop Acc
```

F1 makes screen return to main menu and change parameters of downup algorithm.
F2 simply exits system and returns to DOS.
F4 starts acc.
F6 stops acc.

6) There are four options in Serial Talk mode.
Alt-F Parameters : change transmission baud rate.
Alt-R receive : enter file for receiving.
Alt-T transmit : enter file for transmitting.
Alt-X exit : return to main menu.

Figure A – 1

FLOW CHART OF OPERATION INSTRUCTIONS
(Codiff Algorithm)



APPENDIX B

OPERATION INSTRUCTIONS FOR CRACK IDENTIFICATION SYSTEM

-------- Downup Method --------

I. Preparations before starting system --

1) Create the following DOS directories:

```
x: /crack/program -- Store output file.
X: /crack/report  -- Store executable files.
                          userdu.exe.
                     Store screen file. screen1,
                                        screen3.du
```

II. Procedures of operation --

1) Make sure that userdu.exe is accessible under current directory. Then type userdu and press ENTER.

2) Before entering the main menu, four questions need to be answered. This system can support three lasers running simultaneously.

```
Do you want to run Slave A ? (y/n)
Do you want to run Slave B ? (y/n)
Do you want to run Slave c ? (y/n)
Enter the header (output file names):
```

ps. When the error message "parameters transmission time-out" is shown, check if master board in VME bus is active, or hardware transmission problems happen, so that acknowledge from VME bus is lost.

3) The main menu is shown as follows :

```
        1:  HEADER  : xxx
        2:  Active Laser bit  : A,B,
        3:  Car Speed = 35.0 mph
        4:  Points of Mean Bar = 4
        5:  Points of Slope Bar = 8
        6:  Difference on Slope Bar = 12
        7:  Threshold of Depth = 15
        8:  Maximum Crack Width = 64


   F2-Quit System           F3-Count Mode
   F8-Show Slope Freq       F10-Serial Talk
```

Option 1 : Rename header name.

Option 2 : Change active laser bit according to the binary form of number given.
For example,

7(111) means A, B and C laser are active.
6(110) means B and C laser are active.
5(101) means A and C laser are active.
.... etc.

Option 3 : Change initial car speed (15 <==> 70). Car speed would be updated along with interrupt from outside AREN signal.

Option 4--8 : Parameters related to downup algorithm.

F2 simply exits system and return to DOS.
F3 enters Count Mode. See step 4.
F8 enters Show Slope Freq Mode. See step 5.
F10 enters Serial Communication Mode. See step 6.

4) The screen of Count Mode is shown as follows :

Header : cof
Mbar = 4, Sbar = 8, Slope = 12
Car Speed = 35
Threshold for depth = 15, Maximum width = 64

	LaserA	LaserB	LaserC	Comm A,B	Comm B,C
None					
1-3					
> 3					
Block					
Alleq					
Counts					
Errors					

F1-Change Parameter F2-Quit System F5-Reset Counter

F1 makes screen return to main menu and change parameters of downup algorithm.
F2 simply exits system and returns to DOS.
F5 resets all the counters to zero.

ps.1: The rows, None, 1-3 and >3, show the number of feet that contains cracks, none, 1-3 and >3.

2: Example --
For certain feet, if laser A detects 2 cracks and B detects 5 cracks, then there are 2 cracks for common A,B.
Laser A would be added 1 on 1-3 and added 2 on counts
Laser B would be added 1 on >3 and added 5 on counts
Comm A,B would be added 1 on 1-3 and added 2 on counts

3. The row, Errors, shows the number of feet that contains either invalid data or no data received at all. Invalid data means the data is out of range.

No data received means error with laserA or laserB.

5) Before entering Show Slope Freq Mode, we have to tell system the detected laser number and the scale number.
   Enter the detected Laser number (0-2) :
   Enter the scale (8,16,32) :

   Then the screen of Show Slope Freq Mode is shown as follows :

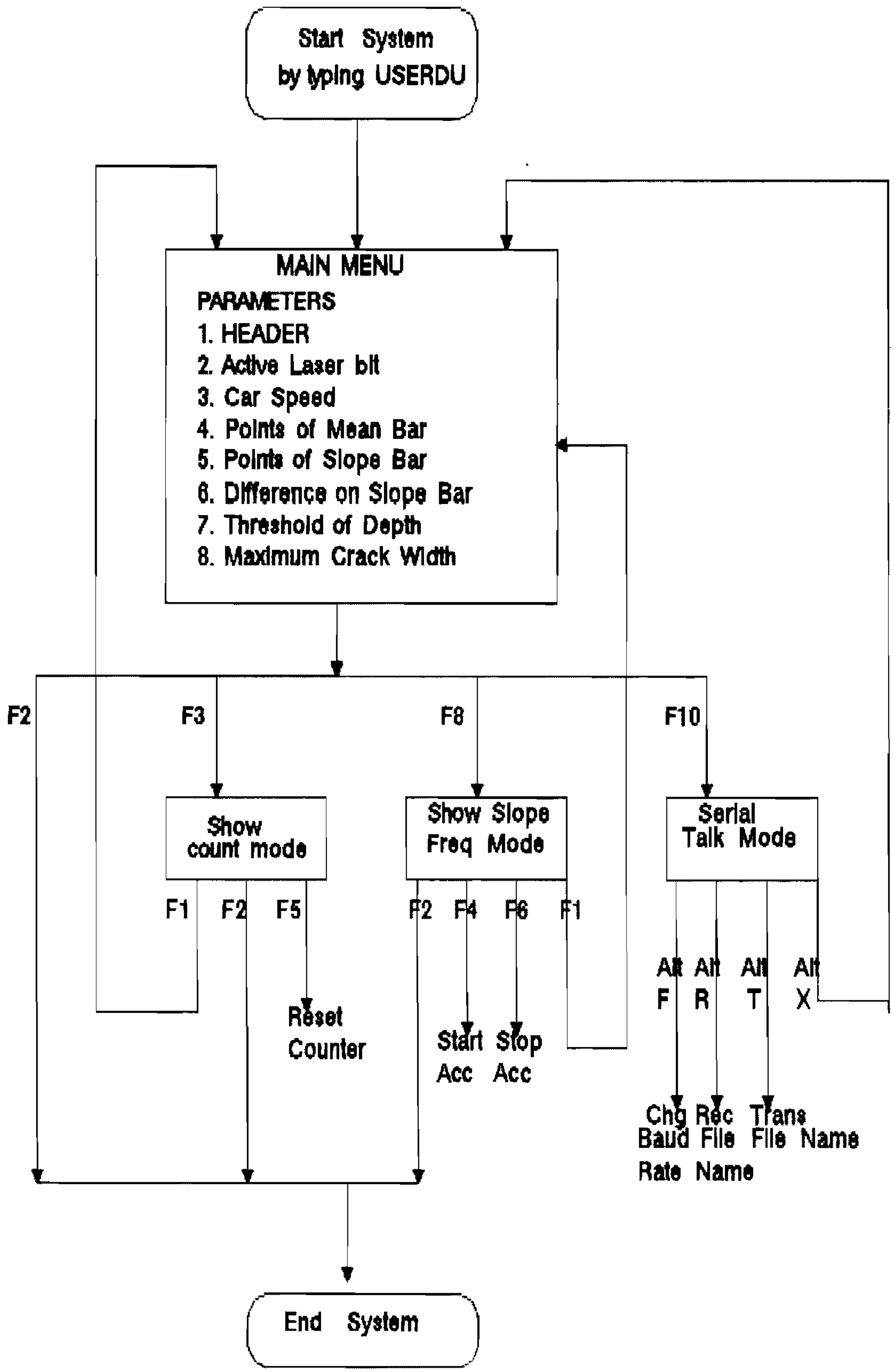
```
  Frequency for Recent          *7 points
    Class   8   16   24   32   ---

 F1-Rtn Change Mode    F2-Exit System
 F4-Start Acc          F6-Stop Acc
```

F1 makes screen return to main menu and change parameters of downup algorithm.
F2 simply exits system and returns to DOS.
F4 starts acc.
F6 stop acc.

6) There are four options in Serial Talk mode.
   Alt-F Parameters : change transmission baud rate.
   Alt-R receive : enter file for receiving.
   Alt-T transmit : enter file for transmitting.
   Alt-X exit : return to main menu.

Figure B – 1

FLOW CHART OF OPERATION INSTRUCTIONS
(Downup Algorithm)



APPENDIX C

XVME - 240 DIO and DT2817 -- INTERFACE

1) The pin function of JK1 on XVME - 240 DIO --

JK1 PIN ASSIGNMENT

PIN NUMBER	PORT	PIN FUNCTION
1	0	Data bit 0
2	0	Data bit 1
3	0	Data bit 2
4	0	Data bit 3
5	0	Data bit 4
6	0	Data bit 5
7	0	Data bit 6
8	0	Data bit 7
9	0*	Interrupt Input Line (Bit 0 of Interrupt Input Register)
10	0*	Flag Output Line (Bit 0 of Flag Output Register)
11	-	GND
12	-	GND
13	1	Data bit 0
14	1	Data bit 1
15	1	Data bit 2
16	1	Data bit 3
17	1	Data bit 4
18	1	Data bit 5
19	1	Data bit 6
20	1	Data bit 7
21	1*	Interrupt Input Line (Bit 1 of Interrupt Input Register)
22	1*	Flag Output Line (Bit 1 of Flag Output Register)
23	-	GND
24	-	GND
25	2	Data bit 0
26	2	Data bit 1
27	2	Data bit 2
28	2	Data bit 3
29	2	Data bit 4
30	2	Data bit 5
31	2	Data bit 6
32	2	Data bit 7
33	2*	Interrupt Input Line (Bit 2 of Interrrupt Input Register)
34	2*	Flag Output Line (Bit 2 of Flag Output Register)
35	-	GND

36	-	GND
37	3	Data bit 0
38	3	Data bit 1
39	3	Data bit 2
40	3	Data bit 3
41	3	Data bit 4
42	3	Data bit 5
43	3	Data bit 6
44	3	Data bit 7
45	3*	Interrupt Input Line (Bit 3 of Interrupt Input Register)
46	3*	Flag Output Line (Bit 3 of Flag Output Register)
47	-	GND
48	-	GND
49	-	GND
50	-	GND

2) The pin function of J1 on DT2817 --

J1 PIN ASSIGNMENTS

SIGNAL NAME	PIN	NO.	SIGNAL NAME
Digital Ground	1	2	Digital Ground
Port 0, bit 0	3	4	Port 1, bit 0
Port 0, bit 1	5	6	Port 1, bit 1
Port 0, bit 2	7	8	Port 1, bit 2
Port 0, bit 3	9	10	Port 1, bit 3
Port 0, bit 4	11	12	Port 1, bit 4
Port 0, bit 5	13	14	Port 1, bit 5
Port 0, bit 6	15	16	Port 1, bit 6
Port 0, bit 7	17	18	Port 1, bit 7
+5V Out (1A max)	19	20	+5V Out (1A max)
Digital Ground	21	22	Digital Ground
Digital Ground	23	24	Digital Ground
Digital Ground	25	26	Digital Ground
Digital Ground	27	28	Digital Ground
Digital Ground	29	30	Digital Ground
+5V Out (1A max)	31	32	+5V Out (1A max)
Port 2, bit 0	33	34	Port 3, bit 0
Port 2, bit 1	35	36	Port 3, bit 1
Port 2, bit 2	37	38	Port 3, bit 2
Port 2, bit 3	39	40	Port 3, bit 3
Port 2, bit 4	41	42	Port 3, bit 4
Port 2, bit 5	43	44	Port 3, bit 5
Port 2, bit 6	45	46	Port 3, bit 6
Port 2, bit 7	47	48	Port 3, bit 7
Digital Ground	49	50	Digital Ground

3) The connection of DIO JK1 and DT2817 J1 on connector --

The layout of connector

DIO		DT2817	
50	49	1	2
48	47	3	4
46	45	5	6
44	43	7	8
42	41	9	10
40	39	11	12
38	37	13	14
36	35	15	16
34	33	17	18
32	31	19	20
30	29	21	22
28	27	23	24
26	25	25	26
24	23	27	28
22	21	29	30
20	19	31	32
18	17	33	34
16	15	35	36
14	13	37	38
12	11	39	40
10	9	41	42
8	7	43	44
6	5	45	46
4	3	47	48
2	1	49	50

4) The connection of JK1 and J1

	DIO JK1	DT2817 J1
	44	48
P	43	46
O	42	44
R	41	42
T	40	40
	39	38
3	38	36
	37	34
	32	47
P	31	45
O	30	43
R	29	41
T	28	39
	27	37
2	26	35
	25	33
	20	18
P	19	16
O	18	14
R	17	12
T	16	10
	15	8
1	14	6
	13	4
	8	17
P	7	15
O	6	13
R	5	11
T	4	9
	3	7
0	2	5
	1	3

APPENDIX D

SELCOM OPTOCATOR and XVME240 DIO INTERFACE

1) The pin functions of j11 on OPTOCATOR :
J11 is digital parallel output form R-A unit no. 1 and R-A unit no. 2. We assign signal of laser A to J11 output from R-A unit 1, and assign signal of laser B to J11 output from R-A unit 2.

J11 PIN ASSIGNMENT

SIGNAL NAME	J11 output (R-A UNIT1)	J11 output (R-A UNIT2)	SIGNAL NAME
Invalid	a18	c18	Invalid
Spare	a19	c19	Spare
11 MSB	a20	c20	11 MSB
10	a21	c21	10
9	a22	c22	9
8	a23	c23	8
7	a24	c24	7
6	a25	c25	6
5	a26	c26	5
4	a27	c27	4
3	a28	c28	3
2	a29	c29	2
1	a30	c30	1
0 LSB	a31	c31	0 LSB
Flag	a32	c32	Flag
Ground	a4,a5,a6	c4,c5,c6	Ground

2) The pin function of JK2 on XVME-240 DIO --

JK2 PIN ASSIGNMENT

PIN NUMBER	PORT	PIN FUNCTION
1	4	Data bit 0
2	4	Data bit 1
3	4	Data bit 2
4	4	Data bit 3
5	4	Data bit 4
6	4	Data bit 5
7	4	Data bit 6
8	4	Data bit 7
9	4*	Interrrupt Input Line (Bit 4 of Interrupt Input Register)
10	4*	Flag Output Line (Bit 4 of Flag Output Register)
11	-	GND
12	-	GND
13	5	Data bit 0
14	5	Data bit 1
15	5	Data bit 2
16	5	Data bit 3
17	5	Data bit 4
18	5	Data bit 5
19	5	Data bit 6
20	5	Data bit 7
21	5*	Interrupt Input Line (Bit 5 of Interrupt Input Register)
22	5*	Flag Output Line (Bit 5 of Flag Output Register)
23	-	GND
24	-	GND
25	6	Data bit 0
26	6	Data bit 1
27	6	Data bit 2
28	6	Data bit 3
29	6	Data bit 4
30	6	Data bit 5
31	6	Data bit 6
32	6	Data bit 7
33	6*	Interrupt Input Line (Bit 6 of Interrupt Input Register)
34	6*	Flag Output Line (Bit 6 of Flag Output Register)
35	-	GND
36	-	GND

37	7	Data bit 0
38	7	Data bit 1
39	7	Data bit 2
40	7	Data bit 3
41	7	Data bit 4
42	7	Data bit 5
43	7	Data bit 6
44	7	Data bit 7
45	7*	Interrupt Input Line (Bit 7 of Interrupt Input Register)
46	7*	Flag Output Line (Bit 7 of Flag Output Register)
47	-	GND
48	-	GND
49	-	GND
50	-	GND

3) The connection of OPTOCATOR and XVME DIO on connector

The layout of the connector

OPTOCATOR		DIO	
c1	a1	49	50
c2	a2	47	48
c3	a3	45	46
c4	a4	43	44
c5	a5	41	42
c6	a6	39	40
c7	a7	37	38
c8	a8	35	36
c9	a9	33	34
c10	a10	31	32
c11	a11	29	30
c12	a12	27	28
c13	a13	25	26
c14	a14	23	24
c15	a15	21	22
c16	a16	19	20
c17	a17	17	18
c18	a18	15	16
c19	a19	13	14
c20	a20	11	12
c21	a21	9	10
c22	a22	7	8
c23	a23	5	6
c24	a24	3	4
c25	a25	1	2
c26	a26		
c27	a27		
c28	a28		
c29	a29		
c30	a30		
c31	a31		
c32	a32		

4) The pin connection of J11 and JK2

	JK2 pin	J11 pin
P	1	a22
O	2	a23
R	3	a20
T	4	a21
4	5	a18
	6	a32
	13	a31
P	14	a30
O	15	a29
R	16	a28
T	17	a27
5	18	a26
	19	a25
	20	a24
G		
N	23	b4
D	24	b5
	25	b22
P	26	b23
O	27	b20
R	28	b21
T	29	b18
6	30	b32
G	35	b31
N	36	b30
D		
	37	b29
P	38	b28
O	39	b27
R	40	b26
T	41	b25
7	42	b24

APPENDIX E

AUTOCORRELATION DIFFERENCE METHOD

```
/************************************************************************
/*
/*  PROGRAM NAME : CODIFF.C
/*
/*  AUTHOR: Chuying Kuo
/*
/*  DATE  : Jan, 1988
/*
/*  PURPOSE: This C program is the preversion of MC68020 assembly code
/*           for detecting the crack by using the autocorelation
/*           difference algorithm for processing the distance
/*           measurement from laser system.
/*  ALGORITHM: this program is to calculate the codifference value
/*             R(0)-R(4) according to the formula :
/*    R(0) = (sumxx - (sumx * sumx)/(vrange-lag) )/(vrange-lag); */
/*    R(4) = (sum(x[i]x[i+4] - (sum(x[i]) * sum(x[i+4])/(vrange-lag) )
/*            /(vrange-lag); */
/***********************************************************************/

#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
#include <alloc.h>

#define VRANGE  16

   char tail, *inname, *outname;
   int i,j, eof;
   FILE  *infile, *outfile;
   long data,grnd;


void compute(void)

{  int m, c, lag=4;
   long value[VRANGE+1];
   long  sumx=0, sumy=0,sumxi=0, sumxy=0,  sumxx=0,diff;
   long autocorelate, variance;

  while(1)
      {
                 for  (m = 1; m <= VRANGE; m++)
                        {
   /************************************/
   /* get data and accumulate the sums
   /************************************/
                          if ( fscanf(infile, "%ld ", &data)==EOF)
                                    exit(1);
```

```
if (m==VRANGE)
    fprintf(outfile, "%ld  ",data);
else
    fprintf(outfile, "%ld \n ",data);

data -= grnd;
sumx = sumx + data;
value[m] =data;
if (m<=VRANGE-lag)
    sumxi = sumxi + data;
if (m> lag)
   {
     sumy = sumy + data;
     sumxy = sumxy + value[m] * value[m-lag];
   }
sumxx = sumxx + data * data;
}
```

```
/*
```

```
*/
   /***********************************/
   /* calculate covariances and codifference
   /***********************************/
                   variance = (sumxx - (sumx * sumx)/VRANGE )
                               /VRANGE;
                   autocorelate = (sumxy - (sumxi * sumy)/(VRANGE-lag) )
                               /(VRANGE-lag);

                   diff = variance - autocorelate;

   /***********************************/
   /* initialize the accumulate variables
   /***********************************/
                   sumx = 0;
                   sumxi = 0;
                   sumy = 0;
                   sumxx = 0;
                   sumxy = 0;

                    if (diff >1000 )
                          diff = 1000;
                    fprintf(outfile,"  %ld \n",
                          diff);
         }
}


main()

{
  int temp;

   printf("enter input file name :");
   scanf("%s",inname);
   if ( (infile =fopen(inname,"r")) == NULL)
        { printf("Can't open file");
          exit(i) ;
        }

   printf("enter output file name :");
   scanf("%s",outname);
     if ( (outfile =fopen(outname,"w")) == NULL)
        { printf("Can't open outfile");
          exit(i) ;
        }


   printf("enter ground value :");
   scanf("%ld",&grnd);   /* get the scale value */
   compute();
   fclose(infile);
```

```
fclose(outfile);}
```

APPENDIX F

RUNNING-MEAN DOWNUP METHOD

```
/*********************************************************************/*
/*  PROGRAM NAME : DOWNUP.C
/*
/*  AUTHOR: Chuying Kuo
/*
/*  DATE   : Mar 19, 1989
/*
/*  PURPOSE:    This C program is the C version of MC68020 for
/*              detecting the crack by using down&up algorithm. for
/*  ALGORITHM: The idea behind the algorithm is that a crack is formed
/*                 by a sharp negative (down) slope  following a
/*                 sharp postive (up) slope based on the relative
/*              distance from the road surface to the laser probe.
/*              A running mean value is used to filter the noise
/*              from laser measurement and establish the base.
/**********************************************************************/
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>
#include <alloc.h>
#include <math.h>
#include <float.h>

   char inname[30], outname[30];
   long data, depth=0,diff,max,tc, pos, slope, width;
   long negative[2];/* store the negative slope  position and depth*/
   long mean[17],rmean, rtotal,mbar,mi, value[17],sbar,si ;
   long i,j, end, start;
   FILE  *infile, *outfile, *repfile;
```

```
/**************************************************************
/* This routine is to get the input parameter and initialize
/* the mean buffer , the data value buffer and the variables.
/**************************************************************/
initialize()
{
     printf("Enter input file name :");
     scanf("%s", inname);
     if ( (infile =fopen(inname,"r")) == NULL)
        { printf ("Can't open file");
          exit(i) ;
        }

     printf("Enter no. of points of mean :");
     scanf("%ld", &mbar);
     printf("Enter no. of points for the slope bar :");
     scanf("%ld", &sbar);
     printf("Enter the normal slope for the slope bar :");
     scanf("%ld", &slope);
     printf("Enter the threshold for the crack depth :");
     scanf("%ld", &tc);
     printf("Enter the maximum number of points for a crack :");
     scanf("%ld", &max);
     if ((outfile = fopen("paslope.lst","w")) == NULL)
             { printf("Can't open outfile");
               exit(i);
             }
     if ((repfile = fopen("paslope.rep","w")) == NULL)
             { printf("Can't open  report file");
               exit(i);
             }
         fscanf(infile, "%ld ", &data);
         rtotal = data *mbar;
         for (i=0;i<mbar;i++)
              value[i] = data;
         mi=0;
         si = 0;
         for (i=0;i<sbar;i++)
              mean[i] = data;
         diff = 0;
         depth =0;
         fprintf(outfile, "%ld  %ld  %ld\n",data, rmean,diff);

}  /* initialize */
```

```
/***********************************************************
/* This routine is to find the crack with its width and depth
/* by looking for a pair of sharp slopes, which are negative
/* first and positive next
/***********************************************************/
find_crack()
{
     if (depth == 0)
         if (diff > slope || diff < -slope)
             {
                 depth = diff;
                 start = pos-1;
             }
         else;  /* normal profile frequency value, ignored*/
     else
     if (depth > 0)
          if (diff < slope)
              {
                  end = pos-sbar;
                  if (negative[1] !=0)
                         {
      /*                   if (negative[1]< -depth)
                               depth = negative[1];
                           else
                               depth = -depth;
      */                   /* write start,(depth),end */
                           depth = (-negative[1] + depth )/2;
                           fprintf(repfile, "%ld %ld %ld\n ",
                                negative[0], depth , end-negative[0]);
                         }
                      else;    /* unmatched positive ignore*/
                  negative[1] = 0; /* delete negative record */
                  depth = 0;
              }
          else  /* diff still > slope */
              if (depth < diff)
                  depth = diff;
              else ;  /* depth not incresed */

     else   /* depth < 0 */
          if (diff > -slope)
              {
                   negative[0]= start;
                   negative[1] = depth;
                   depth = 0;
              }
          else  /* diff still < -slope */
              if (depth > diff)
                  depth = diff;
              else ;  /* depth not decresed */
}  /* find_crack */
```

```
/*************************************************************
/* This routine is to calculate the  running mean and slope for
/* the slope bar ; store them to proper buffer; point to next
/**************************************************************/
calculate_mean_slope()
{
      pos++;
      value[(mi+mbar)&0xf]=data;
      rtotal = rtotal+data-value[mi];
      rmean = rtotal/mbar;
      mean[(si+sbar)&0xf] = rmean;
      diff = rmean-mean[si];
      mi++;
      mi&=0xf;
      si++;
      si&=0xf;
      fprintf(outfile, "%ld %ld %ld \n",data,rmean, diff);

} /* calculate mean slope */

/****************************************************************
/* This routine reads in the crack report data and then translate
/* them into lotus plots input data format
/**************************************************************/
trans_crack_inform()
{
    int no=0;
    long depth, pos, width;
     if ((infile = fopen("paslope.rep","r")) == NULL)
              { printf("Can't open  report file for another infile");
                exit(i);
              }

     if ((outfile = fopen("crack.lst","w")) == NULL)
              { printf("Can't open second outfile ");
                exit(i);
              }

      for (pos=1;fscanf(infile, "%ld %ld %ld ", &start,&depth,
                &width)!=EOF;);
          {
            for (;pos < start; pos++)
                  fprintf(outfile, "0 \n");

            if (width < max && depth >= tc)
               {
                for (;pos < width+1+start; pos++)
                      fprintf(outfile, "%ld \n",-depth );

                no++;
               }
          }
```

```
    printf("total no. of cracks is %d \n",no);
    fclose(outfile);
    fclose(infile);

}  /* trans_crack_inform */
```

```
/**************************************************************
/****** main program control the whole program sequence  ***** /
/**************************************************************/
main()

{
     initialize();
     for (pos=1;fscanf(infile, "%ld  ", &data)!=EOF;)
          {
               calculate_mean_slope();
               find_crack();
          }

     /* write down the end records */
     fprintf(repfile, "%ld  0  %ld\n ", pos, pos);

     fclose(outfile);
     fclose(repfile);
     fclose(infile);

     trans_crack_inform();

} /* main program */
```