

N70-12026
1MX-61314

NASA CONTRACTOR
REPORT

Report No. 61314

CASE FILE
COPY

COMPUTER PROGRAM FOR CROSSED-BEAM STUDIES
OF CLEAR AIR TURBULENCE PROGRAM
DESCRIPTION (MLTCOR)

By A. Wachowski and M. R. Phillips

IIT Research Institute
10 West 35th Street
Chicago, Illinois 60616

September 1969

Final Report

Prepared for

NASA-GEORGE C. MARSHALL SPACE FLIGHT CENTER
Marshall Space Flight Center, Alabama 35812

1. REPORT NO. NASA CR-61314		2. AUTHOR AND AUTHOR NO.		3. RECIPIENT'S CATALOG NO.	
4. TITLE AND SUBTITLE COMPUTER PROGRAM FOR CROSSED-BEAM STUDIES OF CLEAR AIR TURBULENCE PROGRAM DESCRIPTION (MLTCOR)				5. REPORT DATE September 1969	
				6. PERFORMING ORGANIZATION CODE	
7. AUTHOR(S) A. Wachowski and M. R. Phillips				8. PERFORMING ORGANIZATION REPORT NO. J6145	
9. PERFORMING ORGANIZATION NAME AND ADDRESS IIT Research Institute 10 West 35th Street Chicago, Illinois 60616				10. WORK UNIT NO.	
				11. CONTRACT OR GRANT NO. NAS8-21300	
				13. TYPE OF REPORT & PERIOD COVERED Contractor Report May 14, 1968, Aug. 13, 1969	
12. SPONSORING AGENCY NAME AND ADDRESS NASA-George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812				14. SPONSORING AGENCY CODE	
15. SUPPLEMENTARY NOTES Distribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the author or organization that prepared it.					
16. ABSTRACT The multichannel correlation program (MLTCOR) described in this report is fairly general and was developed to calculate statistics for aerodynamic and atmospheric turbulence based on the crossed-beam technique. The piecewise concept was used in writing this program. A long-time series is logically broken up into a series of smaller pieces. Correlations, mean values, root-mean-square (RMS) values and variances are then calculated for each piece along with their standard statistical errors. A dynamic storage allocation scheme was used in the program to provide maximum storage for data and to avoid the use of fixed length data arrays. Correlations can be performed on essentially infinite length records. Time series data are brought into the computer in blocks to fill the available storage, and data for the maximum time lag are saved from the previous block to make correlations continuous across the core loads of data. The program is written in FORTRAN IV using modular techniques and dynamic storage allocation. The mathematical formulation was developed in-house by NASA and the programming and interfacing to MSFC facilities was accomplished by IITRI.					
17. KEY WORDS			18. DISTRIBUTION STATEMENT PUBLIC RELEASE <i>E. D. Geissler</i> E. D. GEISSLER Dir, Aero-Astrodynamic Lab, MSFC		
19. SECURITY CLASSIF. (of this report) U		20. SECURITY CLASSIF. (of this page) U		21. NO. OF PAGES 135	
				22. PRICE	

FOREWORD

This final report, entitled "Computer Program for Crossed-Beam Studies of Clear Air Turbulence, Program Description (MLTCOR)," presents the results of research undertaken by IIT Research Institute (IITRI) from 14 May 1968 to 13 August 1969. The research was performed for National Aeronautics and Space Administration (NASA), Marshall Space Flight Center, Alabama, under Contract NAS8-21300 (IITRI Project J6145).

ACKNOWLEDGEMENT

The authors would like to take this opportunity to make acknowledgement to F. R. Krause and R. R. Jayroe for their cooperation and technical assistance in developing the computer code for (MLTCOR) and also to J. A. Jones for making available, as needed, computational support.

CONTENTS

<u>Section</u>	<u>Page</u>
SUMMARY	1
1. MATHEMATICAL FORMULATION	2
A. Definition of Symbols	3
1. Independent Variables	3
2. Operators	5
3. Special Notations	6
B. Basic Computations	6
2. GENERAL DESCRIPTION	11
3. DYNAMIC STORAGE ALLOCATION	16
4. LINK 1 - THE INPUT LINK	19
A. Input Tapes	19
B. Input Problem Parameters	24
C. Central Differences	24
D. Moving Mean and Sample Reduction	25
E. Calibration	25
1. Sine Wave Calibration	25
2. Step Calibration	27
F. ESSA Data Logger Output	27
G. MSFC Digitizer Output	29
5. LINK 2 - THE CORRELATION LINK	29
6. LINK 3 - THE OUTPUT LINK	40
7. USAGE	46
A. Input	46
B. Output	52
REFERENCES	63
APPENDIX A - MLTCOR COMPUTER PROGRAM USERS MANUAL	65
1. Introduction	65
2. Input Problem Parameters	65
3. Input/Output Tapes	70
4. Diagnostic	70
5. Users Restrictions and Notes	71
6. Job Setup	73
APPENDIX B - PROGRAM LISTINGS	77
APPENDIX C - DETRENDING AND NORMALIZATION OF CORRELATION FUNCTION	123
APPENDIX D - EXPECTED ERROR CURVES	131

ILLUSTRATIONS

<u>Figure</u>	<u>Page</u>
1. Basic Program Structure	12
2. Overlay Diagram	14
3. Main Program	15
4. LINK 1-1	20
5. LINK 1-2	21
6. ESSA Data Logger Output Format	22
7. MSFC Digitizer Output Format	23
8. ESSA Header Record	28
9. MSFC Header Record	29
10. LINK 2-1	30
11. LINK 2-2	31
12. Correlation Request	33
13. Tables	33
14. LINK 2 Storage Allocation	35
15. Storage Use	39
16. LINK 3-1	41
17. LINK 3-2	42
18. LINK 3 Storage Allocation	45
19. LINK 1 Reports	54
20. LINK 2 Channel Averages	55
21. Piecewise Correlations	56
22. LINK 3 Modified Correlation	57
23. Final Tabulation	58
24. Piecewise Correlation	59
25. Piecewise Modified Correlation	60
26. Accumulative Correlation and Standard Error	61
27. Accumulative Modified Correlation and Standard Error	62

TABLES

I	Mathematical Analysis of Stationary, Infinitely Long-Time Histories	4
II	Basic Averages	7
III	Accumulation Procedures	9
IV	Standard Errors	10
V	Piecewise Statistics	37
VI	Modified Correlations	44
VII	Expected Error and Confidence Levels	134

COMPUTER PROGRAM FOR CROSSED-BEAM STUDIES
OF CLEAR AIR TURBULENCE
PROGRAM DESCRIPTION (MLTCOR)

A. Wachowski
M. R. Phillips
IIT Research Institute

SUMMARY

The multichannel correlation program (MLTCOR) described in this report is fairly general and was developed to calculate statistics for aerodynamic and atmospheric turbulence based on the crossed-beam technique.

A dynamic storage allocation scheme was used in the program to provide maximum storage for data and to avoid the use of fixed length data arrays. Correlations can be performed on essentially infinite length records. Time series data are brought into the computer in blocks to fill the available storage, and data for the maximum time lag are saved from the previous block to make correlations continuous across the core loads of data. The program is written in FORTRAN IV using modular techniques and dynamic storage allocation.

The piecewise concept was used in writing this program. A long-time series is logically broken up into a series of smaller pieces. Correlations, mean values, root-mean-square (RMS) values and variances are calculated for each piece along with their standard statistical errors.

The program uses strong and very effective detrending and normalization techniques (see Appendix C) developed over the years to cope with problems encountered while attempting to derive meaningful information from aerodynamic and atmospheric data. The detrending and normalization techniques employed minimizes the effects of DC shifts and trends as well as drifting

gains which are unavoidable. Until recently, methods used were not adequate for extracting meaningful information for reliable data interpretation.

The associated errors are calculated as guidelines for interpreting the data (see Appendix D). The parameters can be monitored concurrently with "piecewise" averages for analyzing data behavior. When these error parameters exceed "built-in" confidence limits, this reflects a degree of stationarity that would not be acceptable to the previous accumulation of pieces.

The program provides for accumulating over separate periods of the data record with provisions for deleting individual data pieces within the accumulation period. With this provision data pieces can be discarded or accumulated in such a fashion that only the stationary pieces (within certain confidence limits) are used in the accumulation.

The mathematical formulation was developed in-house by NASA and the programming and interfacing to MSFC facilities was accomplished by IITRI.

1. MATHEMATICAL FORMULATION

The purpose of the computer program is to statistically analyze a given set of simultaneously recorded time histories which represent the output of a group of meteorological sensors. The desired statistical analysis is based on the mathematical model (Ref. 1) of a stationary and infinitely long-time history.

Within this model, accurate mean values, RMS values, correlation functions and covariance functions would be defined by the time integrals listed in Table I. Each of the correlation calculations would be performed for several combinations of signal pairs. One signal, x_j , is always to be taken from the group of records $j = 1, 2, 3$, that is to be delayed in time. The other signal, y_k , is to be taken from the remaining signals $k = k_1, k_2 \dots k_N$, for which the need of a time delay is not anticipated.

The records of meteorological sensors are unfortunately neither stationary nor infinitely long. The actual calculations will therefore provide only various approximations to the mathematical model. To define these mathematical approximations adequately, it is necessary to define a number of terms and to introduce the symbols used in defining the operations to be performed on the input data.

A. Definition of Symbols

1. Independent Variables.-

t	Observation time
T	Integration time
ΔT	Piece length
i	Piece number
e	Sampling period
$\frac{1}{e}$	Sampling rate
b	Sample reduction factor
$\frac{1}{be}$	Effective sampling rate
$\bar{e} = be$	Effective sampling period
$M = \frac{\Delta T}{be}$	Number of samples per piece
τ	Time lag

TABLE I
MATHEMATICAL ANALYSIS OF STATIONARY,
INFINITELY LONG-TIME HISTORIES

DESIRED STATISTICS	TIME INTEGRATION
Mean Values	$\langle x \rangle_j = \lim_{T_q - T_{q-1} \rightarrow \infty} \frac{1}{T_q - T_{q-1}} \int_{t=T_{q-1}}^{t=T_q} x_j(t) dt$ $\langle y \rangle_k = \lim_{T_q - T_{q-1} \rightarrow \infty} \frac{1}{T_q - T_{q-1}} \int_{t=T_{q-1}}^{t=T_q} y_k(t) dt$
Mean Square Values	$\langle x^2 \rangle_j = \lim_{T_q - T_{q-1} \rightarrow \infty} \frac{1}{T_q - T_{q-1}} \int_{T_{q-1}}^{T_q} x_j^2(t) dt$ $\langle y^2 \rangle_k = \lim_{T_q - T_{q-1} \rightarrow \infty} \frac{1}{T_q - T_{q-1}} \int_{T_{q-1}}^{T_q} y_k^2(t) dt$
Second Moment About Mean	$\langle \sigma^2 \rangle_j = \lim_{T_q - T_{q-1} \rightarrow \infty} \frac{1}{T_q - T_{q-1}} \int_{t=T_{q-1}}^{T_q} (x_j(t) - \langle x \rangle_j)^2 dt$ $\langle \sigma^2 \rangle_k = \lim_{T_q - T_{q-1} \rightarrow \infty} \frac{1}{T_q - T_{q-1}} \int_{t=T_{q-1}}^{T_q} (y_k(t) - \langle y \rangle_k)^2 dt$
Correlation Function $R_{jk}(\tau)$	$R_{jk}(\tau) = \lim_{T_q - T_{q-1} \rightarrow \infty} \frac{1}{T_q - T_{q-1}} \int_{t=T_{q-1}}^{T_q} (x_j(t-\tau) y_k(t) dt$
Covariance Function $C_{jk}(\tau)$	$C_{jk}(\tau) = R_{jk}(\tau) - \langle x \rangle_j \langle y \rangle_k$

$\tau_{\max}$	Maximum time lag
$\tau_{\min}$	Minimum time lag
$\Delta\tau$	Time lag increment
$L = \frac{\tau_{\max} - \tau_{\min}}{\Delta\tau}$	Number of lag points
$\beta = \frac{\Delta\tau}{\bar{e}}$	Number of samples per lag interval
J	Index of delayed channel
K	Index of undelayed channel
X	Delayed channel
Y	Undelayed channel
t_q	Accumulation start time
t_{q+1}	Accumulation end time

2. Operators.-

()	Statistic
$\overline{(\quad)}_i = \frac{1}{\Delta T} \int_{T=(i-1)\Delta T}^{T=i\Delta T} (\quad) dt$	Piecewise mean
$\overline{\overline{(\quad)}}_m = \frac{1}{m} \sum_{i=1}^m \overline{(\quad)}_i$	Accumulative mean
$\Delta^2 \overline{(\quad)}_m = \frac{1}{m-1} \sum_{i=1}^m \left(\overline{(\quad)}_i - \overline{\overline{(\quad)}}_m \right)^2$	Piecewise statistical error
$\Delta^2 \overline{\overline{(\quad)}}_m = \frac{1}{m} \Delta^2 \overline{(\quad)}_m$	Accumulative statistical error
$\tau \overline{(\quad)} = \frac{1}{2\tau_{\max}} \int_{-\tau_{\max}}^{+\tau_{\max}} \overline{(\quad)} d\tau$	Average over time lag

3. Special Notations.- To indicate that an average has been performed on a basic quantity, which is dependent upon a number of variables, a special notation is used. A subscript to the right of a symbol indicates that the subscript is variable, while a subscript to the left of a symbol indicates an average has been performed with respect to this subscript. Thus, the basic piecewise accumulative mean $(\overline{\quad})$ may be dependent on the delayed and undelayed channels j and k , and the time lag τ . This would be indicated as

$$(\overline{\quad})_{jk\tau}$$

The average over time lags would be indicated as

$$\tau(\overline{\quad})_{jk} = \frac{1}{2\tau} \int_{-\tau_{\max}}^{+\tau_{\max}} (\overline{\quad})_{\tau jk} d\tau$$

This special notation is required to indicate some of the error terms calculated in the program.

B. Basic Computations

The processing of very long records through a computer with a limited storage capacity is possible by breaking the record into successive pieces of length ΔT . A time integral over such a single piece is then approximated by summation over the samples $n = 1, 2, \dots, M$, which are contained in the piece. Averages over longer periods may then be established by accumulating the information from all pieces, $i = 1, 2, \dots, m$. The resultant "accumulative" mean can be averaged further by summing over all time lags (Ref.2) $\ell = -L + 1, -L + 2, \dots, -1, 0, 1, \dots, +L$ and over several specially selected channels ($j = 1, 2, 3$ and $k = k_1, k_2, k_N$). Table II gives a list of such basic averages.

The deviations from stationarity may also be partially removed by assigning statistical weights and ordinate shifts (Ref. 3 and Appendix C) to each piece.

TABLE II
BASIC AVERAGES

AVERAGE	SAMPLING	SUMMATION
Piecewise Average (Single piece)	$t = n \bar{\epsilon}$ $\Delta T = M\bar{\epsilon}$	$\overline{(\quad)}_i = \frac{1}{M} \sum_{m=1}^M (\quad)_{n=(i-1)M+m}$
Accumulative Average (All previous pieces)	$i = 1, 2, \dots, m$ $T_q - T_{q-1} = m\Delta T$	$\overline{\overline{(\quad)}} = \text{Output of accumulation subroutine, Table III}$
Time Lag Average (All lags and pieces)	$i = 1, 2, \dots, m$ $ \tau_{\max} - \tau_{\min} \leq L\beta\bar{\epsilon}$ $\pm \tau_{\ell} = \pm \tau_{\min} + \ell\beta\bar{\epsilon}$	$\tau(\overline{\overline{(\quad)}}) = \frac{1}{2L} \sum_{\ell=-L+1}^{+L} \overline{\overline{(\quad)}}(\tau_{\ell})$
Channel Average (Selected channels all pieces)	$j = 1, 2, 3$ $k = k_1; k_2; k_3; \dots k_N$	${}_{jk}\overline{\overline{(\quad)}}^2 = \frac{1}{N} \left\{ \overline{\overline{(\quad)}}_{jk_1}^2 + \overline{\overline{(\quad)}}_{jk_2}^2 + \overline{\overline{(\quad)}}_{jk_3}^2 + \overline{\overline{(\quad)}}_{jk_N}^2 \right\}$

This provides a multitude of different accumulative averages, which differ from a straight time integral by the particular weight, and shifts that were chosen. Four such accumulation procedures (listed in Table III) are used since they have proved very effective in removing temporal trends of statistical averages.

The desired statistical averages listed in Table I cannot be calculated exactly, since any deviation from stationarity and infinite record length will introduce statistical errors. The standard statistical errors of the various averages listed in Table II can be calculated by employing the deviations from the piecewise means from the accumulative means. The associated standard errors are listed in Table IV. These standard errors are functionally dependent on integration time, $T = T_q - T_{q-1}$ and allows one to indirectly determine some dominant deviations from stationarity. Furthermore, these errors are required whenever the confidence limits of the various averages are requested.

Each new piecewise average i , is used to continuously update the accumulative averages over the previous m pieces. This leads to two recursion formulas for updating the accumulative averages (Ref.4) and the standard error of these averages.

The recursion relationships are as follows.

Accumulative averages:

$$\overline{(\overline{\quad})}_{m+1} = \frac{m}{m+1} \overline{(\overline{\quad})}_m + \frac{1}{m+1} \overline{(\quad)}_{m+1}$$

Standard error of accumulative average:

$$\begin{aligned} \Delta^2 \left[\overline{(\overline{\quad})}_{m+1} \right] &= \frac{m-1}{m} \Delta^2 \left[\overline{(\quad)}_m \right] \\ &+ \frac{m+1}{m^2} \left[\overline{(\quad)}_{m+1} - \overline{(\overline{\quad})}_{m+1} \right]^2 \end{aligned}$$

TABLE III
ACCUMULATION PROCEDURES

TYPE	ACCUMULATION SUBROUTINE, $\overline{\overline{(\quad)}}$
Straight Time Integration	$\overline{\overline{(\quad)}}_m = \frac{1}{m} \sum_{i=1}^m \overline{(\quad)}_i$
Statistical Weighting	$\overline{\overline{(\quad)}}_w = \frac{1}{m} \sum_{i=1}^m \frac{\overline{(\quad)}_i}{\overline{\sigma}_{ji} \overline{\sigma}_{ki}}$
Ordinate Shifts	$\overline{\overline{(\quad)}}_d = \frac{1}{m} \sum_{i=1}^m \left\{ \overline{(\quad)}_i (\tau_\ell) - \tau \overline{(\quad)}_i \right\}$
Combined Weighting and Shifting*	$\overline{\overline{(\quad)}}_c = \frac{1}{m} \sum_{i=1}^m \frac{\overline{\overline{(\quad)}}_i (\tau_\ell) - \tau \overline{(\quad)}_i}{\overline{\sigma}_{ji} \overline{\sigma}_{ki}}$

* If the piecewise means are not given as a function of time lag, the accumulations d and c are to be replaced with the substitutions

$$\overline{\overline{(\quad)}}_d = \overline{\overline{(\quad)}}_m \quad \text{and} \quad \overline{\overline{(\quad)}}_c = \overline{\overline{(\quad)}}_w.$$

TABLE IV
STANDARD ERRORS

TYPE	SUMMATION
Standard Error of Piecewise Average	$\Delta^2 [\overline{\tau}]_m = \frac{1}{m-1} \sum_{i=1}^m (\tau_i - \overline{\tau}_m)^2$
Standard Error of Accumulative Average	$\Delta^2 [\overline{\tau}]_m = \frac{1}{m} \Delta^2 [\overline{\tau}]_m = \frac{1}{m(m-1)} \sum_{i=1}^m (\tau_i - \overline{\tau}_m)^2$
Standard Error of Time Lag Average	$\tau_{\Delta}^2_{jk} = \tau_{\Delta}^2 [\overline{\tau}] = \frac{1}{2L} \sum_{\ell=-L+1}^{+L} \Delta^2 \overline{\tau}_{jk} (\tau_{\ell})$
Standard Error of Channel Average	$k\tau_{\Delta}^2_j = \frac{1}{N} \left\{ \overline{\Delta}_{jk_1}^2 + \overline{\Delta}_{jk_2}^2 + \overline{\Delta}_{jk_3}^2 + \dots + \overline{\Delta}_{jk_N}^2 \right\}$

The values for the first piece are assumed to be

$$\overline{\overline{(\quad)}}_1 = \overline{(\quad)}_1$$

and

$$\Delta^2 \left[\overline{\overline{(\quad)}}_1 \right] = 0$$

In summary, the purpose of the program is to analyze simultaneously recorded time histories of meteorological data by calculating an average of mean values, RMS values, and covariance functions, which approximate the mathematical model of stationary and infinitely long-time histories to the largest practical amount. Several averaging procedures (Table II) can be used which provide different approximations by assigning the statistical weights or ordinate shifts to each piece listed in Table III. The success of the various averaging procedures is then established by monitoring the associated standard errors (Table IV) as a function of integration time. A rough description of the actual deviations from stationary is provided by the weights and shifts that resulted in the smallest standard errors.

2. GENERAL DESCRIPTION

The basic program structure for MLTCOR is shown in Figure 1 and consists of LINK 1 - the input link, LINK 2 - the correlation link, and LINK 3 - the output link. Each link of the program is relatively independent of the others allowing for separate execution of each of the links. Common information required between links is stored on magnetic tape and read into storage when execution of an individual link is initiated. Since each link is relatively independent, they are executed as overlays with each link occupying the space of the previous one, and in this way providing for more data storage in each link.

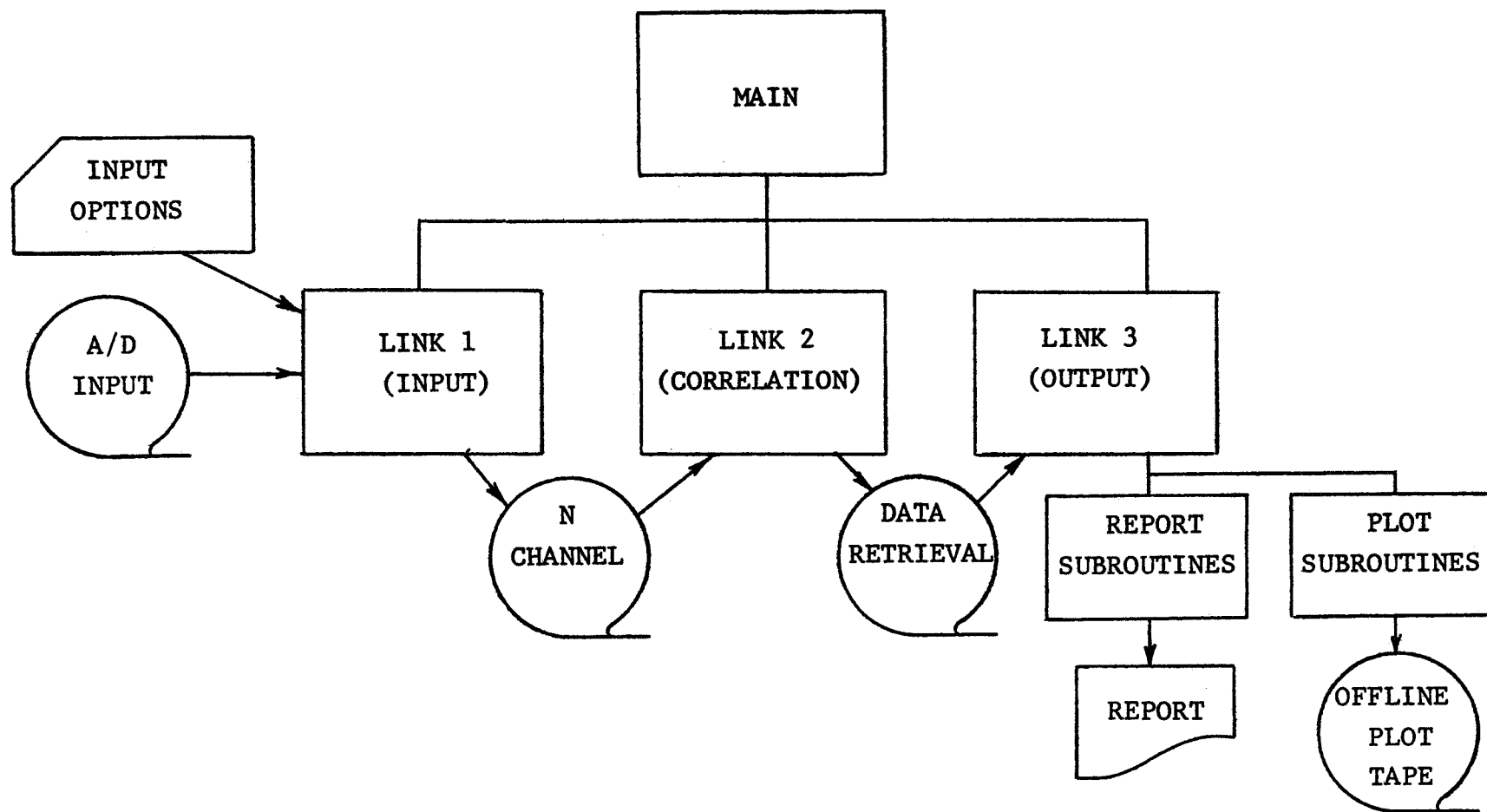


Figure 1 Basic Program Structure

The overlay diagram for the program is shown in Figure 2. The basic functions performed by each link are outlined in the figure and are described in more detail in the later sections.

The main program is a small executive program which controls the execution of the various links and provides for common storage. The flow chart for this program is shown in Figure 3. The main program allows for the execution of each link separately.

LINK 1 performs calibration of the data, sample reduction, and determination of a moving mean. The output from LINK 1 is a block intermediate tape for use in LINK 2. Blocking of the tape is used, i.e., more than one time sample per physical record, since it decreases the length of tape required to store the data and speeds the input/output time for reading and writing the data. The blocking of the intermediate tape in LINK 1 and unblocking in LINK 2 are automatically taken care of by the program.

LINK 2 is the main computational portion of the program and was designed to accommodate multiple auto and cross correlation requests with variable time lags. A special dynamic storage allocation scheme was used to provide maximum storage for data. This scheme avoids the use of fixed length data arrays in the program which would limit its usefulness. The dynamic storage allocation scheme makes it possible to tradeoff between the number of channels to be correlated, maximum time lags, and number of correlations. In fact if the correlations specified require too much storage, the number of correlations requested is automatically reduced until there is sufficient storage to run the problem.

The piecewise concept of analysis was used in creating the program. A long-time series is logically broken up into a series of smaller pieces. Correlations, mean values, RMS values and variances are then calculated for each piece along with their standard statistical errors. Calculations for each piece are accumulated and periodic printouts can be used to monitor piecewise statistics and errors.

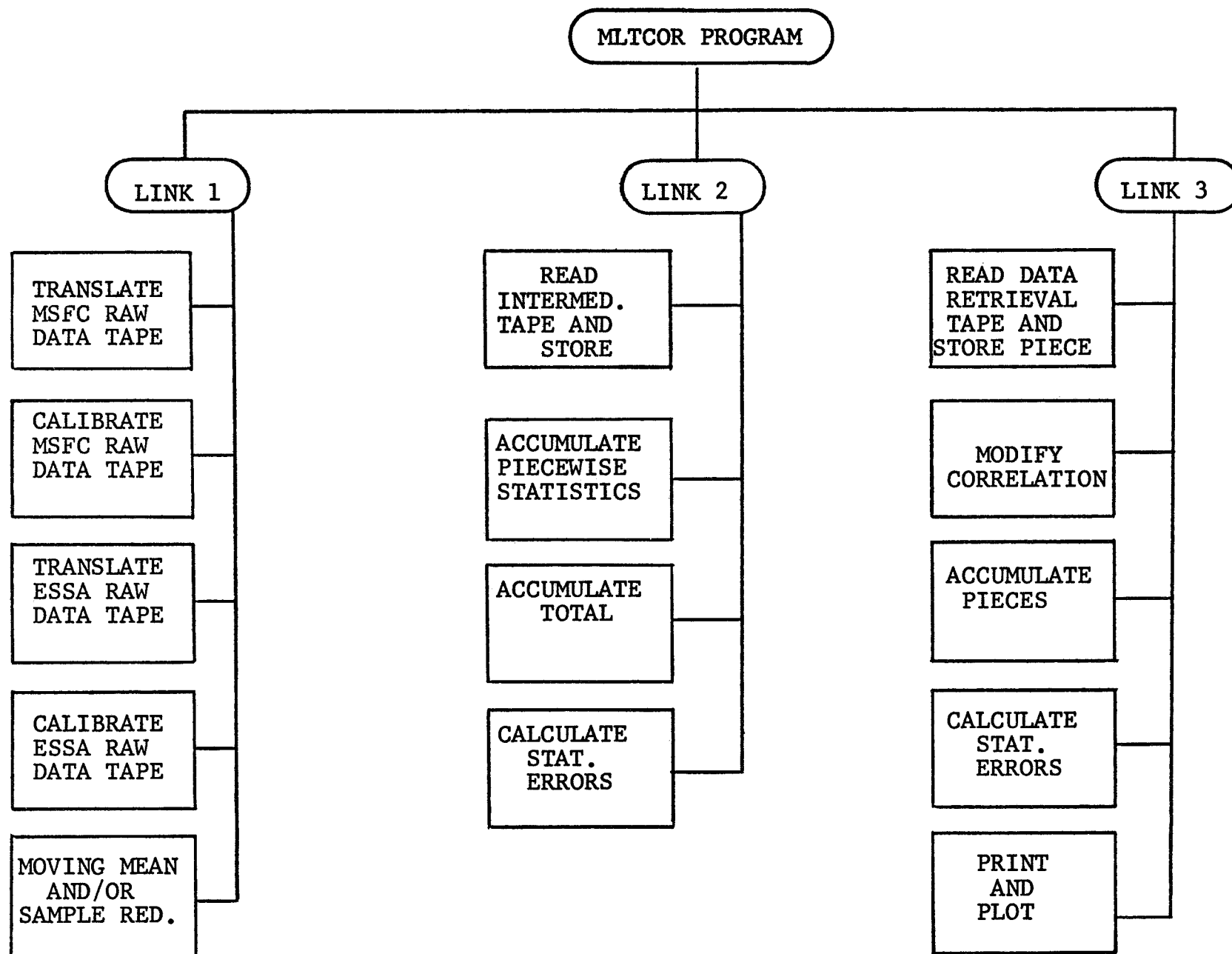


Figure 2 Overlay Diagram

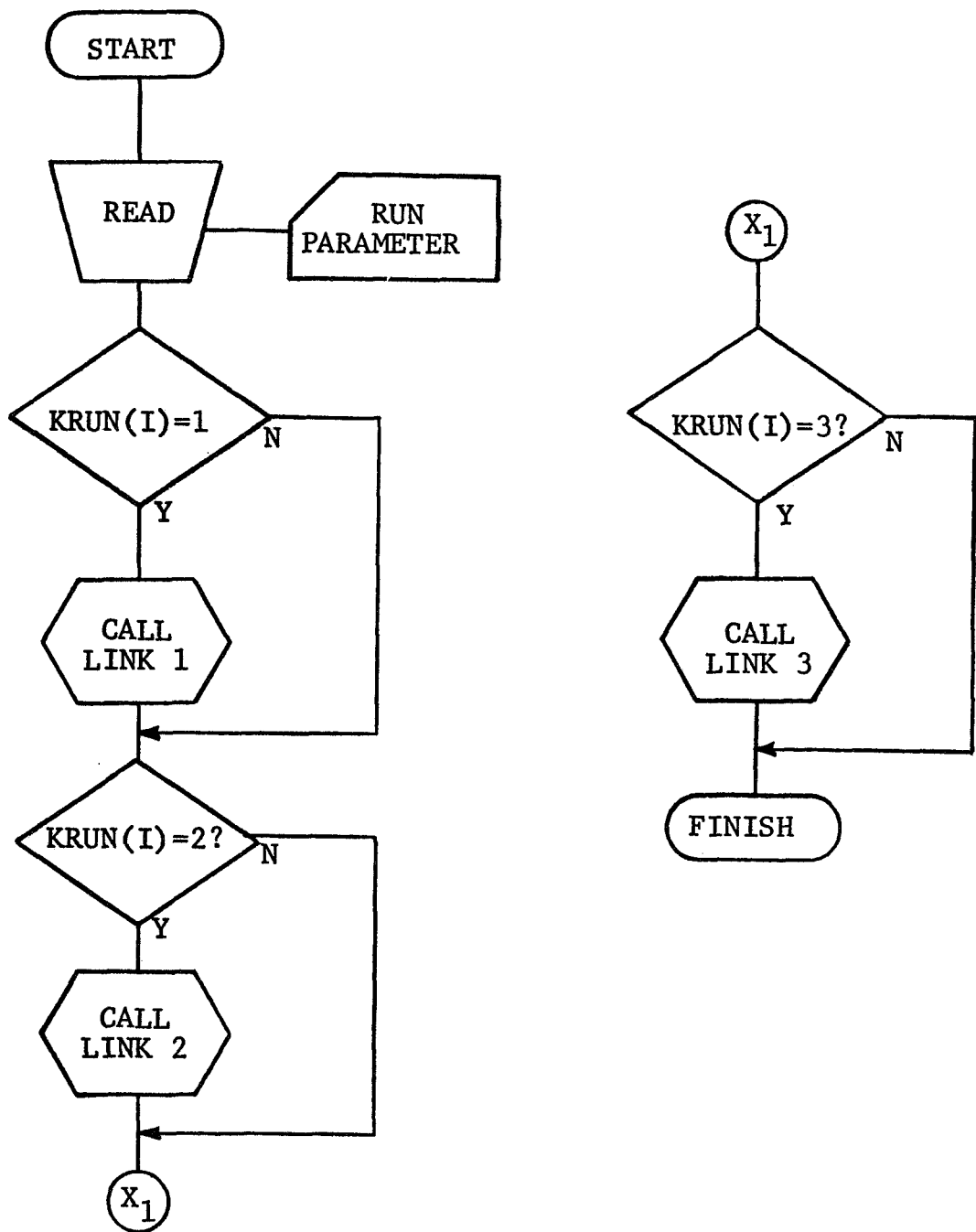


Figure 3 Main Program

Correlations can be performed in LINK 2 on essentially infinite records. Data are continuously brought into the computer in blocks to fill the available storage, and data for the maximum time lag are saved from the previous block to make correlations continuous across core loads of data. Calculating the lagged products for the correlations is the most time consuming portion of the program, therefore, the piecewise correlations and statistics are written out on a magnetic tape called the Information Retrieval Tape (IRT) so that the data may be retrieved in LINK 3.

The correlations formed in LINK 2 are read back in LINK 3 to form modified correlations. At this point a number of standard operations can be performed such as normalizing, detrending, or subtracting the product of the means to form the covariance. Any of these operations can be performed individually or in combination to form modified correlations. The modified piecewise statistics are then summed to form a new accumulation. Printouts or plots of the original or modified correlations can be made for each piece or for the full accumulation. Since only minor calculations are performed in LINK 3, a number of modified correlations can be obtained without rerunning the lagged products which were performed in LINK 2. It is expected that LINK 3 will be run more frequently than the other links to obtain various modified correlations.

3. DYNAMIC STORAGE ALLOCATION

In dealing with large systems on a computer a number of problems arise due to the limitations of internal storage. In many cases these limitations can be overcome through the use of auxiliary memory such as magnetic tape, disc, or drum. A penalty is paid in time for the use of these types of storage, and generally a method of more fully utilizing the high speed core storage of the machine, at the cost of program complexity, is well worthwhile. For many problems of a predictable nature (size

of arrays, flow of processing) preplanned storage allocation is best. When the problem's flow of processing and array sizes are data dependent and variable, dynamic storage allocation is necessary.

Characteristics of dynamic storage allocation are:

- The program at run time is not directly executable, but requires parameters to be set to allocate storage.
- It is not known in advance where in memory data arrays will be held at execute time.

Most operating systems provide good methods of dynamic storage allocation for programs through the use of overlays. Subroutines can be placed on links and called in at execute time when needed as mentioned in the previous section. Unfortunately, FORTRAN IV does not provide the same flexibility for data. For example, an array for storing 1000 values for six channels would require the statement: Dimension X (6,1000). This would reserve 6000 locations of high speed core storage and limit the total number of channels to six and data points to 1000. If a problem required only two channels of data, then 2000 memory locations would be used and 4000 locations would be wasted. Furthermore, it would not be possible to run a problem with eight channels and only 700 data points per channel, even though there is sufficient total storage in the X array. This is due to the column method of array storage used in FORTRAN. A better method of usage is desired which places no arbitrary size restrictions on the problem, and utilizes high speed core memory as efficiently as possible.

The dynamic storage allocation method used in MLTCOR is based on the use of one large array or data pool to store all variable data arrays. This array called U(I) is dimensioned for 20,000 locations. Storage is allocated by assigning pointers which are fixed point numbers to indicate where an array is stored in U(I). The storage pointers are set from data values

read in from cards. As used in LINK 2 of the program, U(1) to U(KESM), is used to store time lag information, and U(KESM) to U(KD) is used to store standard errors, accumulative sums, piece-wise sums and correlations. The remainder of the U array is used to store data points. This portion of the U array is treated as a two-dimensional array with dimensions (NCH,N) where N is the total number of data points and NCH is the number of channels. Both N and NCH can be variable.

The data stored in the U array and space reserved for results are used by variable dimension subroutines. The actual position of the information is communicated to the subroutines through the call statement. As example of a variable dimension subroutine which was actually used in the program is:

```

SUBROUTINE ACCPC(S,S2,S3,DATA,N,NC)

C
C  ACCUMULATE SUM AND SUM OF SQUARES
C    WRITTEN BY A. WACHOWSKI
C  DIMENSION S(NC),S2(NC),DATA(NC,N)
    DIMENSION S3(NC)
    XN=N
    DO 1 L=1,NC
        S(L)=0.0
1    S2(L)=0.0
C
    DO 3 J=1,NC
        DO 2 I=1,N
            S(J)=S(J) +DATA(J,I)
2    S2(J)=S2(J)+DATA(J,I)*DATA(J,I)
            S(J)=S(J)/XN
            S2(J)=S2(J)/XN
            S3(J)=S2(J)-S(J)**2
3    CONTINUE
    RETURN
END

```

This subroutine calculates the average, the average of the squares, and the variance for N data points and NC channels. The arrays S, S2 and S3 are used to store the results for each channel. The actual call statement in the main subroutine of LINK 2 would be: CALL ACCPC(U(KMP), U(KRP), U(KPSM), U(KK), MSUB, NCH) the fixed point numbers indicating the storage location in the U array. This simple example illustrates how the ordinary subscripting capability for calculations is retained with the dynamic storage allocation scheme. The example is quite simple but shows how the other computational modules are written.

4. LINK 1 - THE INPUT LINK

LINK 1 of the MLTCOR program reads raw data, translates, calibrates, conditions, and writes a buffered output tape for subsequent input to LINK 2 for multiple correlations and statistical calculations. The flow chart for LINK 1 is shown in Figures 4 and 5. LINK 1 was programmed using modular techniques, variable dimensioning and dynamic storage allocation. All operations on the raw data will be done in the DATA storage array except the buffered output data. The data storage is dimensioned to 8000 words to store the maximum number of words that might be blocked on either of the two types of input tapes (ESSA or MSFC).

A. Input Tapes

There are two inputs to LINK 1 of the MLTCOR program:

- ESSA format which is the output of the data logging system located at ESSA, Boulder, Colorado (Figure 6).
- MSFC format which is the output of the digitizer located at NASA in Huntsville, Alabama (Figure 7).

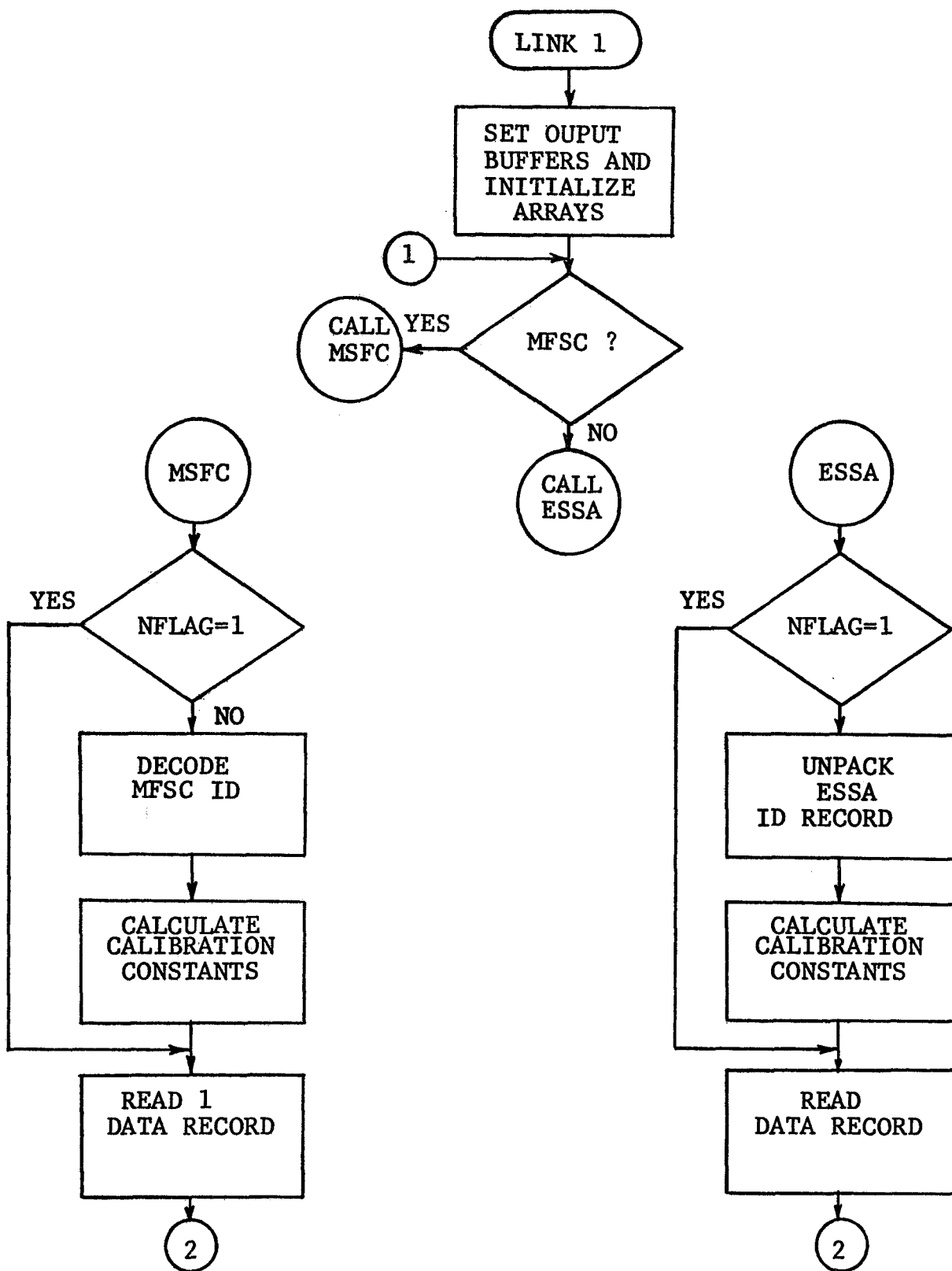


Figure 4 LINK 1-1

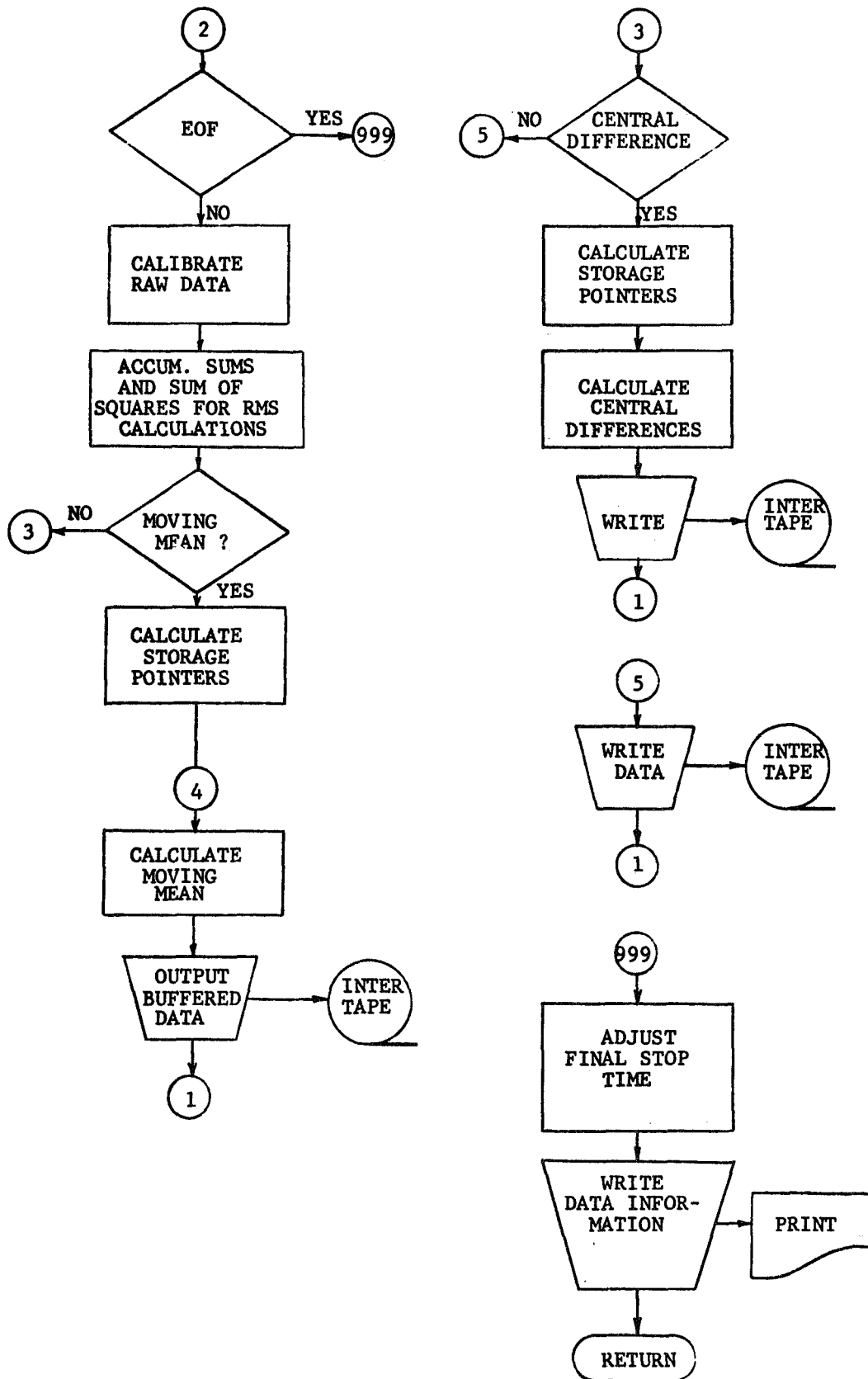


Figure 5 LINK 1-2

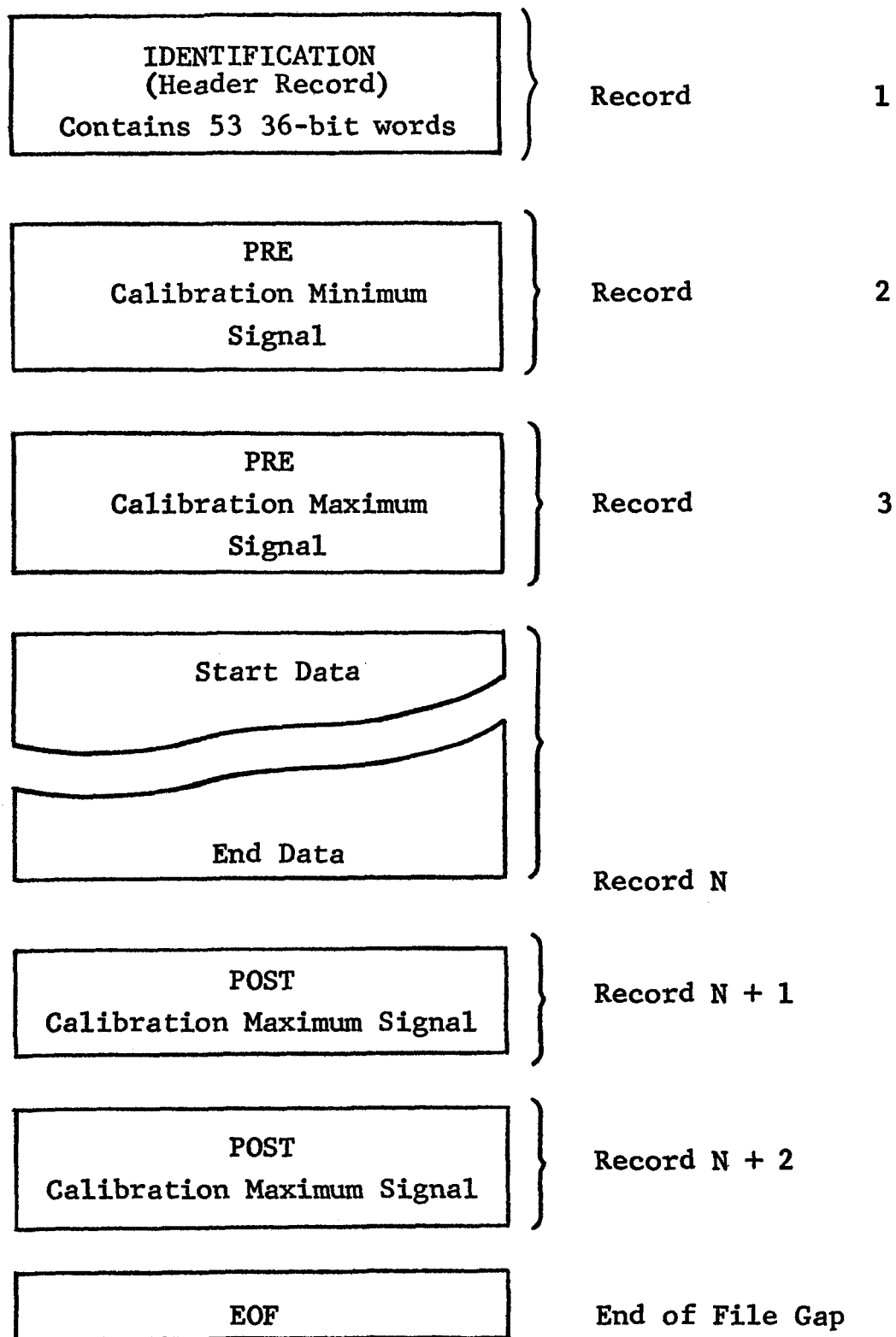


Figure 6 ESSA Data Logger Output Format

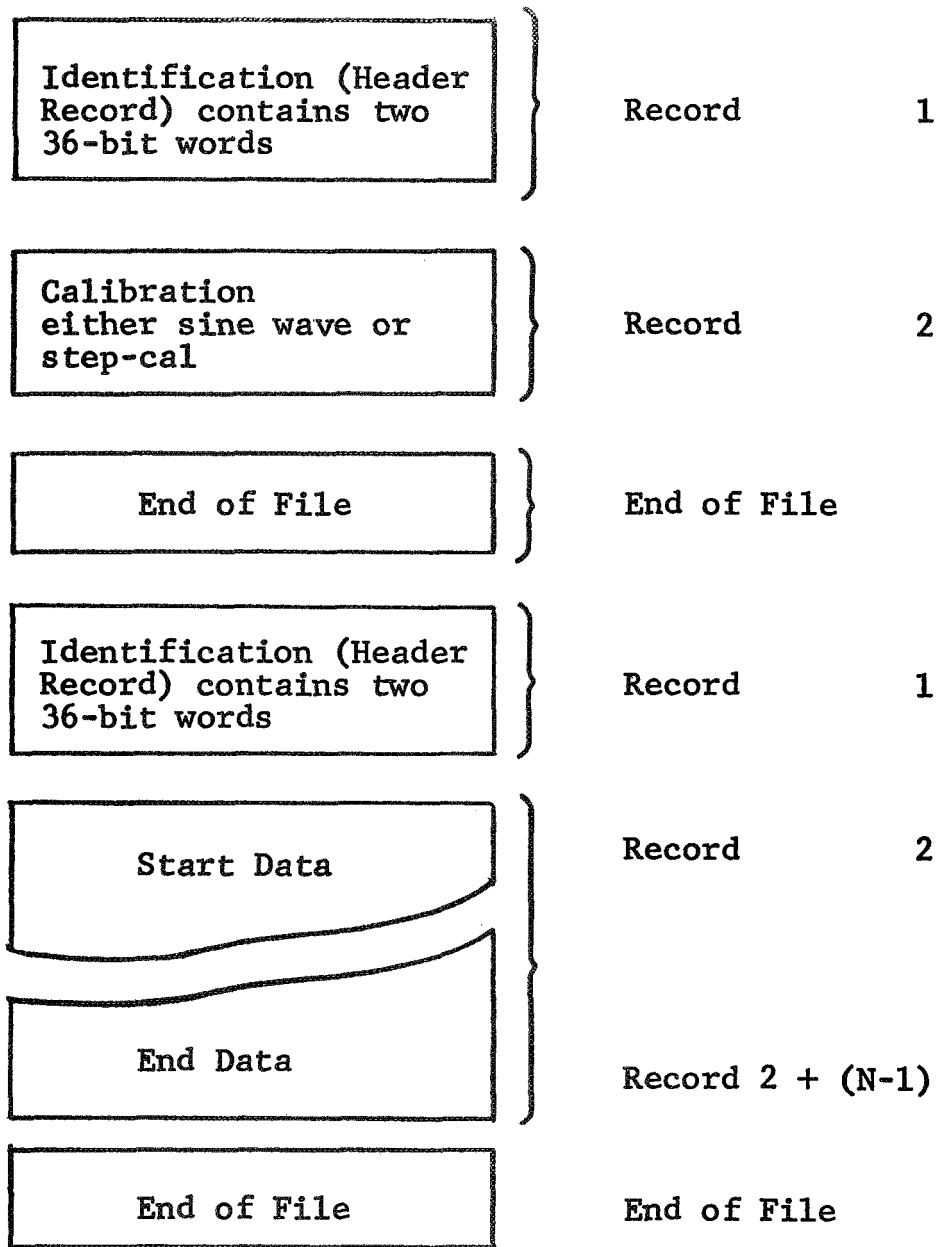


Figure 7 MSFC Digitizer Output Format

B. Input Problem Parameters

The problem parameters that are input to LINK 1 by the user are described here.

NBEAMS	-	Total number of beams in the fan system
NFLSKC	-	Number of files to skip for calibration data
NFLSKD	-	Number of files to skip for data
MVMEAN	-	Number of samples in the moving span if moving mean desired; If not MVMEAN = 0
NCENDF	-	Central differences NCENDF = 1 (Yes) NCENDF = 0 (No)
NOBEAM	-	Total number of beams to perform moving mean or central differences on
NWHICH	-	Beam numbers being affected
TIMSAM	-	Rate at which data are sampled at digitizing time
VALMAX	-	Maximum calibration value
VALMIN	-	Minimum calibration value
START	-	Start time on input tape (Max is 5)
STOP	-	Stop time on input tape (Max is 5)
NFILE	-	Number of files to skip on the ESSA formatted tape for the calibration file
MSFC	--	MSFC format (1 = Yes, 0 = No)
IFREQ	-	LINK 1 output frequency; Example, IFREQ = 3; Every third sample output to tape
LTN	-	Logical tape number input data is loaded on
NBLK	-	Number of records/block desired on LINK 1 output tape
NSTART	-	Number of starts and stops (Max = 5)
NBOB	-	Always set to 0 (used in debugging)
PLYBCK	-	Playback rate (the ratio of the digitizer tape speed to the recorded data tape speed relative to real time)

C. Central Differences

Central differences are calculated from the data by

$$X_{(i)} = X_{(i+2)} - X_{(i)}$$

Where

$i = 1, 2, 3, 4, \dots, N-2$ samples
on the data record.

D. Moving Mean and Sample Reduction

Filtering is performed on the calibrated data by passing the data through a moving mean span preselected by the user, preferably an odd integer. This is performed on the entire data record or selected data intervals controlled by the input parameters (Section 7A). The data are passed through this span successively, one sample at a time. Each data sample, $X_{(K)}$, where

$$X_{(K)} = \frac{1}{N} \sum_{i = K - \frac{N-1}{2}}^{K + \frac{N-1}{2}} X(i)$$

K goes from $\left(\frac{N-1}{2}\right)$ to $T - \left(\frac{N-1}{2}\right)$

where N = number of samples in the moving span, and T = integration time

is calculated and an option is provided to retrieve from the filtered data any sample rate for output to the intermediate tape by incrementing K through the data at the desired reduction rate. This performs the sample reduction.

E. Calibration

1. Sine Wave Calibration. - A calibration signal is put on the analog tape at some specified frequency with a peak-to-peak value of 1.414215 volts. The peak value of the digitized raw data counts is determined by calculating the RMS of the sine wave.

$$RMS = \left[\frac{1}{N} \left(\sum_{i=1}^N X^2(i) \right) - \left(\frac{\sum_{i=1}^N X(i)}{N} \right)^2 \right]^{1/2}$$

where $X(i)$ = Raw data sample in digitizer
counts

and N = number of samples used

The peak values are then determined by

$$\text{Peak}_{\max} = \text{RMS} \times 1.414215$$

and

$$\text{Peak}_{\min} = -\text{RMS} \times 1.414215$$

The calibration constant is then calculated by

$$C = \frac{\text{Cal}_{\max} - \text{Cal}_{\min}}{\text{Peak}_{\max} - \text{Peak}_{\min}}$$

Where

$\text{Cal}_{\max}$ = Maximum engineering unit input by user

$\text{Cal}_{\min}$ = Minimum engineering unit input by user.

The calibrated data for the AC signal then becomes

$$X'(i) = X(i)C$$

An option is provided to calibrate the step-cal, utilizing the sine wave, by making the following calculations

$$\text{Peak}_{\max} = \text{RMS}$$

The calibration constant is then calculated for the step-cal by

$$C = \frac{\text{Cal}_{\max} - 0}{\text{Peak}_{\max} - 0}$$

where $\text{Cal}_{\max}$ = Maximum engineering unit input by user

$\text{Cal}_{\min}$ = Minimum engineering unit input by user.

The calibrated data for the DC signal then becomes

$$X'(i) = X(i)C$$

2. Step Calibration. - A calibration signal is put on the analog tape where the 100 percent signal level is 1 volt and the 0 percent level is 0 volts. This signal is then digitized where minimum raw counts are at the 0 volt level and maximum raw counts are at the 1 volt level. The minimum and maximum values are determined by

$$\bar{X}_{\max} = \frac{1}{N} \sum_{i=1}^N X(i)_{\max}$$

and,

$$\bar{X}_{\min} = \frac{1}{N} \sum_{i=1}^N X(i)_{\min}$$

and the calibration constant is then calculated by

$$C = \frac{\text{Cal}_{\max} - \text{Cal}_{\min}}{\bar{X}_{\max} - \bar{X}_{\min}}$$

In the step calibrations for the DC signal where the slope intercept is not zero, a correction is applied by calculating a bias.

$$\text{BIAS} = -\bar{X}_{\max} + \text{Cal}_{\max} C$$

The calibrated data for the DC signal becomes

$$X'(i) = X(i)C + \text{BIAS}.$$

F. Essa Data Logger Output

The ESSA digital output tape format consists of one or more files. Each file constitutes data for one test which is identified by a header record. The format of the header record is shown in Figure 8.

1. Tape Number	6 Characters
2. Organization	6 Characters
3. File count	6 Characters
4. Date	3 Digits for day of year 2 Digits for year 1 Digit spare
5. Time (Start)	Time in sec 6 Characters
6. Time (End)	Time in sec 6 Characters (Greenwich Time)
7. Name of variable	6 Character names 72 Characters total
12 Channels	
8. Calibration value minimum	Characters/Channel 12 Characters total
12 Channels	
9. Calibration value maximum	Characters/Channel 12 Characters total
12 Channels	
10. Sampling rate	6 Characters sample/sec
11 Sample reduction	6 Characters
12. Anemometer information	6 Characters 54 Characters total

Figure 8 ESSA Header Record

Individual data records consist of not more than 13 words of 12 bits each. The first word is a record count or time word; this is followed by 12 words - one for each channel. This record constitutes one sample time. The individual data records are grouped together in a block of some 30 records and are written as a single physical record on magnetic tape.

The first and last physical record contains 30 calibration records with minimum and maximum levels recorded.

G. MSFC Digitizer Output

The MSFC digital output tape format consists of calibration and data files. The calibration file contains an identification record and one or more data records with the calibration signal. The identification record is two 36-bit words (Figure 9). The calibration data record consists of a 24-bit time word and the remainder of the record is filled with 11-bit digitizer words not to exceed 8000 11-bit words. The data file is formatted the same.

1. Time Scale	BCI*	24 bits
2. Data Type	BCI	6 bits
3. Number of Scans	BCI	24 bits
4. Number of Beams	BCI	12 bits

*BCI - Binary Coded Information

Figure 9 MSFC HEADER RECORD

5. LINK 2 - THE CORRELATION LINK

Flow charts for LINK 2 are shown in Figures 10 and 11. These are first level flow charts which present the general flow of control but not the detailed processing. Referring to Figure 10 the subroutine begins by restoring COMMON from the intermediate tape produced by LINK 1. The problem parameters are then read

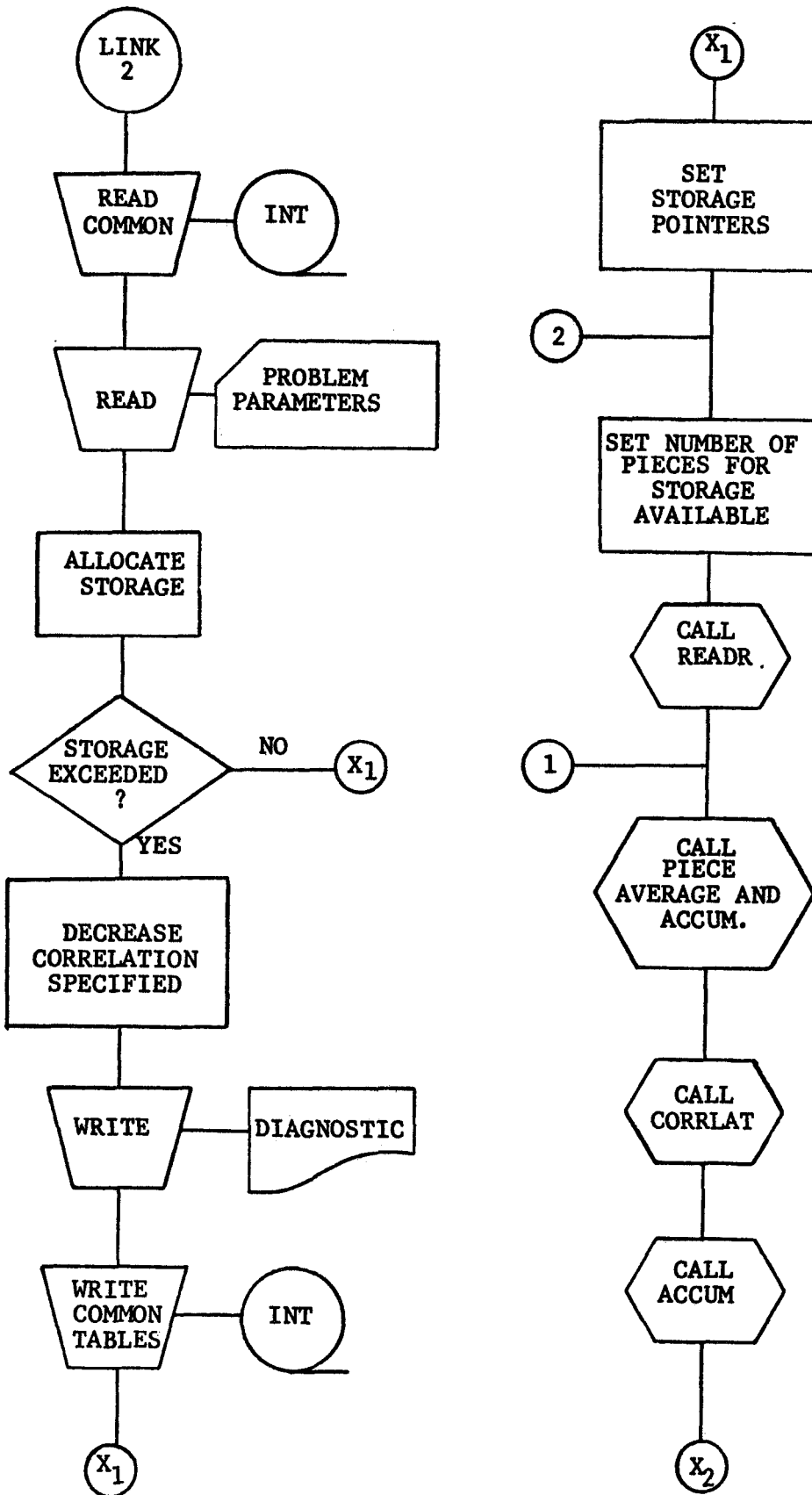


Figure 10 LINK 2-1

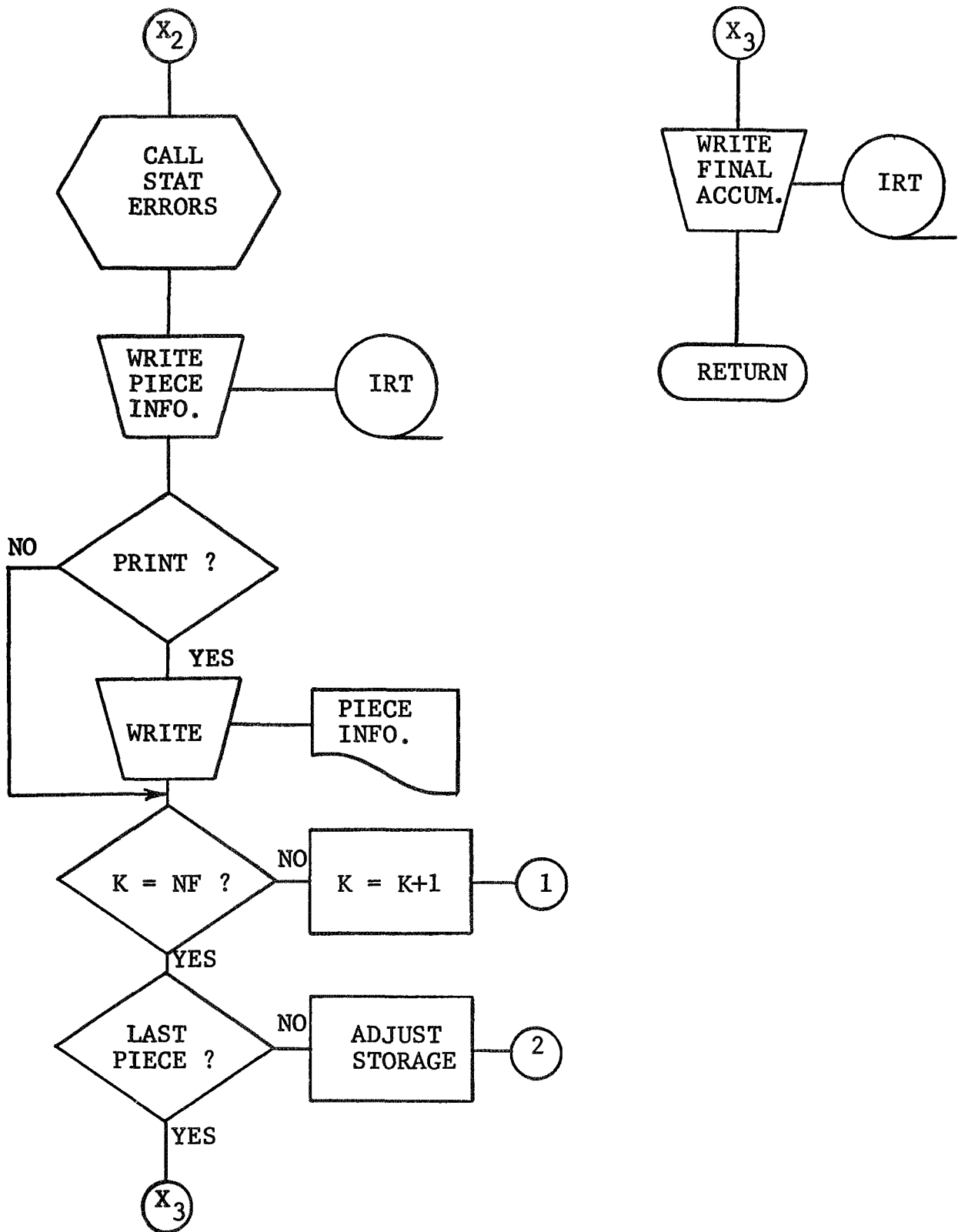


Figure 11 LINK 2-2

specifying piece length, print interval, time-lag range and increment, and finally the correlations to be performed.

A large degree of flexibility has been built into the program for the user in specifying correlations. To accommodate varying input tape formats and provide for efficient storage and processing, the correlation requests will be analyzed for the total number of channels required and a new internal channel number will be assigned for use within the program. Thus, a typical request may appear as:

J = 2, K = 5,6, 4 * 0

J = 3, K = 8,4,2,3,*0

where J is the delayed channel number on the input tape and the K's are the undelayed channels. The input tape may contain up to 12 channels but the correlation requests above require only six. The unused channels will not be stored during the computations and a new channel number will be assigned in the program as follows:

New	Old
1	2
2	3
3	4
4	5
5	6
6	8

The internal identification number simplifies handling in the program and will never be of concern to the user since all output is in terms of the original channel assignments on the input tape.

The typical correlation request is shown in Figure 12. A series of tables are built up in LINK 2 which are used to simplify processing. The first table NCHID (Figure 13) is formed from the J's and K's specified in the correlation requests. NCHID consists of all channels to be used, delayed or undelayed, in calculating the correlations. The channel numbers correspond to the position on the input data tape. The table NCRPC gives the number of

$J = 2, K = 5, 6; 4*0$

$J = 3, K = 8, 4, 2, 3*0$

Figure 12 Correlation Request

NCHID	NCRPC	NCTBL
1 2	1	1
2 5	2 2	2 56
3 6	3 3	3 842
4 3	4	4
5 8	5	5
6 4	6	6

SORT				
NCHID	NCRPC	NCTBL	NCD	NACR
1 2	1 2	1 45	1 1	1 0
2 3	2 3	2 136	2 2	2 2
3 4	3	3	NCRT-3 5	
4 5	4	4	4	
5 6	5	5	5	
6 8	6	6	6	

Figure 13 Tables

correlations requested for a particular delayed channel and the table NCTBL gives the channel numbers of the delayed channels for a given delayed channel.

The table NCHID is sorted to put all channels in the order of increasing channel numbers and the tables NCRPC and NCTBL are constructed with the new channel numbers, which are assigned from the order of NCHID. Thus, for the example, the original channel 2 becomes channel 1, channel 3 becomes channel 2, etc. The total number of channels to be used is NCH.

After the channel numbers have been reassigned two new tables are formed called NCD and NACR. The table NCD consists of the total number of delayed channels and the new channel numbers. The table NACR is an accumulative count of the correlations requested.

The first location of this table is set to zero. Location 2 is set to the number of correlations for channel 1, location 3 is the total number of correlations for channel 1 + 2, etc. In general, the last location in the table indicates the total number of correlations, NCRT. For the above example, NCRT = 5. The table NACR is used to locate information on the correlations in the data pool or U array.

Figure 14 shows the total allocation of storage in the working array U. The array is broken up into 5 basic parts. The first part consists of miscellaneous constants, which is a series of time lags from $-\tau_{\max}$ to $\tau_{\max}$ and velocity lags. The second part is for storage of the standard errors of σ^2 , $\bar{X}$, and $\bar{X}^2$. The third part of the array U is devoted to the accumulations $\overline{\sigma^2}$, $\overline{\bar{X}}$, $\overline{\bar{X}^2}$. The fourth portion of U is used to store the piecewise statistics of σ^2 , $\bar{X}$, $\bar{X}^2$ and the correlations $\overline{XY}$ (R_{JK}) as well as the time lag average of the correlations ($\tau \overline{XY}$). The remainder of the U array from U(KP) is used to store sample points for the channels to be used. It should be noted that all the above regions are variable in length and particularly the correlations (R_{JK})

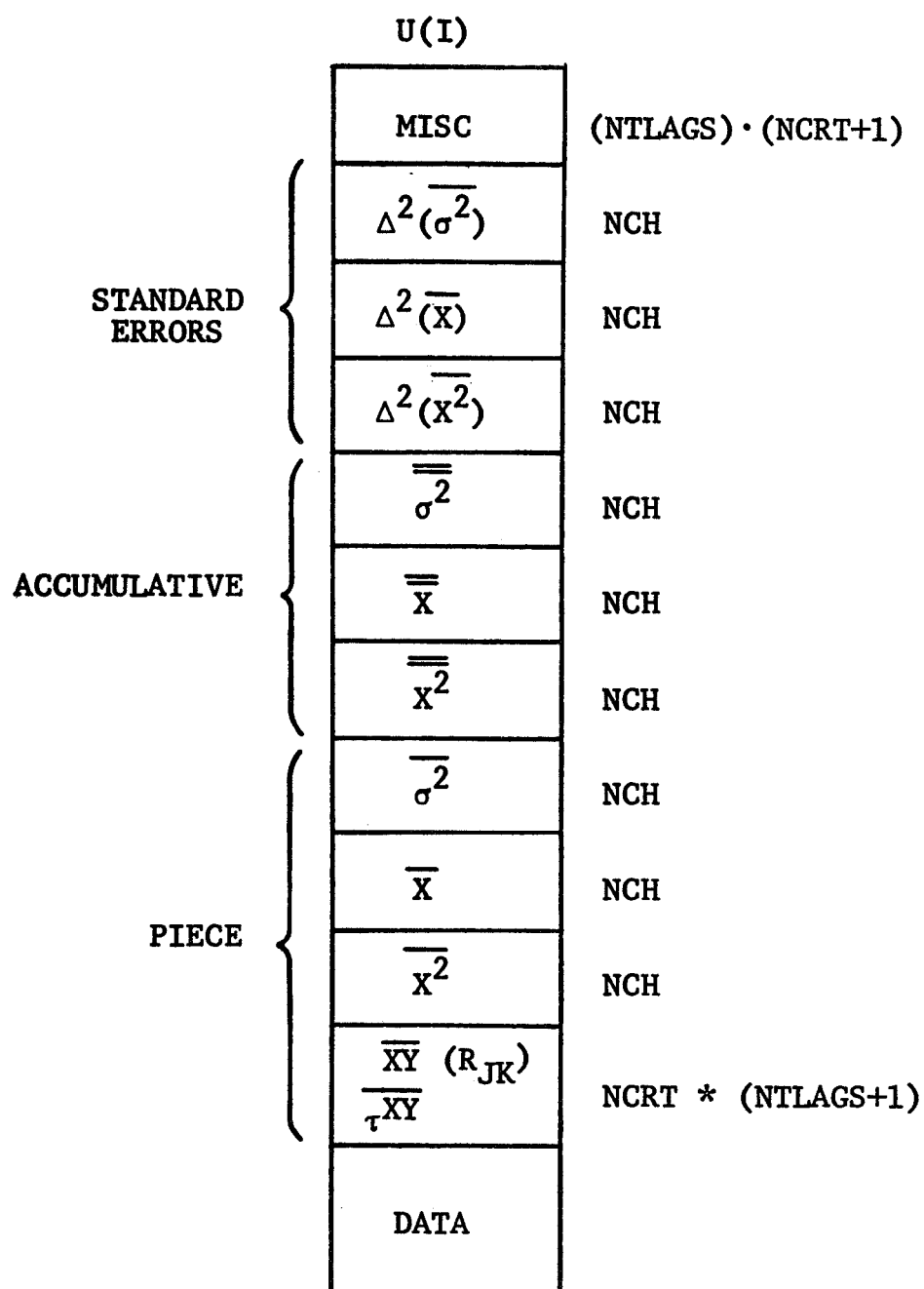


Figure 14 LINK 2 Storage Allocation

require the largest amount of storage $NCRT * (NTLAGS + 1)$. $NTLAGS$ is the total number of positive and negative lags for the correlations. The position pointer names and required space is shown in Table V.

Total storage for the results is equal to:

$$KP = (NCRT + 1) (NTLAGS) + 9 \cdot NCH + NCRT \cdot (NTLAGS + 1)$$

Storage for data samples will be 20,000 - KP. Thus the storage in LINK 2 is controlled by the correlation request. The fewer the number of correlations requested the more storage can be allocated to data samples.

Referring to the flow chart for LINK 2 (Figures 10 and 11) the total storage requirement to run a problem is checked. This is given by the relationship;

$$\begin{aligned} & (NCRT + 1) (NTLAGS) + 9 * NCH \\ & + NCRT * (NTLAGS + 1) \\ & + (MSUB + MAXINC) * NCH < 20000 \end{aligned}$$

where $MSUB = \frac{\Delta T}{\bar{e}}$ and

$$MAXINC = \frac{\tau_{\max}}{\bar{e}}$$

which provides sufficient storage for at least one piece. If the storage is exceeded, then the correlation request is automatically reduced until there is sufficient room to run the problem. This is accomplished in the program simply by reducing the counter ND which is the total number of delayed channels. ND is used in determining NCRT from table NACR (Figure 13) and is also used in controlling the computation of the correlation. Consequently, simply reducing ND reduces the storage requirements and the computations performed. If the storage requirements are exceeded then the specifications are reduced and a diagnostic is printed. Reduction of ND to less than one will terminate the run. If there is sufficient storage to run the problem then the COMMON area and the

TABLE V
PIECEWISE STATISTICS

POSITION POINTER	VARIABLE	DIMENSION
1	LAG VALUES (Time, Velocity)	IF VELLAG $\neq$ 0 (NTLAGS) $\cdot$ (NCRT+1) IF VELLAG = 0 NTLAGS
KESM	$\Delta^2(\overline{\sigma^2})$	NCH
KEAM	$\Delta^2(\overline{X})$	NCH
KEASM	$\Delta^2(\overline{X^2})$	NCH
KASM	$\overline{\sigma^2}$	NCH
KMA	$\overline{X}$	NCH
KRA	$\overline{X^2}$	NCH
KPSM	$\overline{\sigma^2}$	NCH
KMP	$\overline{X}$	NCH
KRP	$\overline{X^2}$	NCH
KTP	$\overline{XY}$	NCRT*(NTLAGS+1)
	$\tau \overline{XY}$	

various tables formed from the correlation requests are written out onto the IRT which is used in LINK 3, and the storage pointers are set as shown in Table V.

Connector 2 (Figure 10) begins the main part of the program. A read subroutine is called to fill the working array U, starting at KD, from the intermediate tape from LINK 1. If all channels are not used by the program during a run, the data will not be stored by the subroutine. Thus, if only two channels are to be correlated, all storage will be used to store the corresponding X and Y data points. If more than one undelayed channel is used, then X, Y_1 , Y_2 , etc., would be stored. After the working storage is filled, the number of pieces in core storage is set and the actual calculations begin at connector 1. The calculation modules are written as variable dimension subroutines. The CALL statement to these subroutines pass the actual storage location of data and results stored in U. Only the piecewise correlations are performed in this link and written onto the IRT. Statistical errors are calculated for only nontime dependent averages. Periodic printouts can be controlled by the variable IPRINT set by the user, which gives the frequency of printout.

In general, more than one piece will be held in core storage at one time. This depends, of course, on the number of correlations requested and the total number of channels required but generally the data portion of the U array will appear as in Figure 15a. MSUB is the number of data points for the piece length ΔT . N_ℓ is the number of data points required for the maximum time lag, $\tau_{\max}$. Thus, data storage will appear as in Figure 15a at the beginning of LINK 2. There will be a number of pieces of length MSUB and enough data points, N_ℓ , to complete the correlation calculations for the last piece. When the correlation for the last piece has been performed then the N_ℓ data points are moved, as shown in Figure 15b, to the beginning of the data area and sufficient data is read in to fill the entire data area again.

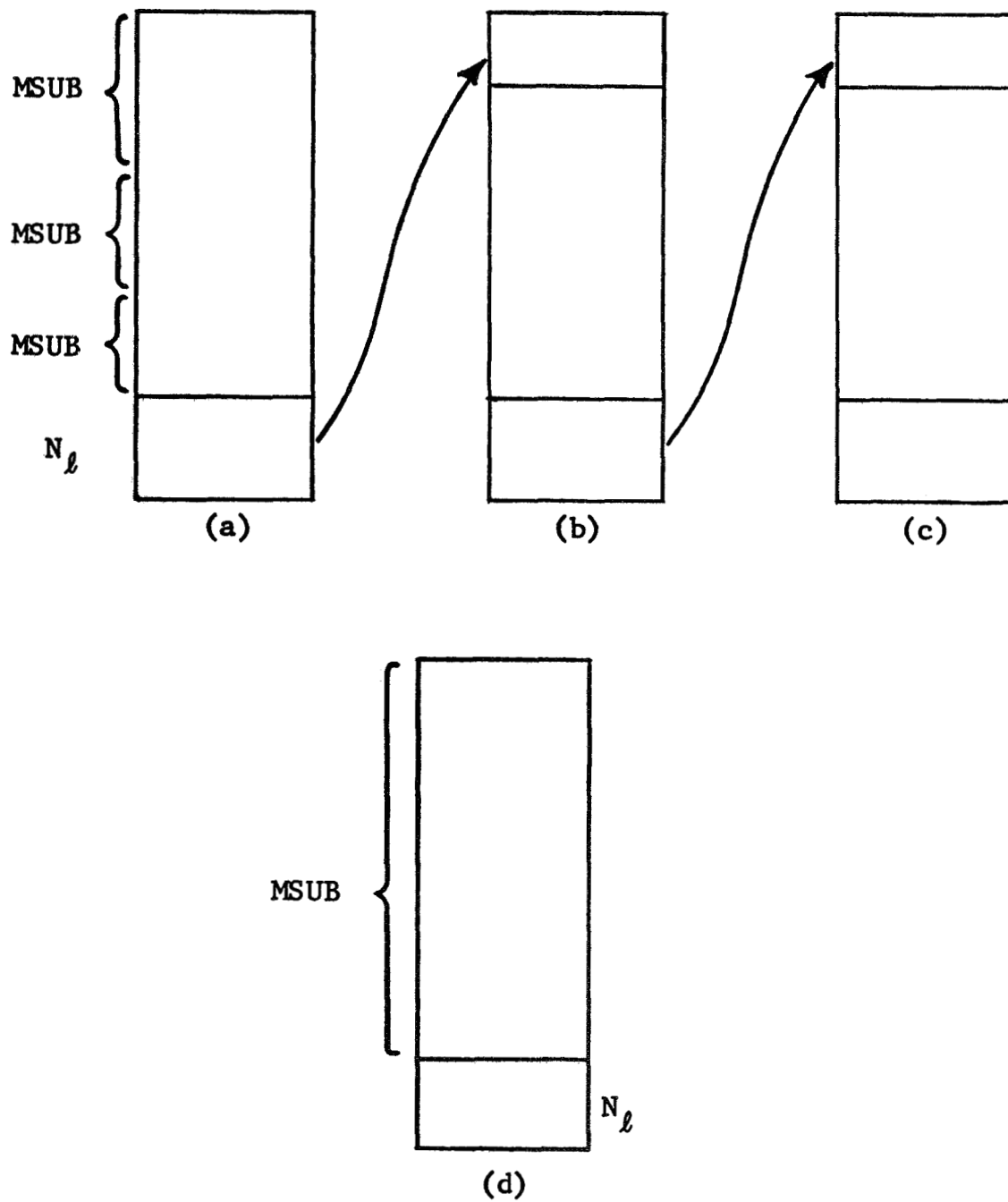


Figure 15 Storage Use

The averaging operations and correlations are performed again. This procedure is continued for successive core loads as shown in Figure 15c. By this method, of operation correlations are kept continuous across core loads of data since the data for the maximum time lag is saved from the previous block. In general, as shown in Figure 15d, the channel averages for a piece represents MSUB data points while the correlations are over $MSUB + N_{\ell}$ data points. As long as there is sufficient room to store one piece, $MSUB + N_{\ell}$, the correlations can be performed without the loss of any information.

6. LINK 3 - THE OUTPUT LINK

Link 3 of the MLTCOR program performs a number of functions other than output. It is in this link of the program that the basic correlations formed in LINK 2 are modified to aid in determining nonstationary trends. Standard operations can be performed on the correlations such as normalizing, (dividing by the product of the standard deviations for each channel) detrending, (which consists of subtracting the time lag average from the correlations) and the covariance (formed by subtracting the product of the means of the two channels from the correlations). Any combination of these standard operations can be performed on the correlations read in from the IRT.

The flow chart for LINK 3 is shown in Figures 16 and 17. LINK 3 can be run a number of times with the same output from LINK 2. The first operation performed by the program is reading all COMMON variables from the IRT and the correlation control tables which were set in LINK 2. The user is then allowed to read in from data cards the piece numbers of pieces that are to be deleted in forming the modified accumulations. The options for forming the modified correlations are read on another data card. The accumulation period, $t_{q+1} - t_q$ are next set from data cards. The format for these data cards is described in Section 7.

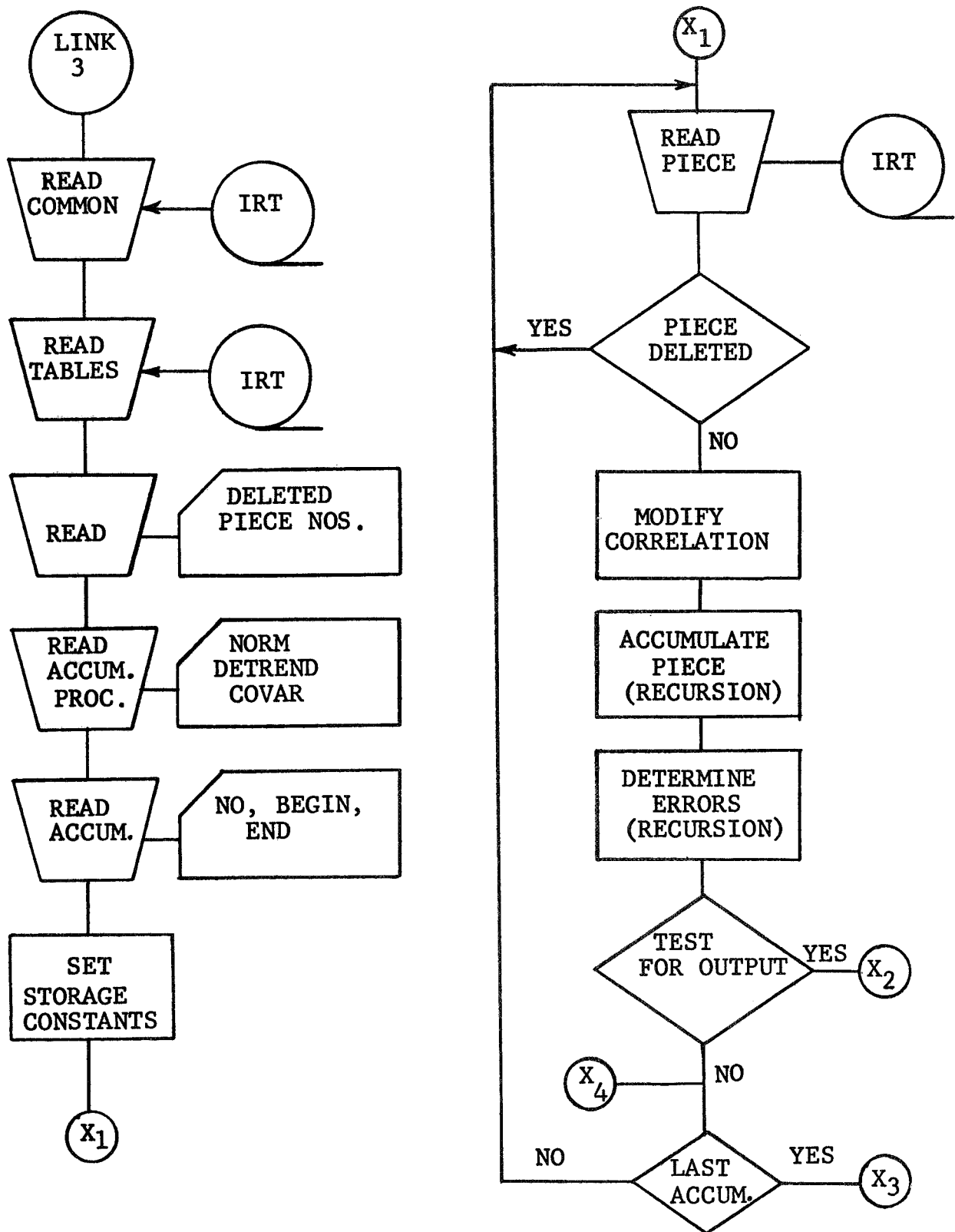


Figure 16 LINK 3-1

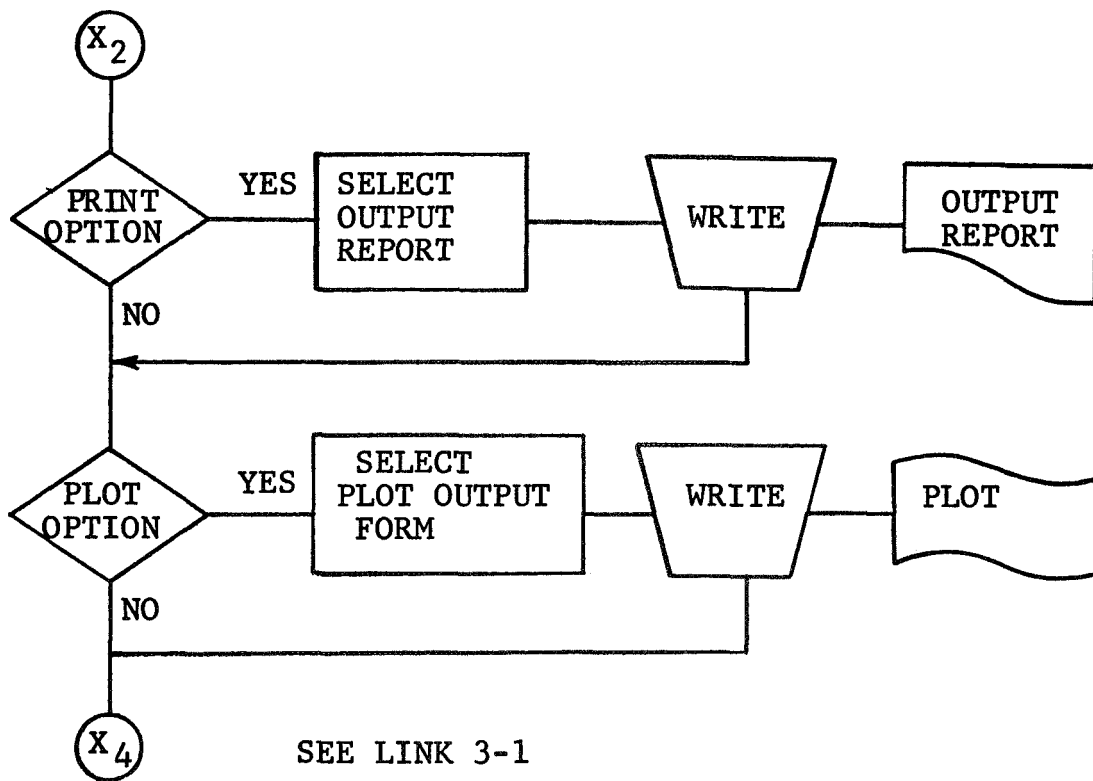
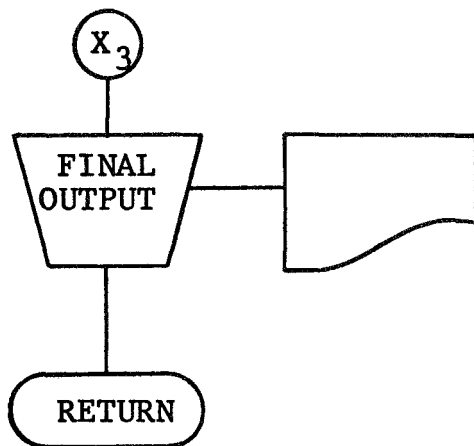


Figure 17 LINK 3-2

With the information supplied above it is possible to set constants for the allocation of core storage. The same dynamic storage allocation scheme that was used in LINK 2 is used in LINK 3 except that a large block of storage is not required for the raw data. In fact it is only necessary to store one set of piecewise correlations at a time. The remainder of storage is used for modified correlations and error terms. The allocation of storage for LINK 3 is shown in Table VI. Column 1 of Table VI gives the name of the fixed point variable used as the storage pointer for the U array. Column 2 defines the quantities calculated with the dimension shown in column 3. The piecewise statistics calculated in LINK 2 are also stored in LINK 3. The same storage pointer names used in LINK 2, shown in Table V, are used in LINK 3.

In LINK 2 the accumulative correlations, time lag averages and the corresponding standard errors were not calculated since in general these quantities will vary depending on the deleted pieces. These quantities are calculated in LINK 3 for the unmodified correlations and stored at KATP and KEXP as shown in Figure 18, which shows the total storage allocation for LINK 3.

Referring to Figure 16 and 17 the main loop of LINK 3 begins at connector X1. The piecewise correlations are read from the IRT and stored as shown in Figure 18. The piece number is checked to see if it is to be deleted. If it is deleted another piece is read and checked. If the piece is not deleted, then the correlations are accumulated and then modified. The modifications that can be made are given by

$$c'(\tau) = \left(\frac{\left[\overline{x_j y_k(\tau)} - \bar{x}_j \cdot \bar{y}_k \right] - \left[\overline{\tau^x j y_k} - \bar{x}_j \cdot \bar{y}_k \right]}{\sigma_j \cdot \sigma_k} \right) \tau^*$$

depending on the following logical variables.

TABLE VI
MODIFIED CORRELATIONS

POSITION POINTER	VARIABLE	DIMENSION
1	Number of pieces selected	NTP
LTLAG	Time lags	NTLAGS
LVLAG	Velocity lags	NCRT * NTLAGS (if velocity lags requested)
ICHA	$\overline{\Delta^2} = \overline{\Delta_{JK}^2}$	NCD
IESM	$\Delta^2(\overline{\sigma^2})$	NCH
IEAM	$\Delta^2(\overline{X})$	NCH
IEASM	$\Delta^2(\overline{X^2})$	NCH
IEXP	$\Delta^2(\overline{XY})_m$ $\tau \Delta^2(\overline{XY})_m$ }	NCRT*(NTLAGS+1)
LASM	$\overline{\sigma^2}$	NCH
LMA	$\overline{X}$	NCH
LRA	$\overline{X^2}$	NCH
LXA	$\overline{XY}_m$ $\tau \overline{XY}_m$ }	NCRT*(NTLAGS+1)
MPSM	$\overline{\sigma^2}$	NCH
MMP	$\overline{X}$	NCH
MRP	$\overline{X^2}$	NCH
MTP	$\overline{XY}_m$ $\tau \overline{XY}_m$ }	NCRT*(NTLAGS+1)
KEXP	$\Delta^2(\overline{XY})$ $\tau \Delta^2(\overline{XY})$ }	NCRT*(NTLAGS+1)
KATP	$\overline{XY}$ $\tau \overline{XY}$ }	NCRT*(NTLAGS+1)

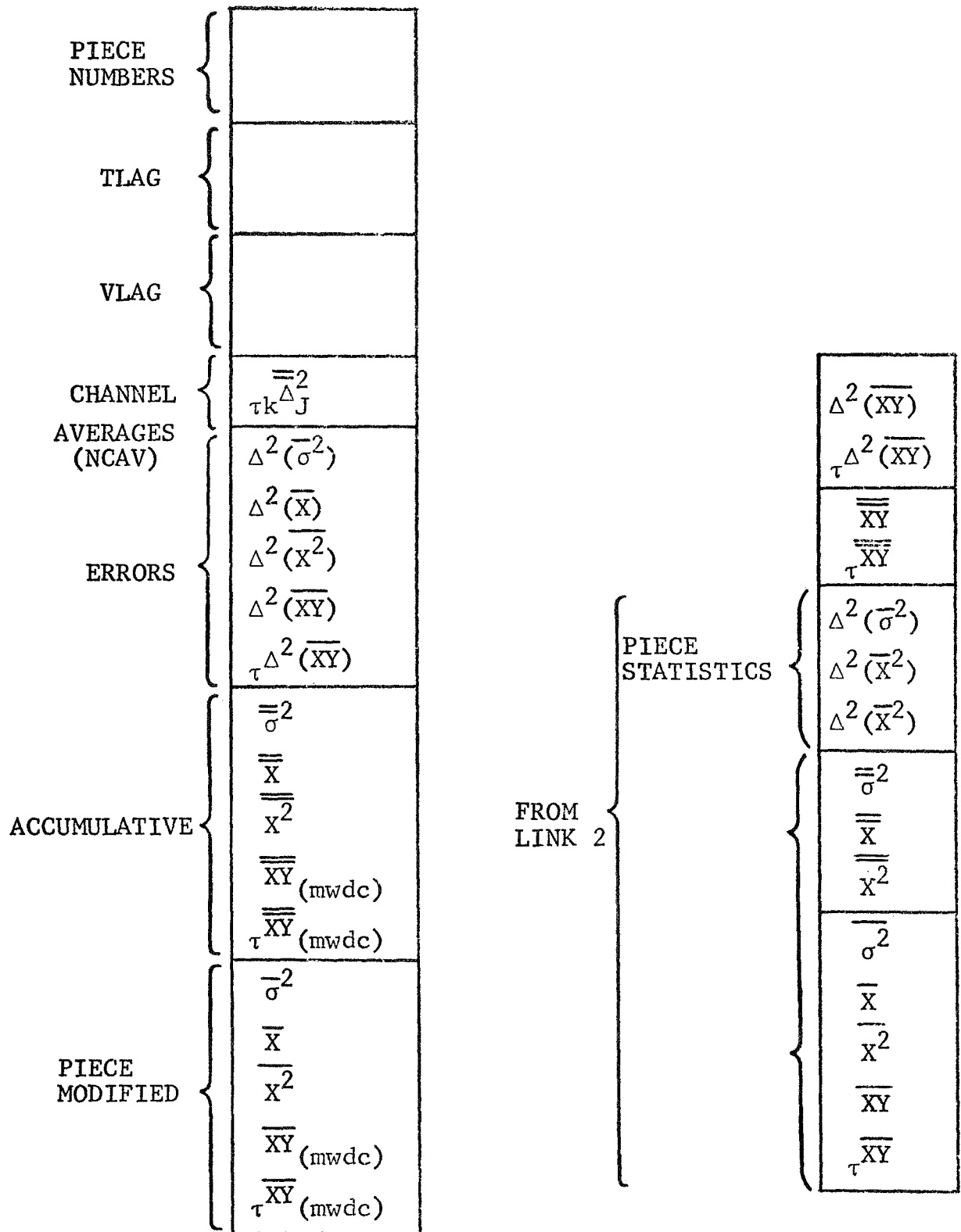


Figure 18 LINK 3 Storage Allocation

```

COVAR = .FALSE. .  $\bar{x}_j$  .  $\bar{y}_k$  = 0
NORM   = .FALSE. .  $\sigma_j$            = 1.
                 $\sigma_k$            = 1.
DTREND= .FALSE. .  $\overline{\tau^{x_j \cdot y_k}} - \bar{x}_j \cdot \bar{y}_k$  = 0
PREMLT= .FALSE. .  $\tau^* = 1.$ 

```

If these variables are true, then the quantities take on their respective values.

These modified correlations are then used to determine the standard errors. The final part of the program is a test for the different types of output and for the final accumulation. Up to five accumulation periods may be specified by the user and after each segment the accumulation regions must be cleared to start a new accumulation. The output from LINK 3 is discussed in Section 7B.

7. USAGE

A. Input

Almost all the card input to the MLTCOR is provided by use of the NAMELIST statement except for one alphanumeric title card. The NAMELIST statement is described in the FORTRAN IV manual, but in general it essentially allows the user free form input of parameters. The general form of a card group would be:

```

COL. 1
$NAME
X = 1.0, y = 2.0, ... etc.
$END

```

The description which follows is taken from the IBM FORTRAN IV manual, Form C28-6390-2, and explains the general form that data items may take.

1. Variable name = constant
where variable name may be an array element name or a simple variable name. Subscripts must be integer constants.
2. Array name = set of constants (separated by commas)
where k*constant may be included to represent k constants (k must be an unsigned integer). The number of constants must be equal to the number of elements in the array.
3. Subscripted variable = set of constants (separated by commas)
where k*constant may be included to represent k constants (k must be an unsigned integer). A data item of this form results in the set of constants being placed in consecutive array elements, starting with the element designated by the subscripted variable.

Constants used in the data items may take any of the following forms:

- a. integers
- b. real numbers
- c. double-precision numbers
- d. complex numbers, which must be written in the usual form, (C1, C2), where C1 and C2 are real numbers
- e. logical constants, which must be written as T or .TRUE., and F or .FALSE.

Logical and complex constants may be associated only with logical and complex variables, respectively. The various input variable and options are explained here.

<u>NAMelist</u> <u>NAME</u>	<u>DATA</u>	<u>SUBROUTINE</u>
\$LINKNØ		MAIN
KRUN	Restart constant	
	Example:	
	KRUN = 1,2,3 runs all three links in sequence.	
	KRUN = 2 will run only LINK 2.	
\$NAMDAT		
NBEAMS	Number of beams or sensors	
	Example:	
	NBEAMS= 7 indicates there are seven sensors or seven beams on input tape.	
NFLSKC	Number of files to be skipped before reading MSFC calibration records	
	Example:	
	NFLSKC = 3 indicates calibration data are in file 4.	
NFLSKD	Number of files to be skipped before reading MSFC data records	
	Example:	
	NFLSKD = 2 indicates data are in file 3.	
MVMEAN	Moving mean span desired for filtering data (preferably an odd integer)	
	Example:	
	MVMEAN = 11; the filter span to be used will be 11 points; if MVMEAN = 0, this option will be bypassed.	

NCENDF Central differences
 Example:
 NCENDF = 1; central differences are calculated; if
 NCENDF = 0 this option will be bypassed.

NOBEAM The total number of beams on which to be operated
 Example:
 NOBEAM = 3 indicates three beams of NBEAMS on
 which to be operated; if MVMEAN = 0
 this option is ignored.

NWHICH Which beams of the NBEAMS on which to be operated
 Example:
 NWHICH = 1,3,4 indicates beams 1,3, and 4 only;
 if MVMEAN = 0 this option is ignored.

TIMSAM Sample rate of input data
 Example:
 TIMSAM = 0.1 indicates data were sampled at ten
 samples per sec.

VALMAX Maximum value for calibrations
 Example:
 VALMAX = 1.0, 1.0, 1.0 indicates three channels are
 calibrated to maximum value of 1.0.

VALMIN Minimum value for calibrations
 Example:
 VALMIN = -1.0, -1.0, -1.0

START Start time on raw data input tape
 (there can be up to five start times)
 Example:
 START = 1.0, 1.5, 2.0, 2.5, 6.0 indicates five
 starting points on the input tape.

STOP Stop time on raw data input tape
 (There can be up to five stop times)
 Example:
 STOP = 1.4, 2.0, 2.4, 5.9, 6.5 indicates the five stop
 times for each respective start time in the
 example above.

NFILE Number of files to be skipped before reading ESSA data
Example:
NFILE = 3 indicates ESSA data are in file 4.

MSFC Flag to the program indicating MSFC formatted tape
Example:
MSFC = 1 will translate MSFC formatted tapes; if
MSFC = 0, program will translate ESSA formatted tapes.

IFREQ Desired output sample rate to intermediate tape
Example:
IFREQ = 1; every sample will be output to the intermediate tape.
IFREQ = 10; every tenth sample will be output to the intermediate tape; IFREQ must be ≥ 1 .

LTN Logical tape unit input tape is mounted on
Example:
LTN = 10 indicates tape unit 10 is being used for input.

NBLK Number of records per block on ESSA output tape
This variable is used to block the output intermediate tape and to adjust the record size of MSFC output tape.
Example:
NBLK = 30 indicates ESSA tapes are blocked 30 records per block and the intermediate output tape will be blocked 30 records per block.

NSTART Number of starts and stops
Example:
NSTART = 3 indicates there are three start and stop times.

NBOB N/A; this should always be 0
Example:
NBOB = 0

PLYBCK Slow down rate of the analog tape relative to the digitizer tape

Example:

PLYBCK = 8.0 indicates the analog tape was
run at one-eighth the speed of the
digitizer tape.

NSTEP Flag for step calibrations

Example:

NSTEP = 1 indicates data are DC, but a sine
wave signal is used to calibrate.

\$ TIMES

LINK 2

TSTART Start time

TIMMAX Maximum integration time

DELTAT Piece length

IPRINT Print frequency

Example:

IPRINT = 10 printout every tenth piece

\$ DELAY

LINK 2

TAUMIN Minimum lag time

TAUMAX Maximum lag time

DELTAU Lag time resolution

VELLAG Velocity lag control

Example:

VELLAG = 0 for no velocity lags

= 1 calculate velocity lags

SEPART Beam separation

\$CORR

ASSIGN
LINK 2

J Delayed channel number.

K Undelayed channel numbers

(No more than six)

Example:

j = 1, K = 2,3,4, 3*0

This card group is repeated for as many
correlations as required. J = 0 signals
the last correlation request thus a
minimum of two \$CORR groups are required.

SUBROUTINE

\$DELETE

LINK 3

NUMPCE Piece numbers (maximum of ten numbers per group) This card group is repeated as often as required.
NUMPCE = 10*0 signals the last card group
Example:
NUMPCE = 5,7,8,10,12,14, 4*0
NPRINT Print frequency for LINK 3

\$MODIFY

Logical variables

LINK 3

DTREND If true, detrend
NORM If true, normalize
COVAR If true, form covariance
PREMLT If true, multiply by τ
PLOT If true, plot on SC 4020

\$ACCUM

LINK 3

NO Accumulation number
BEGIN Start time
END End time
Up to five accumulations may be specified;
the last group is indicated by NO = 0
Example:
NO = 1, BEGIN = 0.0, END = 1.5

The last data card for LINK 3 is a comments card which describes the type of accumulation procedure used. This is an alphanumeric data card with up to 72 columns available for a comment. It differs from the other data cards in that there is no NAMELIST associated with it and it should be the last card of the data decks.

B. Output

The output from the MLTCOR program has been organized into a series of standard reports that are printed out from the three basic links. These reports were prepared with the use of a special program called SOFT (Simple Output Format Translator).

The SOFT program translates report specifications in a simplified form to FORTRAN IV format statements which will produce the desired output. The SOFT program is a standard SHARE program available from IBM, SHARE distribution number 3379. The types of reports that can be expected from the various links of the program are shown in Figures 19 through 23. These output reports are the principal form of output from the program but it is possible to request plots of the correlations and modified correlations for an SC 4020, Cathode Ray Tube Output Recorder. These plots are requested by setting PLOT = T in the \$MODIFY namelist group of parameters. Examples of these plots are shown in Figures 24 through 27.

```

*****
+EXPERIMENT NUMBER= 000000      ORGANIZATION=G34      TAPE NUMBER=000001 +
+
+   PAGE NUMBER=*** FILE NUMBER=      IDATE=080 69   RUN NUMBER=000000 +
*****

```

```

*****CALIBRATION IDENTIFICATION*****
+CHANNEL+BEGIN RECORD+END RECORD+MINIMUM+MAXIMUM+      CALIBRATION      +      ENGINEERING      +
+ NUMBER+  TIME      +  TIME      + VALUE + VALUE +      SLOPE  + INTERCEPT +  MINIMUM  +  MAXIMUM  +
*****
+ 1 + 140000 + 150000 + -1971.2+ 1983.2+ 0.00253+ 14.98486+ 10.00000+ 20.00000+
+ 2 + 140000 + 150000 + -2036.4+ 2047.0+ 0.00073+ -0.00389+ -1.50000+ 1.50000+
+ 3 + 140000 + 150000 + -2047.0+ 2047.0+ 0.00244+ 15.00006+ 10.00000+ 20.00000+
+ 4 + 140000 + 150000 + -2047.2+ 2040.3+ 0.00245+ 15.00841+ 10.00000+ 20.00000+
+ 5 + 140000 + 150000 + -2010.5+ 2018.2+ 0.00248+ 14.99051+ 10.00000+ 20.00000+
+ 6 + 140000 + 150000 + -0.8+ -0.0+ 0.00000+ 0.00000+ 0.00000+ 0.00000+
+ 7 + 140000 + 150000 + -5.4+ -5.0+ 0.00000+ 0.00000+ 0.00000+ 0.00000+
+ 8 + 140000 + 150000 + 1534.7+ 1534.7+ 0.00000+ -0.00000+ 0.00000+ 0.00000+
+ 9 + 140000 + 150000 + -2.0+ -1.0+ 0.00000+ 0.00000+ 0.00000+ 0.00000+
*****

```

```

*****MODIFICATION OF ORIGINAL TIME SERIES*****
+CHANNEL+SAMPLING+ORIGINAL +ADDITIONAL+MOVING+ CENTRAL +CHANNEL + BEGIN + END +
+ NUMBER+ RATE + SAMPLE + SAMPLE + MEAN +DIFFERENCE+VARIABLE+RECORD +RECORD +
+      +REDUCTION+ REDUCTION+      +      +      + TIME + TIME +
*****
+ 1 + 10.00+ 0.00+ 1 + 0+ 0 + YT+ 140000+ 150000+
+ 2 + 10.00+ 0.00+ 1 + 0+ 0 + ST+ 140000+ 150000+
+ 3 + 10.00+ 0.00+ 1 + 0+ 0 + XT+ 140000+ 150000+
+ 4 + 10.00+ 0.00+ 1 + 0+ 0 + FT+ 140000+ 150000+
+ 5 + 10.00+ 0.00+ 1 + 0+ 0 + GT+ 140000+ 150000+
+ 6 + 10.00+ 0.00+ 1 + 0+ 0 + 000000+ 140000+ 150000+
+ 7 + 10.00+ 0.00+ 1 + 0+ 0 + 000000+ 140000+ 150000+
+ 8 + 10.00+ 0.00+ 1 + 0+ 0 + 000000+ 140000+ 150000+
+ 9 + 10.00+ 0.00+ 1 + 0+ 0 + 0 RCNT+ 140000+ 150000+
*****

```

Figure 19 LINK 1 Reports

```

*****
+ EXPERIMENT NUMBER= 000000 ORGANIZATION=000000 TAPE NUMBER=000000 +
+
+ PAGE NUMBER=*** FILE NUMBER=000000 DATE=000000 RUN NUMBER=000000 +
*****
*****AVERAGES WITH ZERO TIME DELAY*****
+
+ PIECE NUMBER= 6 +
+ ACCUMULATION NUMBER= 6 + PIECEWISE + PIECEWISE + ACCUMULATIVE + MEAN + ACCUMULATIVE + MEAN-SQUARED +
+ ACCUMULATION START= 0.00+CHANNEL+ MEAN VALUE + ROOT-MEAN + MEAN VALUE + STANDARD + ROOT MEAN + STANDARD +
+ ACCUMULATION TIME= 3749.76+ NUMBER+ + SQUARE-VALUE+ + ERROR + SQUARE VALUE+ ERROR +
+
+ PIECE LENGTH= 624.96+
+
+
+ 1 + -0.01301364+ 0.08416948+ -0.01155653+ 0.00000111+ 0.06076186+ 0.00000077+
+ 2 + -0.04709072+ 0.06046378+ -0.04388484+ 0.000000639+ 0.05322358+ 0.00000005+
+ 3 + -0.01699706+ 0.01910937+ -0.01555780+ 0.00000029+ 0.01784660+ 0.00000000+
+ 4 + -0.06015912+ 0.06511997+ -0.05967946+ 0.00000009+ 0.06236742+ 0.00000001+
+ 5 + 0.00501663+ 0.02143924+ 0.00779503+ 0.00000119+ 0.04546366+ 0.00000097+
+
+
+ *****AVERAGES WITH ZERO TIME DELAY*****
+
+ PIECE NUMBER= 6 +
+ ACCUMULATION NUMBER= 6 + PIECEWISE + PIECEWISE + ACCUMULATIVE + MEAN + ACCUMULATIVE + STANDARD +
+ ACCUMULATION START= 0.00+CHANNEL+ MEAN VALUE + STANDARD + MEAN VALUE + STANDARD + STANDARD + ERROR +
+ ACCUMULATION TIME= 3749.76+ NUMBER+ + DEVIATION + + ERROR + DEVIATION + VARIANCE +
+
+ PIECE LENGTH= 624.96+
+
+
+ 1 + -0.01301364+ 0.00115252+ -0.01155653+ 0.00000111+ 0.000059215+ 0.00000076+
+ 2 + -0.04709072+ 0.00023972+ -0.04388484+ 0.000000639+ 0.00014582+ 0.00000002+
+ 3 + -0.01699706+ 0.00001271+ -0.01555780+ 0.00000029+ 0.00001250+ 0.00000000+
+ 4 + -0.06015912+ 0.00010358+ -0.05967946+ 0.00000009+ 0.00005460+ 0.00000000+
+ 5 + 0.00501663+ 0.00007241+ 0.00779503+ 0.00000119+ 0.00033337+ 0.00000097+
+
+

```

Figure 20 LINK 2 Channel Averages

```

*****
*EXPERIMENT NUMBER= Q00000      ORGANIZATION=Q00000  TAPE NUMBER=Q00000 *
*
*   PAGE NUMBER=*** FILE NUMBER=Q00000DATE=Q00000  RUN NUMBER=Q00000 *
*****
*****AVERAGES WITH TIME DELAY*****
*
*   PIECE NUMBER=      6   +AREA UNDER
*ACCUMULATION NUMBER=      6   +CORRELATION=  0.00243350+
*ACCUMULATION START=      0.00+CORRELATED
*ACCUMULATION TIME= 3749.76+CHANNELS
*   PIECE LENGTH= 624.96+ 1 AND 1
*****
*   TIME DELAY++VELOCITY DELAY++ PIECEWISE +
*               ++               ++CORRELATION+
*****
*   -20.00000 ++      0.000   ++      0.00090+
*   -19.00000 ++      0.000   ++      0.00078+
*   -18.00000 ++      0.000   ++      0.00056+
*   -17.00000 ++      0.000   ++      0.00028+
*   -16.00000 ++      0.000   ++     -0.00001+
*   -15.00000 ++      0.000   ++     -0.00024+
*   -14.00000 ++      0.000   ++     -0.00038+
*   -13.00000 ++      0.000   ++     -0.00041+
*   -12.00000 ++      0.000   ++     -0.00034+
*   -11.00000 ++      0.000   ++     -0.00026+
*   -10.00000 ++      0.000   ++     -0.00021+
*    -9.00000 ++      0.000   ++     -0.00019+
*    -8.00000 ++      0.000   ++     -0.00017+
*    -7.00000 ++      0.000   ++     -0.00002+
*    -6.00000 ++      0.000   ++      0.00037+
*    -5.00000 ++      0.000   ++      0.00113+
*    -4.00000 ++      0.000   ++      0.00275+
*    -3.00000 ++      0.000   ++      0.00424+
*    -2.00000 ++      0.000   ++      0.00566+
*    -1.00000 ++      0.000   ++      0.00670+
*     0.00000 ++      0.000   ++      0.00708+
*     1.00000 ++      0.000   ++      0.00670+
*     2.00000 ++      0.000   ++      0.00566+
*     3.00000 ++      0.000   ++      0.00424+
*     4.00000 ++      0.000   ++      0.00275+
*     5.00000 ++      0.000   ++      0.00113+
*     6.00000 ++      0.000   ++      0.00037+
*     7.00000 ++      0.000   ++     -0.00002+
*     8.00000 ++      0.000   ++     -0.00017+
*     9.00000 ++      0.000   ++     -0.00019+
*    10.00000 ++      0.000   ++     -0.00021+
*    11.00000 ++      0.000   ++     -0.00026+
*    12.00000 ++      0.000   ++     -0.00034+
*    13.00000 ++      0.000   ++     -0.00041+
*    14.00000 ++      0.000   ++     -0.00038+
*    15.00000 ++      0.000   ++     -0.00024+
*    16.00000 ++      0.000   ++     -0.00001+
*    17.00000 ++      0.000   ++      0.00028+
*    18.00000 ++      0.000   ++      0.00056+
*    19.00000 ++      0.000   ++      0.00078+
*    20.00000 ++      0.000   ++      0.00090+
*****

```

Figure 21 Piecewise Correlations

```

*****
*EXPERIMENT NUMBER= 000000 ORGANIZATION=000000 TAPE NUMBER=000000 *
*
* PAGE NUMBER=*** FILE NUMBER=000000 DATE=000000 RUN NUMBER=000000 *
*****
*****AVERAGES WITH TIME DELAY*****
*
* PIECE NUMBER= 7 *AREA UNDER *ACCUMULATIVE AREA UNDER *CORRELATED *
* ACCUMULATION NUMBER= 1 *CORRELATION= 0.00120206 *CORRELATION= 0.00069064 * CHANNELS *
* ACCUMULATION START= 0.00 *AREA UNDER *ACCUMULATIVE AREA UNDER *
* ACCUMULATION TIME= 4175.00 *MODIFIED *MODIFIED * 1 AND 1 *
* PIECE LENGTH= 425.00 *CORRELATION= -0.00420377 *CORRELATION= -0.00318567 *
*****
*TIME DELAY*VELOCITY DELAY* PIECEWISE *PIECEWISE *ACCUMULATIVE*STANDARD*ACCUMULATIVE*STANDARD*
*
* ** **CORRELATION** MODIFIED **CORRELATION** ERROR ** MODIFIED ** ERROR **
*
* ** **CORRELATION** **CORRELATION** **CORRELATION**
*****
* -20.0000 ** 0.000 ** 0.00079 ** -0.06292 ** 0.00024 ** 0.00000 ** -0.13279 ** 0.00081 **
* -19.0000 ** 0.000 ** 0.00070 ** -0.07583 ** 0.00022 ** 0.00000 ** -0.13187 ** 0.00088 **
* -18.0000 ** 0.000 ** 0.00051 ** -0.10220 ** 0.00016 ** 0.00000 ** -0.13693 ** 0.00042 **
* -17.0000 ** 0.000 ** 0.00026 ** -0.13834 ** 0.00009 ** 0.00000 ** -0.14633 ** 0.00019 **
* -16.0000 ** 0.000 ** -0.00002 ** -0.17707 ** 0.00002 ** 0.00000 ** -0.15667 ** 0.00017 **
* -15.0000 ** 0.000 ** -0.00025 ** -0.21000 ** -0.00004 ** 0.00000 ** -0.16425 ** 0.00038 **
* -14.0000 ** 0.000 ** -0.00040 ** -0.23071 ** -0.00006 ** 0.00000 ** -0.16705 ** 0.00048 **
* -13.0000 ** 0.000 ** -0.00044 ** -0.23706 ** -0.00005 ** 0.00000 ** -0.16615 ** 0.00084 **
* -12.0000 ** 0.000 ** -0.00038 ** -0.22858 ** -0.00002 ** 0.00000 ** -0.16629 ** 0.00067 **
* -11.0000 ** 0.000 ** -0.00031 ** -0.21791 ** -0.00002 ** 0.00000 ** -0.17387 ** 0.00040 **
* -10.0000 ** 0.000 ** -0.00025 ** -0.21031 ** -0.00005 ** 0.00000 ** -0.19038 ** 0.00021 **
* -9.0000 ** 0.000 ** -0.00023 ** -0.20699 ** -0.00012 ** 0.00000 ** -0.21198 ** 0.00017 **
* -8.0000 ** 0.000 ** -0.00019 ** -0.20129 ** -0.00016 ** 0.00000 ** -0.22714 ** 0.00021 **
* -7.0000 ** 0.000 ** -0.00003 ** -0.17840 ** -0.00012 ** 0.00000 ** -0.21750 ** 0.00024 **
* -6.0000 ** 0.000 ** 0.00039 ** -0.11921 ** 0.00010 ** 0.00000 ** -0.16240 ** 0.00021 **
* -5.0000 ** 0.000 ** 0.00118 ** -0.00799 ** 0.00056 ** 0.00000 ** -0.04644 ** 0.00015 **
* -4.0000 ** 0.000 ** 0.00282 ** 0.22451 ** 0.00155 ** 0.00000 ** 0.20007 ** 0.00007 **
* -3.0000 ** 0.000 ** 0.00432 ** 0.43787 ** 0.00246 ** 0.00000 ** 0.43218 ** 0.00004 **
* -2.0000 ** 0.000 ** 0.00577 ** 0.64229 ** 0.00333 ** 0.00000 ** 0.65182 ** 0.00005 **
* -1.0000 ** 0.000 ** 0.00682 ** 0.79101 ** 0.00396 ** 0.00000 ** 0.81125 ** 0.00009 **
* 0.0000 ** 0.000 ** 0.00720 ** 0.84589 ** 0.00419 ** 0.00000 ** 0.86958 ** 0.00011 **
* 1.0000 ** 0.000 ** 0.00682 ** 0.79101 ** 0.00396 ** 0.00000 ** 0.81125 ** 0.00009 **
* 2.0000 ** 0.000 ** 0.00577 ** 0.64229 ** 0.00333 ** 0.00000 ** 0.65182 ** 0.00005 **
* 3.0000 ** 0.000 ** 0.00432 ** 0.43787 ** 0.00246 ** 0.00000 ** 0.43218 ** 0.00004 **
* 4.0000 ** 0.000 ** 0.00282 ** 0.22451 ** 0.00155 ** 0.00000 ** 0.20248 ** 0.00007 **
* 5.0000 ** 0.000 ** 0.00118 ** -0.00799 ** 0.00056 ** 0.00000 ** -0.04644 ** 0.00015 **
* 6.0000 ** 0.000 ** 0.00039 ** -0.11921 ** 0.00010 ** 0.00000 ** -0.16240 ** 0.00021 **
* 7.0000 ** 0.000 ** -0.00003 ** -0.17840 ** -0.00012 ** 0.00000 ** -0.21750 ** 0.00024 **
* 8.0000 ** 0.000 ** -0.00019 ** -0.20129 ** -0.00016 ** 0.00000 ** -0.22714 ** 0.00021 **
* 9.0000 ** 0.000 ** -0.00023 ** -0.20699 ** -0.00012 ** 0.00000 ** -0.21198 ** 0.00017 **
* 10.0000 ** 0.000 ** -0.00025 ** -0.21031 ** -0.00005 ** 0.00000 ** -0.19038 ** 0.00021 **
* 11.0000 ** 0.000 ** -0.00031 ** -0.21791 ** -0.00002 ** 0.00000 ** -0.17387 ** 0.00040 **
* 12.0000 ** 0.000 ** -0.00038 ** -0.22858 ** -0.00002 ** 0.00000 ** -0.16629 ** 0.00067 **
* 13.0000 ** 0.000 ** -0.00044 ** -0.23706 ** -0.00005 ** 0.00000 ** -0.16615 ** 0.00084 **
* 14.0000 ** 0.000 ** -0.00040 ** -0.23071 ** -0.00006 ** 0.00000 ** -0.16705 ** 0.00048 **
* 15.0000 ** 0.000 ** -0.00025 ** -0.21000 ** -0.00004 ** 0.00000 ** -0.16425 ** 0.00038 **
* 16.0000 ** 0.000 ** -0.00002 ** -0.17707 ** 0.00002 ** 0.00000 ** -0.15667 ** 0.00017 **
* 17.0000 ** 0.000 ** 0.00026 ** -0.13834 ** 0.00009 ** 0.00000 ** -0.14633 ** 0.00019 **
* 18.0000 ** 0.000 ** 0.00051 ** -0.10220 ** 0.00016 ** 0.00000 ** -0.13693 ** 0.00042 **
* 19.0000 ** 0.000 ** 0.00070 ** -0.07583 ** 0.00022 ** 0.00000 ** -0.13187 ** 0.00048 **
* 20.0000 ** 0.000 ** 0.00079 ** -0.06292 ** 0.00024 ** 0.00000 ** -0.13279 ** 0.00081 **
*****

```

Figure 22 LINK 3 Modified Correlation

```

*****
* EXPERIMENT NUMBER= 000000 ORGANIZATION=000000 TAPE NUMBER=000000 *
*
* PAGE NUMBER=*** FILE NUMBER=000000 DATE=000000 RUN NUMBER=000000 *
*****

```

```

*****
* CORRELATED CHANNELS= 1 AND 1+ * MODIFIED * * ACCUMULATION PROCEDURE USED= *
* ACCUMULATION NUMBER= 1 * CORRELATION+CORRELATION+ CHANNEL * THEY WERE ALL USED *
* END OF ACCUMULATION= 4375.00 * STATISTICAL+STATISTICAL+STATISTICAL+ *
* START OF ACCUMULATION= 0.00 * ERROR * ERROR * ERROR *
*****
* AVERAGING TIME+*****
*
* 4375.0000 * 0.00000002+ 0.0022959+ 0.0022959+
*****

```

```

*****
* CORRELATED * MEAN VALUE+ ROOT MEAN *
* CHANNELS * * SQUARE VALUE+ * VELOCITY * * MODIFIED *
* 1 * -0.011618+ 0.0040538 * TIME DELAY+ DELAY * CORRELATION+CORRELATION+
* 1 * -0.011618+ 0.0040538+ * * *
*****
* -20.000000+ 0.000000+ 0.0002378+ -0.1327852+
* -19.000000+ 0.000000+ 0.0002162+ -0.1318671+
* -18.000000+ 0.000000+ 0.0001630+ -0.1369294+
* -17.000000+ 0.000000+ 0.0000893+ -0.1463339+
* -16.000000+ 0.000000+ 0.0000150+ -0.1566706+
* -15.000000+ 0.000000+ -0.0000386+ -0.1642489+
* -14.000000+ 0.000000+ -0.0000589+ -0.1670547+
* -13.000000+ 0.000000+ -0.0000487+ -0.1661454+
* -12.000000+ 0.000000+ -0.0000197+ -0.1662904+
* -11.000000+ 0.000000+ -0.0000192+ -0.1738694+
* -10.000000+ 0.000000+ -0.0000545+ -0.1903777+
* -9.000000+ 0.000000+ -0.0001160+ -0.2119777+
* -8.000000+ 0.000000+ -0.0001607+ -0.2271428+
* -7.000000+ 0.000000+ -0.0001153+ -0.2175049+
* -6.000000+ 0.000000+ 0.0001048+ -0.1623974+
* -5.000000+ 0.000000+ 0.0005638+ -0.0464379+
* -4.000000+ 0.000000+ 0.0015505+ 0.2026835+
* -3.000000+ 0.000000+ 0.0024601+ 0.4321819+
* -2.000000+ 0.000000+ 0.0033304+ 0.6518241+
* -1.000000+ 0.000000+ 0.0039620+ 0.8112452+
* 0.000000+ 0.000000+ 0.0041936+ 0.8695849+
* 1.000000+ 0.000000+ 0.0039620+ 0.8112452+
* 2.000000+ 0.000000+ 0.0033304+ 0.6518241+
* 3.000000+ 0.000000+ 0.0024601+ 0.4321819+
* 4.000000+ 0.000000+ 0.0015505+ 0.2026835+
* 5.000000+ 0.000000+ 0.0005638+ -0.0464379+
* 6.000000+ 0.000000+ 0.0001048+ -0.1623974+
* 7.000000+ 0.000000+ -0.0001153+ -0.2175049+
* 8.000000+ 0.000000+ -0.0001607+ -0.2271428+
* 9.000000+ 0.000000+ -0.0001160+ -0.2119777+
* 10.000000+ 0.000000+ -0.0000545+ -0.1903777+
* 11.000000+ 0.000000+ -0.0000192+ -0.1738694+
* 12.000000+ 0.000000+ -0.0000197+ -0.1662904+
* 13.000000+ 0.000000+ -0.0000487+ -0.1661454+
* 14.000000+ 0.000000+ -0.0000589+ -0.1670547+
* 15.000000+ 0.000000+ -0.0000386+ -0.1642489+
* 16.000000+ 0.000000+ 0.0000150+ -0.1566706+
* 17.000000+ 0.000000+ 0.0000893+ -0.1463339+
* 18.000000+ 0.000000+ 0.0001630+ -0.1369294+
* 19.000000+ 0.000000+ 0.0002162+ -0.1318671+
* 20.000000+ 0.000000+ 0.0002378+ -0.1327852+
*****

```

Figure 23 Final Tabulation

RUN NO.= 993
 UNMOD ERROR = 4.88901×10^{-03}
 MOD ERROR = 4.88901×10^{-03}
 TOTAL PIECE NO.= 38.
 ACCUM. NO.= 1.
 CORRELATION CHANNELS= 1. AND 1.

PIECEWISE CORRELATION
 DATE= 070969
 FINAL SAMPLE RATE= 5.00000×10^{-03}
 ACCUM TIME= 4.56000×10^{-01}
 PIECE LENGTH= 1.20000×10^{-02}
 ORGANIZATION= MSFC
 CHANNEL VARIABLES= 000000000 000000

470960
 029 000

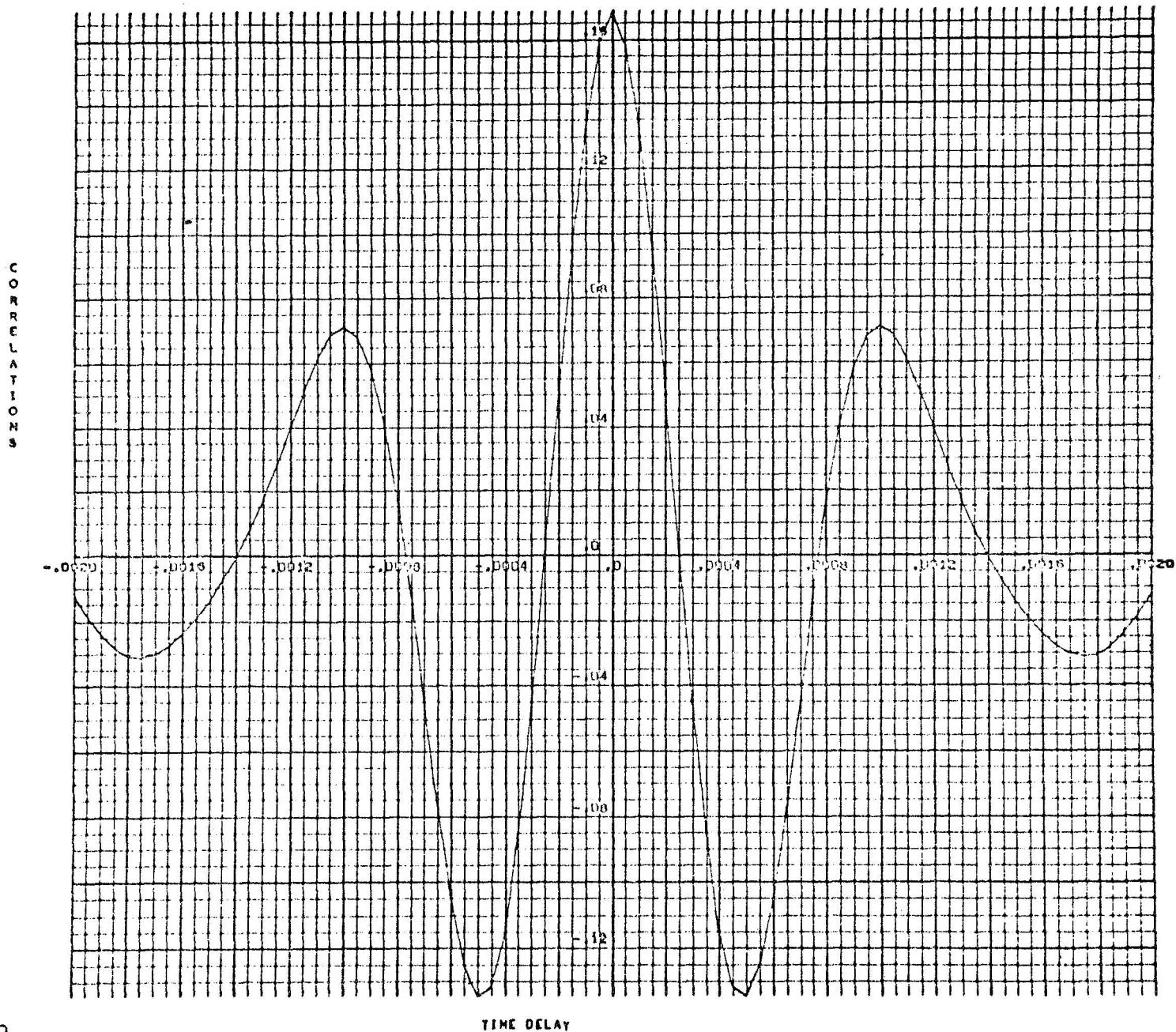


Figure 24 Piecewise Correlation

RUN NO.= 993
 UNMOD ERROR = 4.88901×10^{-03}
 MOD ERROR = 4.88901×10^{-03}
 TOTAL PIECE NO.= 38.
 ACCUM. NO.= 1.
 CORRELATION CHANNELS= 1. AND 1.

PIECEWISE MODIFIED CORRELATION
 DATE= 070969
 FINAL SAMPLE RATE= 5.00000×10^{-05}
 ACCUM TIME= 4.56000×10^{-01}
 PIECE LENGTH= 1.20000×10^{-02}
 ORGANIZATION= MSFC
 CHANNEL VARIABLES= 000000 AND 000000

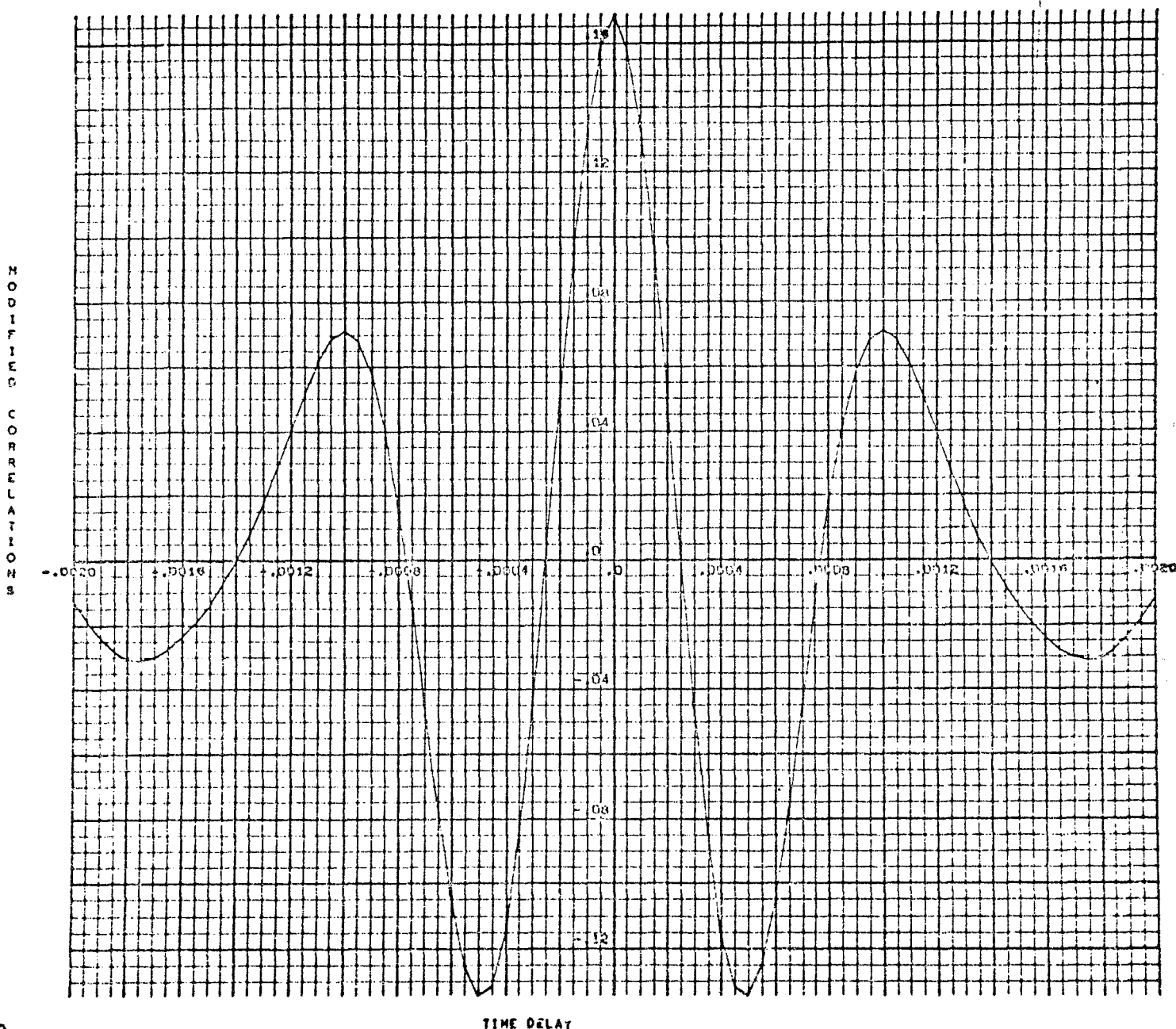
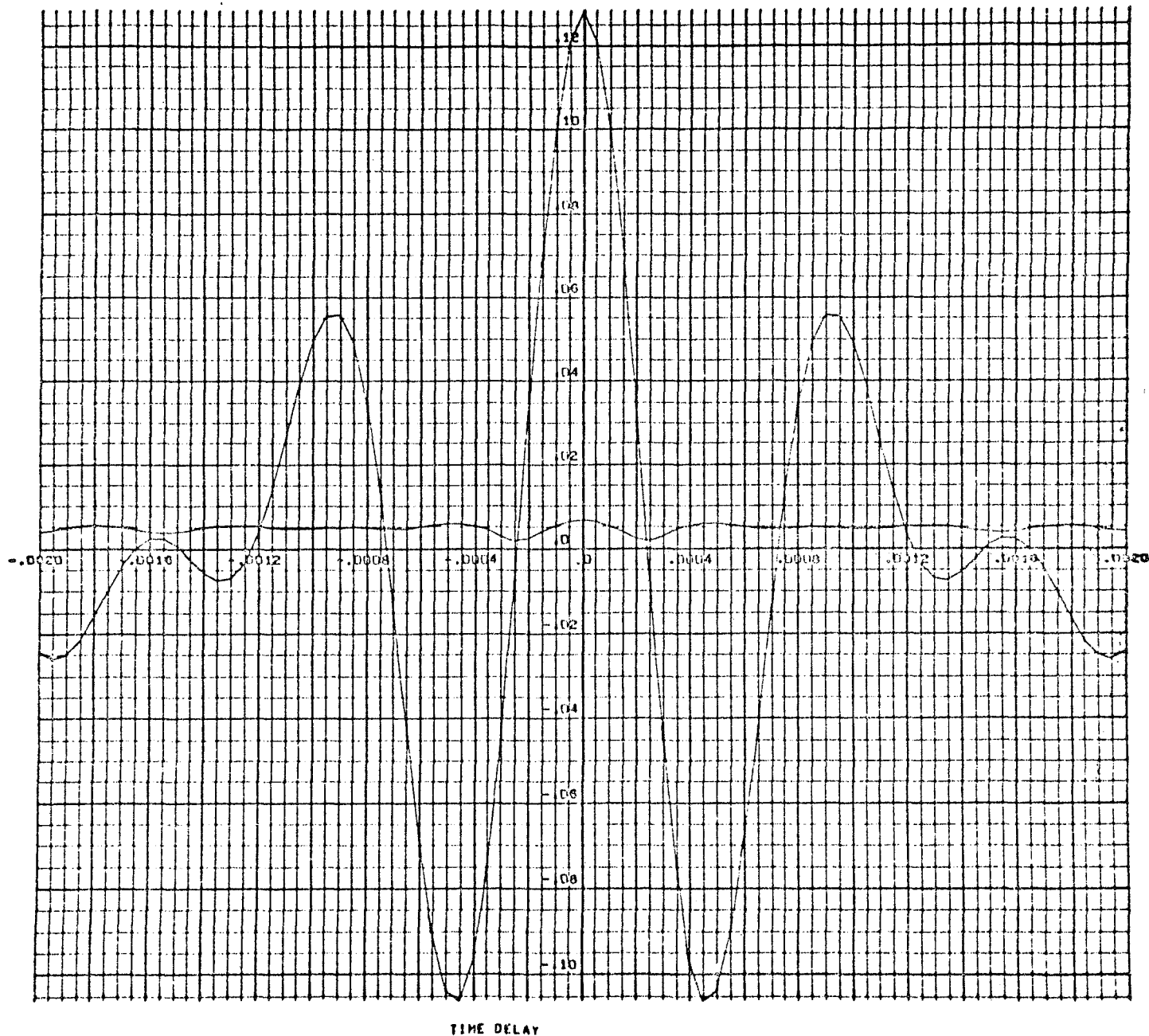


Figure 25 Piecewise Modified Correlation

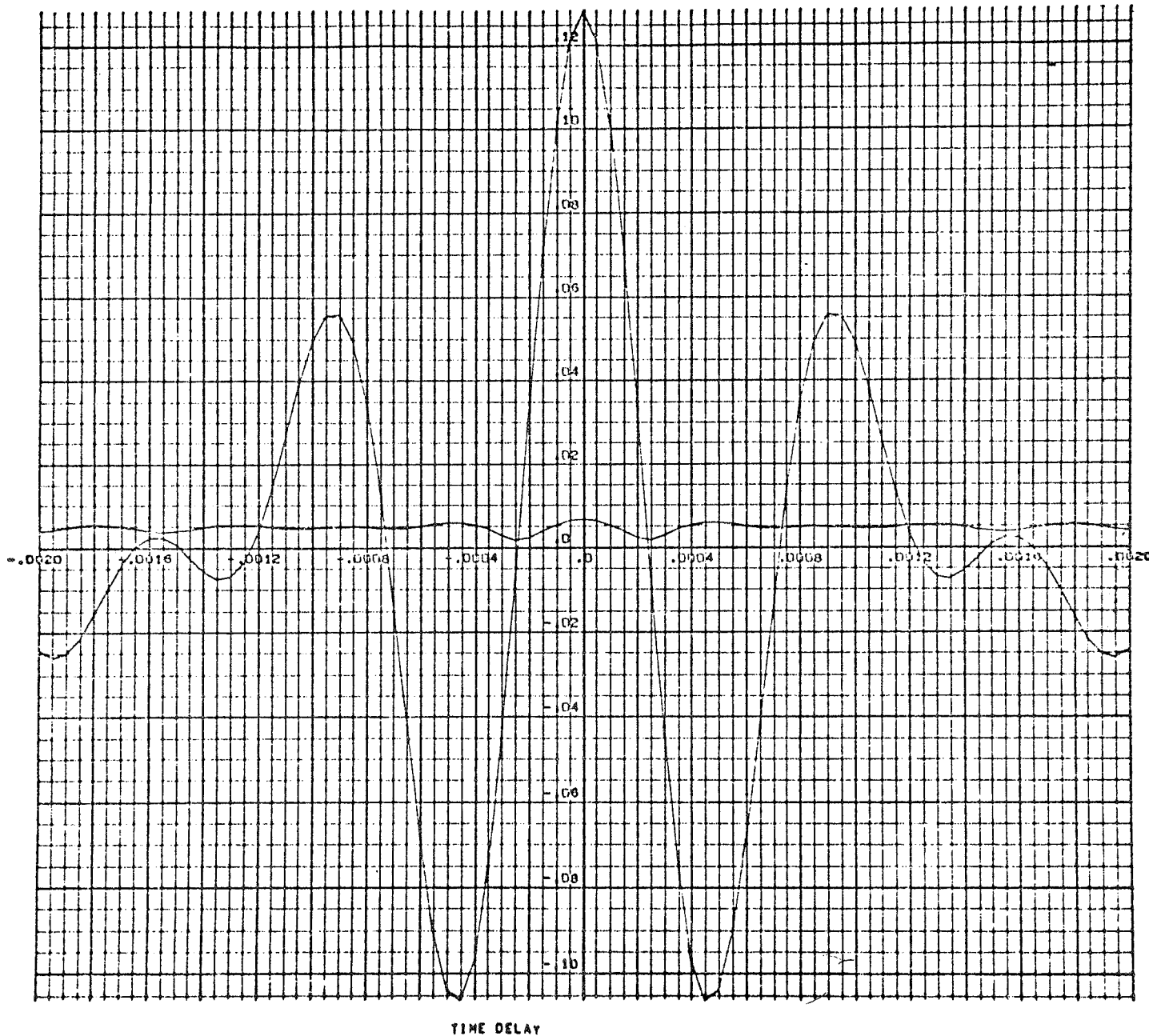
CORRELATION AND STANDARD ERROR



RUN NO.: 993
 ACCUMULATIVE CORRELATION AND STANDARD ERROR
 UNMOD ERROR = 4.8890×10^{-03}
 MOD ERROR = 4.8890×10^{-03}
 TOTAL PIECE NO. = 36
 ACCUM. NO. = 1
 CORRELATION CHANNELS = 1. AND 1.
 DATE = 070969
 FINAL SAMPLE RATE = 5.00000×10^{-05}
 ACCUM TIME = 4.56000×10^{-01}
 PIECE LENGTH = 1.20000×10^{-02}
 ORGANIZATION = MSFC
 CHANNEL VARIABLES = 000000 AND 000000

Figure 26 Accumulative Correlation and Standard Error

MODIFIED CORRELATION AND STANDARD ERROR



RUN NO.= 993
 ACCUMULATIVE MODIFIED CORRELATION AND STANDARD ERROR
 UNKNOWN ERROR = 4.88901×10^{-03}
 MOD ERROR = 4.88901×10^{-03}
 TOTAL PIECE NO.= 38.
 ACCUM. NO.= 1.
 CORRELATION CHANNELS= 1. AND 1.
 DATE= 070939
 FINAL SAMPLE RATE= 5.00000×10^{-03}
 ACCUM TIME= 4.56000×10^{-01}
 PIECE LENGTH= 1.20000×10^{-02}
 ORGANIZATION= MSFC
 CHANNEL VARIABLES= 00000000 000000

470960
 032 000

Figure 27 Accumulative Modified Correlation and Standard Error

REFERENCES

1. F. R. Krause and R. R. Jayroe, Development of New Computer Code for Cross-Beam Data, 1968.
2. R. R. Jayroe and M. Y. Su, Optimum Averaging Time of Meteorological Data with Time Dependent Means, 1968.
3. F. R. Krause and B. C. Hablutzel, Noise Elimination by Piecewise Cross Correlation of Photometer Outputs, 1968.
4. F. R. Krause, J. A. Jones, M. T. Fisher, and J. Pooley, Digital Analysis of Random Processes by Piecewise Estimated Covariance Functions, 1967.

Blank Page

APPENDIX A
MLTCOR COMPUTER PROGRAM USERS MANUAL

1. INTRODUCTION

The purpose of this manual is to establish a thorough understanding of how to use the MLTCOR program and explain in detail the input parameters, and some of the system features of the IBM 7094 and the Univac 1108 EX VIII. Restrictions and problems a user might encounter are explained in detail. Program diagnostics are listed and are self-explanatory. These help the user to identify problems in jobs that have been set up.

2. INPUT PROBLEM PARAMETERS

A. LINK 1

1. KRUN Restart constant.
Example:
KRUN = 1,2,3; runs all three links in sequence.
KRUN = 2; will run only LINK 2.
2. NBEAMS Number of beams or sensors.
Example:
NBEAMS = 7; indicates there are 7 sensors or
7 beams on input tape.
3. NFLSKC Number of files to be skipped before reading
MSFC Calibration records.
Example:
NFLSKC = 3; indicates calibration data are
in file 4.
4. NFLSKD Number of files to be skipped before reading
MSFC data records.
Example:
NFLSKD = 2; indicates data are in file 3.

5. MVMEAN Moving mean span desired for filtering data.
 (Preferably an odd integer).
 Example:
 MVMEAN = 11; the filter span to be used will
 be 11 points. If MVMEAN = 0, this option
 will be by-passed.
6. NCENDF Central differences.
 Example:
 NCENDF = 1; central differences are calculated.
 NCENDF = 0; this option will be by-passed.
7. NOBEAM The total number of beams on which to be operated.
 Example:
 NOBEAM = 3; indicates 3 beams of NBEAMS on which
 to be operated.
 If MVMEAN = 0, this option is ignored.
8. NWHICH Which beams of the NBEAMS on which to be operated.
 Example:
 NWHICH = 1,3,4; indicates beams 1,3, and 4 only.
 If MVMEAN = 0; this option is ignored.
9. TIMSAM Sample rate of input data.
 Example:
 TIMSAM = .1; indicates data were sampled at
 10 samples per second.
10. VALMAX Maximum value for calibrations.
 Example:
 VALMAX = 1.0, 1.0, 1.0; indicates three channels
 are calibrated to maximum value of 1.0.
11. VALMIN Minimum value for calibrations.
 Example:
 VALMIN = -1.0, -1.0, -1.0; indicates three
 channels are calibrated to minimum value of -1.0.
12. *START Start time on raw data input tape. (There can be
 up to 5 start times).
 Example:
 START = 1.0, 1.5, 2.0, 2.5, 6.0; indicates 5
 starting points on the input tape.
13. *STOP Stop time on raw data input tape. (There can be
 up to 5 stop times).
 Example:
 STOP = 1.4, 2.0, 2.4, 5.9, 6.5; indicates the
 5 stop times for each respective start time in
 the example above.

14. NFILE Number of files to be shipped before reading ESSA data.
Example:
NFILE = 3; indicates ESSA data are in file 4.
15. MSFC Flag to the program indicating MSFC formatted tape.
Example:
MSFC = 1; will translate MSFC formatted tapes.
If MSFC = 0, program will translate ESSA formatted tapes.
16. IFREQ Desired output sample rate to intermediate tape.
Example:
IFREQ = 1; every sample will be output to the intermediate tape.
IFREQ = 10; every tenth sample will be output to the intermediate tape. IFREQ must be ≥ 1 .
17. LTN Logical tape unit input tape is mounted on.
Example:
LTN = 10; indicates tape unit 10 is being used for input.
18. *NBLK Number of records per block on ESSA output tape. This variable is used to block the output intermediate tape and to adjust the record size of MSFC output tape.
Example:
NBLK = 30; indicates ESSA tapes are blocked 30 records per block and the intermediate output tape will be blocked 30 records per block.
19. NSTART Number of starts and stops.
Example:
NSTART = 3; indicates there are 3 start and stop times.
20. *NBOB N/A. This should always be 0.
Example:
NBOB = 0.
21. PLYBCK Slow down rate of the analog tape relative to the digitizer tape.
Example:
PLYBCK = 8.0; indicates the analog tape was run at 1/8 the speed of the digitizer tape.
22. NSTEP Flag for step calibrations.
Example:
NSTEP = 1; indicates data are DC, but a sine wave signal is used to calibrate.

23. NDEBUG Intermediate printout of time series data.
Example:
NDEBUG = 0; no printout
NDEBUG = 1; ID's and calibration data printed.
NDEBUG = 2; ID's calibration and time series data printed.
24. RUN Identifies run number.
Example:
RUN = 6H 207
25. ORGAN Identifies organization.
Example:
ORGAN = 6H MSFC

B. LINK 2

1. COVAR Removes the mean from each data piece. This option is a logical operation and should be set to true (T) or false (F)
Example:
COVAR = T; mean for each data piece will be removed.
2. START Start time for LINK 2 if different from LINK 1, or if restarting program at LINK 2. There can be up to 5 start times.
3. STOP Stop time for LINK 2. Same as above.
4. DELTAT Data piece length.
Example:
Deltat = 450.0; piece length of each correlation will be 450.0 seconds long.
5. IPRINT Intermediate printout option
Example:
IPRINT = 10; every tenth piece calculated will be printed.
6. TAUMIN Minimum lag time of correlations.
Example:
TAUMIN = 0.0; there will be no time displacement of correlated data for the first lag point.
7. TAUMAX Same as above except for the maximum lag time.
8. DELTAU Interval between lag points. This also determines the number of lag points used.

9. *VELLAG This should be set to "0" or "1". 1 if velocity as a function of time lags are desired. 0 if not.
10. SEPART Physical separation of the correlated beams. These should be specified in the same order as correlations are specified.
Example:
SEPART = 128.0, 180.0; the first two beams correlated are separated by 128.0 meters.
11. *J Undelayed channel to be correlated.
Example:
J = 5; channel 5 will be the undelayed channel.
12. K Delayed channel to be correlated. There can be up to six undelayed channels.
Example:
K = 1,2,3,4,5,6

C. LINK 3

1. NUMPCE Piece numbers to be deleted from the accumulation of pieces.
Example:
NUMPCE = 6, 18, 19, 31; these pieces will be deleted from the accumulations. There can be up to ten deletions.
2. NPRINT Intermediate printout of correlations from LINK 3. Same as IPRINT in LINK 2.
3. NORM Correlations will be normalized if this option is set to (T). Set to (F) if correlations are not to be normalized. This is a logical operation.
Example:
NORM = T; correlations will be normalized
4. DTREND Removes the trend in the data being correlated. Logical operation same as above.
5. PREMLT Multiplies the correlation value by its associated time lag. Logical operation same as above.
6. PLOT Plots all correlations. Logical operation same as above.
7. *NO Accumulation number
8. BEGIN Time to begin accumulation

9. END Time to end accumulation
10. Blank card or card containing alphanumeric data explaining the type of modifications used on the data.

3. INPUT/OUTPUT TAPES (7094)

A. System Tapes

- (1) A1:(N/A); System residence containing library subroutines.
- (2) A2:(5); Card input (namelist data).
- (3) A5:(8); Program residence containing source or object program.
- (4) A6:(10); LINK 1 output.
- (5) A8:(N/A); 4020 plotting output.
- (6) B1:(6); Reports output (LINK 2 and LINK 3).
- (7) B3:(2); Overlay tape.
- (8) B5:(9); Non-system input.
- (9) B6:(11); LINK 2 output.

Input/Output Tapes (1108)

All system usage is located on mass drum storage. Users input and output tapes are selected at run time as specified on the 1108 run cards optional to the programmer. (See job set-up).

4. DIAGNOSTIC

A. LINK 1

- (1) End of file on input tape (B5).
- (2) Irrecoverable parity error on tape (B5).
- (3) System messages.

B. LINK 2

- (1) Storage exceeded.
- (2) Correlations not specified correctly.
- (3) Time span and piece off.
- (4) Time span greater than pieces.
- (5) Insufficient data in time interval.
- (6) Duplicate channels.
- (7) System messages.

C. LINK 3

- (1) System messages.

5. USERS RESTRICTIONS AND NOTES

A. LINK 1

- (1) At digitizing time, the number of words per physical record should not exceed 6000. This will allow enough area in the dynamic storage for data conditioning.
- (2) The logical record lengths of the intermediate tapes are controlled by the NBLK constant. For ESSA tapes this must be the number of records per blocked record on the input tape. For MSFC NBLK controls the number of records per blocked record output. These blocked records should not exceed 512 words. Usually NBLK = 30.

B. LINK 2

- (1) To eliminate storage overflow, the following calculation should be made:

$$\left[12800 - \left(\text{NTLAGS} + \text{NCRT} \times \text{NTLAGS} \times \text{VELLAG} + 9 \times \text{NCH} \right. \right. \\ \left. \left. + \text{NCRT} \times (\text{NTLAGS} + 1) + \text{MAXINC} \times \text{NCH} \right) \right] / (\text{MSUB} \times \text{NCH}) \geq 1$$

Where:

NTLAGS two times number of lags plus one
NCRT total number of correlations
VELLAG input constant
NCH total number of channels being used
MAXINC maximum time lag divided by time sampled
MSUB number of samples per piece

C. LINK 3 - None

NOTES LINK 1

- (1) (12. and 13.) Stop time will be adjusted to the last time on input tape on termination of LINK 1, and passed on to LINK 2 if LINK 1 and LINK 2 are run in sequence. If LINK 2 is run as a restart it is the responsibility of the user to get the stop time from LINK 1 output.
- (2) (18.) If NBLK is selected such that the number of words per plocked record exceeds 512 core locations, some constants and machine instructions are destroyed.
- (3) (20.) This constant is used to print out time, 1/square root of time, mean and RMS values for each channel.

NOTES LINK 2

- (1) (9.) Considerable amount of storage is needed for these values if several correlations are requested. If core storage is marginal this should be set to "0" and the number of correlations times the number of lags will be saved.
- (2) (11.) There can only be one "J" for each set of "K's". These can be repeated until there has been 12 correlations specified.
- (3) (11.) If auto correlations are requested then a DC trend is calculated by removing the square of the means from the area in under the correlation for each auto correlation, and square rooting the value. The area in under the curve replaces $\bar{X}^2$ in the standard error calculations.

NOTES LINK 3

- (1) (7.) There can be up to five accumulation spans and each span number identified by number 7,8,9 can be repeated up to five times.

6. JOB SETUP

JOB SET-UP (7094)

```
$JOB          NASA-JAYROE BIN 309,470960,00,11,14MCE
$EXECUTE      IBJOB
$IJOB         FIOCS,MAP
$FILE         -UNIT03-,NONE
$FILE         -UNIT08-,NONE
$FILE         -UNIT13-,NONE
$FILE         -UNIT14-,NONE
$FILE         -UNIT15-,NONE
$IEDIT        SYSCK1,SCHF01
$IIBFTC MLTCOR                                     MLTC0000
```

```
.
.
.
.
.
.
END
```

```
$DATA
$LINKNO
KRUN=1,2,3,4
$END
$NAMDAT
NBEAMS=2
NFLSKC=0
NFLSKD=1
MVMEAN=0
NCENDF=0
NOBEAM=2
NWHICH=1,2
TIMSAM=.00005
VALMAX=1.0,1.0,1.0,1.0,1.0
VALMIN=-1.0,-1.0
START=0.0
STOP=.5
NFILE=0
MSFC=1
IFREQ=1
LTN=9
NBLK=40
NSTART=1
NBOB=1
PLYBCK=1.
NSTEP=0
NDEBUG=1
$END
```

```

$IDINFO
RUN=6H    993
ORGAN=6H  MSFC
$END
$CVARNC
COVAR=T
$END
$TIMES
STOP=.4900
DELTAT=.012
IPRINT=0
$END
$DELAY
TAUMIN=0.0
TAUMAX=.002
DELTAU=.00005
VELLAG=0
SEPART=.01,.02,.03,.04,.05
$END
$CORR
J=1,K=1,5*0
$END
$CORR
J=0,K=6*0
$END
$DELETE
  NUMPCE=10*0
NPRINT=5
$END
$MODIFY
  NORM=F,DTREND=F,PREMLT=F,PLOT=T
$END
$ACCUM
  NO=1, BEGIN=0.0 , END=.444
$END
$ACCUM
  NO=0, BEGIN=0.0, END=0.0
$END
  NO MODIFICATION
-
-

```

JOB SET-UP (1108)

-RUN,T CSU204,470960,JAYROEBIN309,10,999
-ASG,T 9,T,36124
-ASG,T 10,T
-ASG,T 12,T
-FOR,IS MLTCOR,MLTCOR

.
.
.
.
.
.

END

-MAP,IL MAIN,MAIN
SEG MLTCOR
IN MLTCOR,DEC002
SEG AA*,(MLTCOR)
IN DEC003
SEG BB*,(MLTCOR)
IN DECK 21
SEG CC*,(MLTCOR)
IN DECK31
-XQT,D MAIN
\$LINKNO
KRUN=1,2,3,4
\$END
\$NAMDAT
NBEAMS=9
NFLSKC=0
NFLSKD=0
MVMEAN=0
NCENDF=0
NOBEAM=0
NWHICH=0
TIMSAM=1.0
VALMAX=1.0
VALMIN=1.0
START=0.0
STOP=180.0
NFILE=0
MSFC=0
IFREQ=1
LTN=9
NBLK=40
NSTART=1
NBOB=0
PLYBCK=1.00
NSTEP=0
NDEBUG=2
\$END

```

$IDINFO
RUN=6H      100
ORGAN =6H   MSFC
$END
$CVARNC
COVAR=F
$END
$TIMES
DELTAT =40.0
IPRINT=0
$END
$DELAY
TAUMIN=0.0
TAUMAX=20.0
DELTAU=1.0
VELLAG=0
$END
$CORR
J=4,K=4,5*0
$END
$CORR
J=0,K=6*0
$END
$DELETE
NUMPCE=10*0
NPRINT=1
$END
$MODIFY
NORM=F,DTREND=F,PREMLT=F,PLOT=T
$END
$ACCUM
NO=1,BEGIN=0.0,END=120.0
$END
$ACCUM
NO=0,BEGIN=0.0,END=0.0
$END
      STRAIGHT TIME
-PMD,E
- FIN
- FIN
- FIN
- FIN

```

APPENDIX B
PROGRAM LISTINGS

```

$JOB      NASA-JAYRBE RIN 309,470960,00,11,14MCE
$EXECUTE  18J08
$IBJ08    FIOCS,MAP
$FILE     -UNIT02-,NONE
$FILE     -UNIT12-,NONE
$FILE     -UNIT13-,NONE
$FILE     -UNIT14-,NONE
$FILE     -UNIT15-,NONE
$IBFTC MLTC2R
C          SC(1)= TAPE NUMBER
C          SC(2)= ORGANIZATION
C          SC(3)= FILE COUNT
C          SC(5)= START TIME
C          SC(6)= STOP TIME
C          SC(6+1)= NAME OF VARIABLE ONE
C          .
C          .
C          SC(6+N) = NAME OF NTH VARIABLE
C          SC(6+N+1) = MINIMUM VALUE VARIABLE ONE
C          .
C          .
C          SC(6+2N)= MINIMUM VALUE OF NTH VARIABLE
C          SC(6+2N+1)= MAXIMUM VALUE VARIABLE ONE
C          .
C          .
C          SC(6+3N)= MAXIMUM VALUE OF NTH VARIABLE
C          SC(7+3N)= SAMPLING RATE
C          SC(8+3N)= SAMPLE REDUCTION AT REAL TIME
C          WHERE N= NR OF BEAMS OR SENSORS
C          COMMON START(5),STOP(5),SC(53),VALMIN(12),VALMAX(12)
1          1,SLOPE(12),BIAS(12),LTN,NBEAMS,NFLSKC,NFLSKC,MVMEAN,
2          2TIMSAM,AKDS,IFREQ,MSEC,NFILE,NMS(12),VMEAN(12),NORUN,EXPND,NBPGF,
4          4AVGMIN(12),AVGMAX(12),NELK,NSTART,
3          3CONTIN
          DIMENSION KRUN(4)
          DIMENSION NWHICH(12)
          NAMELIST/LINKN0/KRUN
          NAMELIST/ILINF0/RUN,ORGAN
          NAMELIST/NAMDAT/NBEAMS,NFLSKC,NFLSKC,MVMEAN,NCENCF,N0BEAM,NWHICH,
2          2TIMSAM,VALMAX,VALMIN,
1          1START,STOP,NFILE,MSEC,IFREQ,LTN,NELK,NSTART,N808,PLYBCK,NSTEP
2          2,NDEBLG
          D3 10 11=1,4
          KRUN(1)=0
10         CONTINUE
          CALL SOL2CK (LATE,TIME,FSEC,F60SEC)
          CONTIN=F60SEC
C          READ INPUT CARDS UNDER NAMELIST
          READ (5,LINKN0)
          WRITE (6,LINKN0)
          READ (5,NAMDAT)
          WRITE (6,NAMDAT)
          IF (MSEC) 21,21,20
20         READ (5,ILINF0)
          SC(1)=RUN
          SC(2)=ORGAN
21         CONTINUE
          NWDS=(NBEAMS+1)*NBLK

```

MLTC0000

```

TIMSAM=TIMSAM/PLYBCK
SPØT=0.0
I=1
IF (KRUN(I) .EQ. 1) GØ TØ 1
GØ TØ 4
1 I=I+1
CALL LINK1 (NØØØ,NCENDØ,PLYBCK,NWHICH,NØBEAR,NSTEP,NØEPLG,SPØT)
4 IF (KRUN(I) .EQ. 2) GØ TØ 2
GØ TØ 5
2 I=I+1
CALL LINK2(KRUN,SPØT)
5 IF (KRUN(I) .EQ. 3) GØ TØ 3
GØ TØ 6
3 CØNTINUE
I=I+1
CALL LINK3(KRUN(I))
6 CØNTINUE
IF (KRUN(I) .EQ. 4) GØ TØ 7
GØ TØ 8
7 CØNTINUE
8 CØNTINUE
CALL SCLØCK (DATE,TIME,ESEC,E60SEC)
CØMTIM=(E60SEC-CØMTIM)/60.0
WRITE (6,1199) CØMTIM
1199 FØRMAT (40HØC.P.U. TIME REQUIRED FØR THIS 7Ø94 RUN=F1Ø.5)
STØP
END

```

```

$IBFTC DECKCJ                                     DECKØØØØ
SUBRØUTINE ØUT1(N,EXPNØ,ØRG,NØTAPE,NØPAGE,NØFILE,IDATE,NØRUN)
WRITE(N,1000) EXPNØ,ØRG,NØTAPE,NØPAGE,NØFILE,IDATE,NØRUN
1000 FØRMAT(1H1,74(1H+),/21H +EXPERIMENT NUMBER= ,A6,1X,13HØRGANIZATION FM 1000
2=,A6,14H TAPE NUMBER=,A6,2H +,/2H +,72X,1H+,/2H +,6X,12HPAGE NUMB FM 1000
3ER=,13,13H FILE NUMBER=,A6,5HDATE=,A6,14H RUN NUMBER=,A6,2H +,/1
4H ,74(1H+))
RETURN
END

```

```

$ORIGIN      ALPHA,SYSUT2,REW
$IBFTC LINK11
SUBROUTINE LINK1 (NB08,ACENDF,PLYBCK,NWFICH,N0BEAM,NSTEP,NDEBUG,
1SP0T)
  COMMON START(5),STOP(5),SC(53),VALMIN(12),VALMAX(12)
1,SL0PE(12),BIAS(12),LTN,NRFAMS,NFLSKC,NFLSKC,VMEAN,
2TIPSAM,NWDS,IFREC,MSFC,NFILE,RMS(12),VMEAN(12),N0RUN,EXPNO,N0PAGE,
4AVGMIN(12),AVGMAX(12),NELK,NSTART
  DIMENSION NWHICH(1)
  DIMENSION BUFOUT(512)
  DIMENSION DATA(9000)
  DIMENSION VMEAN(12),RRMS(12)
  DIMENSION COM(1)
  EQUIVALENCE (START(1),COM(1))
  DATA NFLAG/0/
  DATA NCOUNT/0/
C  INITIALIZE VARIABLES FOR RMS CALCULATIONS
1121  FORMAT (1H1,117H TIME CH1 CH2 CH3 CH4
1 CH5 CH6 CH7 CH8 CH9 CH10 CH11
2 )
  NUMCOM=175
  NT0TN0=0
  NFIN=NWDS-NBEAMS
  DO 253 J=1,NBEAMS
  VMEAN(J)=0.0
  VVMEAN(J)=0.0
  RMS(J)=0.0
  RRMS(J)=0.0
253  CONTINUE
1  CONTINUE
  IF (MSFC) 302,302,300
300  CALL TPMSFC (DATA,IERR,PLYBCK,NSTEP,NDEBUG)
  GO TO (301,999,999,301,999),IERR
302  CONTINUE
  CALL TPESSA (DATA,IERR,NDEBUG)
  GO TO (301,999,999,301,999),IERR
301  CONTINUE
  IF (NFLAG .GT. 0) GO TO 400
  WRITE(10) (COM(I),I=1,NUMCOM)
  NFLAG=1
400  CONTINUE
  I=0
150  I=I+1
  DO 160 J=1,NBEAMS
  I=I+1
  DATA(I)=DATA(I)*SL0PE(J)+BIAS(J)
160  CONTINUE
  IF (I .GE. NWDS) GO TO 151
  GO TO 150
151  CONTINUE
C  CALCULATE MEANS AND RMS VALUES HERE
  IF (IERR .EQ. 2) GO TO 401
  IF (NDEPLG .EQ. 0) GO TO 401
  I=0
250  I=I+1
  DO 251 J=1,NBEAMS
  I=I+1
  RMS(J)=RMS(J)+DATA(I)**2
  VMEAN(J)=VMEAN(J)+DATA(I)
251  CONTINUE
  IF (I .GE. NWDS) GO TO 152

```

```

      G0 T0 250
152  CONTINUE
      NT0TN0=N10TN0+NWDS
401  CONTINUE
      IF (MVMEAN) 144,141,145
C    CALCULATE INTEGERS FOR INDEXING IN MOVING MEAN SUBROUTINE
145  FACT0R=(MVMEAN/2)+1
      TIMC0R=FACT0R*TIMSAM
      NFACTR=MVMEAN*(NBEAMS+1)
      ITEMP=(NWDS/NFACTR)*NFACTR+NWDS
      NEWWDS=ITEMP+NFACTR
      NDLSPN=NFACTR
      INDEX=NWDS+1
      INDEX2=INDEX+NFACTR-1
      INDEX6=(NBEAMS+1)*IFREQ
      INDEX3=ITEMP-INDEX6
      INDEX7=ITEMP-NWDS
      AVERAG=MVMEAN
      MVMEAN=-MVMEAN
      INDEX5=ITEMP-(NBEAMS+1)*((IFREQ-1)
      M=1
      II=INDEX
144  CONTINUE
      I=0
      II = II-1
      IF (MVMEAN) 140,141,140
140  II=II+1
      I=I+1
      DATA(II)=DATA(I)
142  IF (I .GT. NWDS) G0 T0 1
      IF (II .EQ. NEWWDS) G0 T0 143
      G0 T0 140
143  CALL VM2VMN (DATA,ITEMP,TIMC0R,NDLSPN,INDEX,AVERAG,NWHICH,N0BEAM)
      INDEX4=INDEX-ITEMP+INDEX5+(NBEAMS+1)*((IFREQ-1)
      INDEX5=INDEX4+NBEAMS
602  D0 600 II=INDEX4,INDEX5
      BUF0UT(M)=DATA(II)
      M=M+1
600  CONTINUE
      IF (M .LT. NWDS) G0 T0 603
      IF (NDEBUG .LE. 1) G0 T0 2000
      WRITE (6,1121)
      D0 2011 IX=1,NBLK
      IY=IX*(NBEAMS+1)
      IZ=IY-NBEAMS
      WRITE (6,1112) (BUF0UT(MM),MM=IZ,IY)
2011 CONTINUE
2000 CONTINUE
      WRITE(10) (BUF0UT(MM),MM=1,NWDS)
      SP0T=BUF0UT(NFIN)
      M=1
603  CONTINUE
      IF (INDEX4 .GE. INDEX3) G0 T0 601
      INDEX4=INDEX4+INDEX6
      INDEX5=INDEX4+NBEAMS
      G0 T0 602
601  III=ITEMP
      D0 146 II=INDEX,INDEX2
      III=III+1
      DATA(II)=DATA(III)
146  CONTINUE

```

```

      II=INDEX2
      GO TO 140
141  CONTINUE
C    SET UP INDEXING FOR CENTRAL DIFFERENCES
      IF (NCENCF) 202,200,201
201  NCEN1=(NBEAMS+1)*2
      NCEN2=NWDS+1
      NCEN3=2*NWDS
      NCEN4=NWDS+NCEN1
      NCEN5=2*NWDS+NCEN1
      NFIN=NCEN3-NBEAMS
      NCENCF=-NCENCF
      II=NCEN2
202  I=0
      II=II-1
203  II=II+1
      I=I+1
      DATA(II)=DATA(I)
      IF (I .GT. NWDS) GO TO 1
      IF (II .EQ. NCEN5) GO TO 204
      GO TO 203
204  II=NCEN2-1
154  II=II+1
      DO 206 J=1,N0BEAM
      IV=II+N*FICH(J)
      NCEN7=IV+NCEN1
      DATA(IV)=DATA(NCEN7)-DATA(IV)
206  CONTINUE
      II=II+NBEAMS
      IF (II .GE. NCEN3) GO TO 205
      GO TO 154
205  CONTINUE
      IF (N0EBUG .LE. 1) GO TO 2001
      WRITE (6,1114)
1114  FORMAT (20H)CENTRAL DIFFERENCES)
      WRITE (6,1121)
      DO 2012 IX=1,NBLK
      IY=IX*(NBEAMS+1)+NWDS
      IZ=IY-NBEAMS
      WRITE (6,1112) (DATA(II),II=IZ,IY)
2012  CONTINUE
1115  FORMAT (1H ,F10.5)
2001  CONTINUE
      WRITE (10) (DATA(II),II=NCEN2,NCEN3)
      SP0T=DATA(NFIN)
      NCEN6=NCEN3
      DO 207 II=NCEN2,NCEN4
      NCEN6=NCEN6+1
      DATA(II)=DATA(NCEN6)
207  CONTINUE
      II=NCEN4
      GO TO 203
200  CONTINUE
      IF (N0EBUG .LE. 1) GO TO 2002
      WRITE (6,1121)
      DO 2010 IX=1,NBLK
      IY=IX*(NBEAMS+1)
      IZ=IY-NBEAMS
      WRITE (6,1112) (DATA(I),I=IZ,IY)
2010  CONTINUE
2002  CONTINUE

```

```

      WRITE (10) (DATA(I), I=1, NWCS)
      SP0T=DATA(NFIN)
212  IF (NB00) 1,1,209
209  NC0UNT=NC0UNT+1
      IF (NC0UNT .GE. NB00) G0 T0 210
      G0 T0 1
210  AVRAGE=NT0TN0/(NBEAMS+1)
      D0 211 J=1,NBEAMS
      VVMEAN(J)=VMEAN(J)/AVRAGE
      RRMS(J)=SQRT((RMS(J)/AVRAGE)-VVMEAN(J)**2)/SQRT(AVRAGE)
211  C0NTINUE
      SQRTT=1.0/SQRT(DATA(1))
      WRITE (6,1112) DATA(1),SQRTT,(VVMEAN(J),RRMS(J),J=1,NBEAMS)
      NC0UNT=C
      G0 T2 1
999  C0NTINUE
      ST0P(NSTART)=SP0T
C    FINISH RMS CALCULATIONS
      IF (NDEBUG .EQ. 0) G0 T0 2003
      AVRAGE=NT0TN0/(NBEAMS+1)
      D0 209 J=1,NBEAMS
      VMEAN(J)=VMEAN(J)/AVRAGE
      RMS(J)=SQRT((RMS(J)/AVRAGE)-VMEAN(J)**2)
208  C0NTINUE
      WRITE (6,1112) (VMEAN(J),RMS(J),J=1,NBEAMS)
2003 C0NTINUE
      WRITE (6,1112) (SLOPE(I),BIAS(I),I=1,NBEAMS)
1112 F0RMAT (1H 12F10.6)
      WRITE (6,1120) (START(I),ST0P(I),I=1,NSTART)
1120 F0RMAT (44H START AND ST0P TIMES 0N LINK 0NF 0UTPUT ARE, 2F12.6)
      CALL 0LT1(6,EXPNO,SC(2),SC(1),N0PAGE,SC(3),SC(4),N0RUN)
      CALL 0LT2(6,NBEAMS,SC(5),SC(6),AVGMIN,AVGMAX,SLOPF,BIAS,VALMIN,
1VALMAX)
      CALL 0LT3(6,NBEAMS,SC(43),SC(44),IFREQ,MVMEAN,NCENDF,SC(8),
1SC(5),SC(6),NWHICH,N0HEAM)
      TIFREQ=IFREQ
      TIMSAM=TIMSAM*TIFREQ
      END FILE 10
      REWIND 10
      REWIND 9
      RETURN
      END

```

DECK000C

84

```

      TIMEFAC=FLOAT(NCOUNT*NBLK)*TIMSAM
      II=0
      IF (NW-NWDS) 7,6,7
7     CONTINUE
6     CONTINUE
      GO TO (5,3,4,5,3),IERR
3     WRITE (6,99)
99    FORMAT (27H0END OF FILE ON INPUT TAPE )
      GO TO 999
98    FORMAT (42H0IRRECOVERABLE PARITY ERROR ON TAPE      )
4     WRITE (6,98)
      NPAR=NPAR+1
      IF (NPAR .GE. 10) GO TO 999
      GO TO 1
C     TERMINATE RUN HERE
C     TRANSLATE TIME
55    I=C
5     CONTINUE
      IF (NFLAG .EQ. 0) GO TO 110
      IF (DATI .LT. START(ISL)) I=I-1
110   I=I+1
400   II=II+1
      IF=II+2
      LW=IF/3
      IP=IF-3*(LW)
      IBT=12*IP
      IDAT=0
      CALL FLD(IDAT,24,12,IBT,IDATA(LW))
      DATI=IDAT
      DATI=DATI+TIMSAM+TIMEFAC
      IF (NFLAG) 11,12,11
11    IF (DATI .LT. START(ISL)) GO TO 8
      IF (DATI .GT. STOP(ISL)) GO TO 10
      GO TO 12
8     II=II+NBEAMS
      IF (II .GE. NWDS) GO TO 1
      GO TO 400
10    IF (ISL .GE. NSTART) GO TO 407
      ISL=ISL+1
      GO TO 8
407   IERR=2
      GO TO 999
12    CONTINUE
      DATA(I)=DATI
C     TRANSLATE DATA FROM WORD
      DO 198 J=1,NBEAMS
      I=I+1
      II=II+1
      IF=II+2
      LW=IF/3
      IP=IF-3*(LW)
      IBT=12*IP
      IDAT=0
      CALL FLD(IDAT,24,12,IBT,IDATA(LW))
      ZZI=AND(IMK(3),IDAT)
      IF (ZZI .EQ. 0) GO TO 500
      IDAT=IDAT-NCZM
500   DATA(I)=IDAT
198   CONTINUE
      IF (II-I) 401,402,403
402   IF (II .GE. NWDS) GO TO 404

```

```

      G0 T0 110
403 IF (II .GE. NWDS) G0 T0 405
      G0 T0 110
401 IF (I .GE. NWDS) G0 T0 406
      G0 T0 110
404 NFLAG1=0
      G0 T0 111
405 NFLAG1=-1
      G0 T0 111
406 NFLAG1=1
      G0 T0 111
111 C0NTINUE
      IF (NFLAG .EQ. 1) G0 T0 999
      IF (NCAL) 130,120,129
120 C0NTINUE
      WRITE (6,1121)
      D0 2010 IX=1,NBLK
      IY=IX*(NBEAMS+1)
      IZ=IY-NBEAMS
      WRITE (6,1112) (DATA(I),I=IZ,IY)
2010 C0NTINUE
      D0 199 I=1,NBEAMS
      AVGMAX(J)=0.0
      AVGMIN(J)=0.0
199 C0NTINUE
      I=0
201 I=I+1
      D0 200 J=1,NBEAMS
      I=I+1
      AVGMIN(J)=AVGMIN(J)+DATA(I)
200 C0NTINUE
      IF (I .GE. NWDS) G0 T0 207
      G0 T0 201
207 C0NTINUE
      DVISOR=NWDS/(NBEAMS+1)
      D0 202 J=1,NBEAMS
      AVGMIN(J)=AVGMIN(J)/DVISOR
202 C0NTINUE
      NCAL=1
      I=0
      G0 T0 1
129 C0NTINUE
      D0 2011 IX=1,NBLK
      IY=IX*(NBEAMS+1)
      IZ=IY-NBEAMS
      WRITE (6,1112) (DATA(I),I=IZ,IY)
2011 C0NTINUE
      I=0
203 I=I+1
      D0 204 J=1,NBEAMS
      I=I+1
      AVGMAX(J)=AVGMAX(J)+DATA(I)
204 C0NTINUE
      IF (I .GE. NWDS) G0 T0 208
      G0 T0 203
208 C0NTINUE
      D0 205 J=1,NBEAMS
      AVGMAX(J)=AVGMAX(J)/DVISOR
205 C0NTINUE
      D0 206 J=1,NBEAMS
      SLOPE(J)=(VALMAX(J)-VALMIN(J))/(AVGMAX(J)-AVGMIN(J))

```

```

BIAS(J)=VALMAX(J)-AVGMAX(J)*SLOPE(J)
206  CONTINUE
      NCAL=-1
      I=1
      NCOUNT=-1
      DATI=-DATI
      NFLAG=1
      GO TO 1
130  CONTINUE
999  RETURN
      END

```

\$J4FTC DECK1B

DECK0000

```

      SUBROUTINE UNPAK
      COMMON START(5),STOP(5),SC(53),VALMIN(12),VALMAX(12)
      1, SLOPE(12),BIAS(12),LTN,NBEAMS,NFLSKC,NFLSKC,MVMEAN,
      2TIMSAM,NWDS,IFREQ,MSFC,NFILE,RMS(12),VMEAN(12)
C      READ 10 IN BCD MODE
      IERR=0
      CALL RECTPR (LTN,3,IFRR,NW,53,SC)
C      SC(1) THRU SC(6+N) ARE ALPHANUMERIC FIELD DATA TO BE
C      WRITTEN OUT A FORMATED
C      SC(6+N THRU SC(8+3N) ARE BCD TO BE CONVERTED TO FLOATING POINT
      N=19
      NN=53
      DO 10 I=N,NN
      CALL TRNLAT(SC(I))
10  CONTINUE
      RETURN
      END

```

\$IBFTC DECK1C

DECK0300

```

SUBROUTINE TRNLAT(SC)
  DIMENSION N(6),NTBL(14),NN(6)
  DATA NTBL(1)/001/
  DATA NTBL(2)/002/
  DATA NTBL(3)/003/
  DATA NTBL(4)/004/
  DATA NTBL(5)/005/
  DATA NTBL(6)/006/
  DATA NTBL(7)/007/
  DATA NTBL(8)/010/
  DATA NTBL(9)/011/
  DATA NTBL(10)/000/
  DATA NTBL(11)/060/
  DATA NTBL(12)/040/
  DATA NTBL(13)/020/
  DATA NTBL(14)/033/
  NFLAG=1
  DO 50 I=1,6
    NN(I)=C
    N(I)=C
50  CONTINUE
    CALL FLC (N(1),30,6,30,SC)
    CALL FLC (N(2),30,6,24,SC)
    CALL FLC (N(3),30,6,18,SC)
    CALL FLC (N(4),30,6,12,SC)
    CALL FLC (N(5),30,6,6,SC)
    CALL FLC (N(6),30,6,0,SC)
    NSCL=0
    DO 100 J=1,6
      IF (N(J) .EQ. NTBL(14)) NSCL=J-1
      IF (N(J) .EQ. NTBL(12)) NFLAG=-NFLAG
100  CONTINUE
      IF (NSCL .GT. 0) GO TO 200
      TSCL=1.0
      GO TO 201
200  TSCL=10**NSCL
201  CONTINUE
      JJ=1
      DO 110 J=1,6
        DO 109 K=1,9
          IF (N(J) .EQ. NTBL(K)) NN(JJ)=K
109  CONTINUE
          IF (N(J) .EQ. NTBL(10)) NN(JJ)=C
          IF (N(J) .EQ. NTBL(11)) NN(JJ)=C
          IF (N(J) .EQ. NTBL(12)) NN(JJ)=C
          IF (N(J) .EQ. NTBL(13)) NN(JJ)=C
          IF (N(J) .EQ. NTBL(14)) JJ=JJ-1
          JJ=JJ+1
110  CONTINUE
          NN(2)=NN(2)*10
          NN(3)=NN(3)*100
          NN(4)=NN(4)*1000
          NN(5)=NN(5)*10000
          NN(6)=NN(6)*100000
          NDUPPY=NN(1)+NN(2)+NN(3)+NN(4)+NN(5)+NN(6)
          SC=NDUPPY
          SC=(SC/TSCL)*FL2AT(NFLAG)
          RETURN
        END

```

\$1BFTC DECK1D

DECK0000

```

      SUBROUTINE VM0VMN (DATA,ITEMP,TIMC0R,NCLSPN,INDEX,AVRAGE,
      INWHICH,N0BEAM)
      COMMON START(5),STOP(5),SC(53),VALMIN(12),VALMAX(12)
      1,SL0PE(12),BIAS(12),LTN,NBEAMS,NFLSKC,NFLSKC,MVMEAN,
      2TIMSAM,NWDS,IFREQ,MSFC,NFILE,RMS(12),VMEAN(12)
      DIMENSION NWHICH(1)
      DIMENSION SUM(12)
      DIMENSION DATA(1)
      LSPAN=NCLSPN+NWDS
      I=INDEX-1
111   I=I+1
C     CALCULATE TIME
      DATA(I)=DATA(I)+TIMC0R
      M=I
      DO 102 J=1,NBEAMS
      SUM(J)=0.0
102   CONTINUE
103   M=M+1
      DO 104 J=1,N0BEAM
      MM=M+NWHICH(J)-1
      JJ=NWHICH(J)
      SUM(JJ)=SUM(JJ)+DATA(MM)
104   CONTINUE
      M=M+NBEAMS
      IF (M .GE. LSPAN) GO TO 112
      GO TO 103
112   CONTINUE
      LSPAN=LSPAN+NBEAMS+1
      DO 110 J=1,N0BEAM
      II=I+NWHICH(J)
      JJ=NWHICH(J)
      DATA(II)=SUM(JJ)/AVRAGE
110   CONTINUE
      I=I+NBEAMS
      IF (I .GE. ITEMP) GO TO 113
      GO TO 111
113   CONTINUE
      RETURN
      END

```

DECK0000

90

```

      IF (NCAL .EQ. 0) NDATA=NWCS
      IF (NCAL .GE. 1) NDATA=NDWS
      IF (NDEBUG .EQ. 0) GO TO 200C
      WRITE (6,1112) NBEAMS,NSCANS,NDWS1,NDWS,NCAL,NWCS,IERR,NSCALE
200C  CONTINUE
C     READ DATA
      GO TO 1
      8  CONTINUE
C     READ DATA WORD AND PLACE IN DATA(1) ARRAY
      ITIME=0
C     TRANSLATE TIME
      CALL FLD (ITIME,12,24,C,1DATA)
      IF (ITIME-ITIME1) 2010,2011,2011
2010  NCTR=NCTR+1
      NCYL=1000000*NCTR
2011  ITIME1=ITIME
      RTIME=ITIME+NCYL
      RTIME=(RTIME/TIMSCL)/PLYBCK
      IF (NDEBUG .EQ. 0) GO TO 2001
      WRITE (6,1113) RTIME
2001  CONTINUE
      RTIME=RTIME-TIMSAM
114   RTIME=RTIME+TIMSAM
      IF (IRECFG) 17,18,17
17    IF (RTIME .LT. START(ISL)) GO TO 20C
      IF (RTIME .GT. STOP(ISL)) GO TO 201
      GO TO 18
20C   II=II+NBEAMS
      NCOUNT=NCOUNT+NBEAMS
      IF (NCOUNT .GE. NDWS1) GO TO 1
      GO TO 114
201   IF (ISL .GE. NSTART) GO TO 202
      ISL=ISL+1
      GO TO 20C
202   IERR=2
      GO TO 999
18    DATA(1)=RTIME
      I=I+1
      DO 100 J=1,NBEAMS
      II=II+1
      LW=II/3+1
      IP=II-3*(LW-1)
      IBT=12*IP+2
      IDAT=0
      CALL FLD (IDAT,26,10,IBT,1DATA(LW))
      DATA(1)=IDAT
      ZZI=AND(IMK(IP+1),1DATA(LW))
      IF (ZZI .NE. 0) DATA(1)=-DATA(1)
      I=I+1
100   CONTINUE
      NCOUNT=NCOUNT+NBEAMS
      IF (I .GE. NDATA) GO TO 116
      IF (NCOUNT .GE. NDWS1) GO TO 1
      GO TO 114
116   IF (NCOUNT .LT. NDWS1) GO TO 118
      IF (NCAL .GE. 1) GO TO 11C
      IRECFG=1
      GO TO 11C
118   IF (NCAL .GE. 1) GO TO 11C
      IRECFG=-1
11C   CONTINUE

```

```

      IF (NCAL-1) 14,13,21
13    CALL SINCAL (DATA,NDATA,NSTEP,IERR)
      GO TO (1,22,22,1,22),IERR
21    CALL STPCAL (DATA,NDATA)
22    IRECFG=1
      ITIME1=0
      NCTR=0
      NCYL=0
      I=1
      GO TO 15
14    CONTINUE
999   RETURN
      END

```

```

$IBFTC DECK1F
      FUNCTION INTBCC(NDUMMY)
      N1=0
      N2=0
      N3=0
      N4=0
      N5=0
      N6=0
      CALL FLC (N1,30,6,30,NDUMMY)
      CALL FLC (N2,30,6,24,NDUMMY)
      CALL FLC (N3,30,6,18,NDUMMY)
      CALL FLC (N4,30,6,12,NDUMMY)
      CALL FLC (N5,30,6,6,NDUMMY)
      CALL FLC (N6,30,6,0,NDUMMY)
      IF (N1 .EQ. 48) N1=0
      IF (N2 .EQ. 48) N2=0
      IF (N3 .EQ. 48) N3=0
      IF (N4 .EQ. 48) N4=0
      IF (N5 .EQ. 48) N5=0
      IF (N6 .EQ. 48) N6=0
      N2=N2*10
      N3=N3*100
      N4=N4*1000
      N5=N5*10000
      N6=N6*100000
      N=N1+N2+N3+N4+N5+N6
      INTBCC=N
      RETURN
      END

```

DECK000C

```

SUBROUTINE SINCAL (DATA,NDATA,NSTEP,IERR)
COMMON START(5),STOP(5),SC(53),VALMIN(12),VALMAX(12)
1,SLØPE(12),BIAS(12),LTN,NBEAMS,NFLSKC,NFLSKC,MVMEAN,
2TIMSAM,NWDS,IFREQ,MSEC,NFILE,RMS(12),VMEAN(12),NØRUN,EXPNO,NØPAGE,
4AVGMIN(12),AVGMAX(12),NBLK,NSTART
DIMENSION DATA(1)
DIMENSION CONST(12)
DOUBLE PRECISION XMEAN(12)
DOUBLE PRECISION SUM(12)
DATA NTØTAL/0/
DATA NFLAG/0/
IF (NFLAG .GT. 0) GØ TØ 121
DØ 122 IJ=1,12
XMEAN(IJ)=0.0
SUM(IJ)=0.0
122 CONTINUE
NFLAG=1
121 CONTINUE
GØ TØ (141,120,120,141,120),IERR
141 CONTINUE
IF (NSTEP) 109,108,109
108 FACTØR=1.414215
FACTR1=2.0
GØ TØ 110
109 FACTØR=1.0
FACTR1=1.0
110 CONTINUE
I=C
103 I=I+1
DØ 104 J=1,NBEAMS
I=I+1
XMEAN(J)=XMEAN(J)+DATA(I)
SUM(J)=SUM(J)+DATA(I)**2
104 CONTINUE
NTØTAL=NTØTAL+1
IF (I .GE. NDATA) GØ TØ 107
GØ TØ 103
107 CONTINUE
RETURN
120 CONTINUE
TØTAL=NTØTAL
DØ 105 J=1,NBEAMS
CONST(J)=(SUM(J)/TØTAL)-((XMEAN(J)/TØTAL)**2)
CONST(J)=SQRT(CONST(J))*FACTØR
SLØPE(J)=(VALMAX(J)-VALMIN(J))/(FACTR1*CONST(J))
BIAS(J)=(-SLØPE(J)*CONST(J))+VALMAX(J)
AVGMIN(J)=-CONST(J)+BIAS(J)/SLØPE(J)
AVGMAX(J)=CONST(J)+BIAS(J)/SLØPE(J)
105 CONTINUE
RETURN
END

```

\$IBFTC DECK1H

DECK0000

```

      SUBROUTINE STPCAL (DATA,NCDATA)
      COMMON START(5),STOP(5),SC(53),VALMIN(12),VALMAX(12)
      1,SLØPE(12),BIAS(12),LIM,NBEAMS,NFLSKC,NFLSKC,MVMEAN,
      2TIMSAM,NWDS,IFREQ,MSEC,NFILE,RMS(12),VMEAN(12),NØRUN,EXPNO,NØPAGE,
      4AVGMIN(12),AVGMAX(12),NELK,NSTART
      DIMENSION DATA(1)
      DØ 10 J=1,NBEAMS
      AVGMIN(J)=0.0
      AVGMAX(J)=0.0
10    CONTINUE
      I=1
      NCØUNT=0
      NBAD=0
11    I=I+1
      TEMP=DATA(I)
17    II=I+NBEAMS+1
      NTEST=ABS(DATA(II)-TEMP)
      IF (NTEST .GT. 100) GØ TØ 15
      NCØUNT=NCØUNT+1
      DØ 30 J=1,NBEAMS
      AVGMIN(J)=AVGMIN(J)+DATA(I)
      I=I+1
30    CONTINUE
      GØ TØ 11
15    NBAD=NBAD+1
      IF (NBAD .EQ. 10) GØ TØ 50
      I=I+NBEAMS+1
      GØ TØ 17
50    CONTINUE
      DIVIDE=NCØUNT
      DØ 60 J=1,NBEAMS
      AVGMIN(J)=AVGMIN(J)/DIVIDE
60    CONTINUE
      NCØUNT=0
      NBAD=0
      I=I-1
12    I=I+1
      TEMP=DATA(I)
18    II=I+NBEAMS+1
      NTEST=ABS(DATA(II)-TEMP)
      IF (NTEST .GT. 100) GØ TØ 16
      NCØUNT=NCØUNT+1
      DØ 31 J=1,NBEAMS
      AVGMAX(J)=AVGMAX(J)+DATA(I)
      I=I+1
31    CONTINUE
      IF (I .GE. NDATA) GØ TØ 51
      GØ TØ 12
16    NBAD=NBAD+1
      IF (NBAD .EQ. 10) GØ TØ 51
      I=I+NBEAMS+1
      GØ TØ 18
51    CONTINUE
      DIVIDE=NCØUNT
      DØ 61 J=1,NBEAMS
      AVGMAX(J)=AVGMAX(J)/DIVIDE
61    CONTINUE
      DØ 62 J=1,NBEAMS
      SLØPE(J)=(VALMAX(J)-VALMIN(J))/(AVGMAX(J)-AVGMIN(J))
      BIAS(J)=VALMAX(J)-AVGMAX(J)*SLØPE(J)

```

```

62  CONTINUE
    RETURN
    END

```

```

$IBFTC DECK1I                                     DECK000J
    SUBROUTINE OUT2(N,NBEAMS,BETIME,ENTIME,VALMIN,VALMAX,SLOPE,
1          XINTER,ENGMIN,ENGMAX)
    DIMENSION VALMIN(1),VALMAX(1),
1          SLOPE(1),XINTER(1),ENGMIN(1),ENGMAX(1)
    WRITE(N,1020)
1020  FORMAT( //1H ,16(1H+),26H CALIBRATION IDENTIFICATION,55(1H+),/50H + FM 1020
2CHANNEL+BEGIN RECORD+END RECORD+MINIMUM+MAXIMUM+,6X,11H CALIBRATION FM 1020
3,6X,1H+,6X,11H ENGINEERING,6X,1H+,/17H + NUMBER+ TIME,5X,7H+ TIM FM 1020
4E,4X,65H+ VALUE + VALUE + SLOPE + INTERCEPT + MINIMUM + MAX FM 1020
5IMUM +,/1H ,97(1H+)) FM 1020
    DO 1 I=1,NBEAMS
1      WRITE(N,1040) I,BETIME,ENTIME,VALMIN(I),VALMAX(I),
1          SLOPE(I),XINTER(I),ENGMIN(I),ENGMAX(I)
1040  FORMAT(4H + ,12,7H + ,A6,6H + ,A6,2H ,2(1H+,F7.1),
14(1H+,F11.5)
2,1H+) FM 1040
    WRITE(N,1060)
1060  FORMAT(1H ,97(1H+)) FM 1060
    RETURN
    END

```

```

$IBFTC DECK1J                                     DECK000C
    SUBROUTINE OUT3(N,NBEAMS,SAMRTE,SSARED,IADRED,IMQVMA,ICDIFF
1          ,CHANVR,BETIME,ENTIME,NWHICH,NBEAM)
    DIMENSION CHANVR(1),NWHICH(1)
    DIMENSION MVMEAN(12),NCENDF(12)
    WRITE(N,1080)
1080  FORMAT( //1H ,15(1H+),36H MODIFICATION OF ORIGINAL TIME SERIES,31(1 FM 1080
2H+),/83H +CHANNEL+SAMPLING+ORIGINAL +ADDITIONAL+MOVING+ CENTRAL + FM 1080
3CHANNEL + BEGIN + END +,/83H + NUMBER+ RATE + SAMPLE + SAMP FM 1080
4LE + MEAN + DIFFERENCE+VARIABLE+RECORD +RECORD +,/2H +,16X,22H+REDU FM 1080
5CTION+ REDUCTION+,6X,1H+,10X,1H+,8X,17H+ TIME + TIME +,/1H ,82(1 FM 1080
6H+)) FM 1080
    DO 2 I=1,NBEAMS
    NCENDF(I)=0
2      MVMEAN(I)=0
    DO 3 I=1,NBEAM
    J=NWHICH(I)
    NCENDF(J)=-ICDIFF
3      MVMEAN(J)=-IMQVMA
    DO 1 I=1,NBEAMS
1      WRITE(N,1100) I,SAMRTE,SSARED,IADRED,MVMEAN(I),
2          NCENDF(I),
1          CHANVR(I),BETIME,ENTIME
1100  FORMAT(4H + ,12,4H + ,F8.2,2H+ ,F8.2,3H+ ,16,3H +,16,3H+
2,16,5H + ,A6,2H+ ,A6,2H+ ,A6,1H+)
    WRITE(N,1120)
1120  FORMAT(1H ,92(1H+)) FM 1120
    RETURN
    END

```

```

$ORIGIN      ALPHA,SYSUT2,REW
$IBFTC LINK22 LIST,N200,DECK
SUBROUTINE LINK2(KRUN,SPRT)
C
C LINK2 -CORRELATION LINK -
C
C ALL BASIC SUMS AND CORRELATIONS ARE PERFORMED
C IN THIS LINK
C
C REGIONS FOR CHANNEL ASSIGNMENTS
C
C INTEGER VELLAG
C DIMENSION NCHID(12),NEWID(12),NCTBL(12, 6),NCRPC(12)
C 1,NACR(12),NCD(12)
C DIMENSION XASTRK(12)
C DIMENSION VLAG(1)
C DIMENSION SEPART(12)
C
C TEMPORARY STORAGE
C
C DIMENSION U(12500),IU(12500),TEMP(512),TLAG(1)
C COMMON START(5),STOP(5),SC(53),VALMIN(12),VALMAX(12)
C 1,SL0PE(12),BIAS(12),LTN,NBEAMS,NFLSKC,NFLSKC,MVMEAN,
C 2TIMSAM,NWDS,IFREC,MSFC,NFILE,RMS(12),VMEAN(12),ACRUN,EXPNO,N0PAGE,
C 4AVGMIN(12),AVGMAX(12),NBLK,NSTART,
C 3C0PTIM
C LOGICAL C0VAR
C DIMENSION C0M(1)
C EQUIVALENCE (START(1),C0M(1))
C EQUIVALENCE (U,IU)
C EQUIVALENCE (TLAG,U)
C EQUIVALENCE (U,VLAG)
C EQUIVALENCE (TIMSAM,ESMPP)
C
C 9000 FORMAT (1H 12I6)
C 9001 FORMAT (1H 10F10.6)
C NUMC0M=175
C NBEAM=12
C SYSTEM PARAMETERS
C
C NAME DESCRIPTION TYPE DIMENSION
C
C MAXSTR - MAXIMUM STORAGE OF DATA PER PL I
C MAXCHN - MAXIMUM NUMBER OF CHANNELS I
C
C MAXCR - MAXIMUM CORRELATIONS PER CHANNEL I
C
C
C MAXSTR=12500
C MAXCHN=12
C MAXCR = 6
C
C INTERMEDIATE TAPE
C
C IRT=11
C IDAT=10
C NFLAG=0
C NBLK=30
C NWPR=NBEAMS+1
C
C PROBLEM PARAMETERS

```

	NAME	DESCRIPTION	TYPE	DIMENSION
C				
C				
C				
C				
	NAMELIST/CVARNC/CZVAR			
	NAMELIST /TIMES/ START ,STOP, DELTAT,IPRINT			
	NAMELIST /DELAY/ TAUMIN,TAUMAX,DELTAU,VELLAG,SEPART			
	READ (IDAT) (CZM(I),I=1,NUMCZM)			
	ESMPP=ESMPP*FLD0AT(IFREQ)			
	STOP(INSTART)=SP2T			
	READ (5,CVARNC)			
	IF (CZVAR) GO TO 180			
	GO TO 1			
180	NFLAG=1			
1	READ(5,TIMES)			
	WRITE(6,TIMES)			
	READ(5,DELAY)			
	WRITE(6,DELAY)			
	WRITE(IRT) (CZM(I),I=1,NUMCZM)			
C				
C	DETERMINE TIME LAGS			
C				
	NLAYS=(TAUMAX-TAUMIN)/DELTAU +.5			
	NBETA=DELTAU/ESMPP +.5			
	NTLAGS=2*NLAYS+1			
	NT=NLAYS+1			
	IIT=NLAYS+1			
	MMIT=NLAYS+1			
	TLAY(IIT)=TAUMIN			
	DO 2 I=2,NT			
	IT=IIT+I-1			
	MIT=MMIT-I+1			
	FI=I-1			
	TLAY(IT)=TAUMIN+FI*DELTAU			
2	TLAY(MIT)=-TLAY(IT)			
C				
C	DETERMINE CONSTANTS FOR STORAGE			
C				
	MAXINC=TAUMAX/ESMPP+.5			
	MSUB=DELTAU/ESMPP+.5			
C				
C	DETERMINE STORAGE REQUIREMENTS			
C	AND SET INDEX NUMBERS			
	CALL ASSIGN(NBEAM ,MAXCR,TEMP,NCHIC,NEWID,NCTBL,NCRPC,NACR			
	1,NCH,ND,NU,NAUT0,NCRT,NCD)			
	CALL CH2LT(ND,NCHID,NCRPC,NCTBL,NBEAM ,MAXCR,NCD)			
94	IF (VELLAG) 92,91,92			
91	MISC1=1			
	MISC2=1			
	GO TO 3			
97	MISC1=NCRT+1			
	MISC2=MISC1			
3	MISC=2*MISC1*NLAYS+MISC2			
	KP=MISC+9*NCH+NCRT*(NTLAGS+1)			
	NUMSP=MAXSTR-KP			
C	CHECK STORAGE			
C				
	MAXCIA=MAXINC*NCH			
	NPT=(NUMSP-MAXINC*NCH)/(MSUB*NCH)			

```

      IF(NPT.GE.1) GO TO 8
      CALL ERR2R( 1,24H STORAGE EXCEEDED           KP,NUMSP)
      ND=ND-1
      IF(ND.LT.1) GO TO 7
      NCRT=NACR(ND+1)
      GO TO 94
7 CALL ERR2R( 1,24H NET SPECIFIED CORRECTLY,NCRT,KP)
  CALL EXIT
8 CONTINUE
  CALL CH2LT(ND,NCHID,NCRPC,NCTBL,NBEAM,MAXCH,NCD)
  WRITE(IRT) NCH,ND,NU,NAUT2,NCRT,NCHID,NEWIC,NCTBL,NCRPC,NACH
C
C
C   DETERMINE VELOCITY LAGS
C
      IF (VELLAG) 93,86,93
93 CONTINUE
      IIV=2*NLAGS
      DO 85 JJ=1,NCRT
      IIV=IIV+2*(NLAGS)*(JJ-1)+JJ
      DO 84 I=1,NLAGS
      IV=IIV+I
      IF (I-NLAGS-1) 81,82,81
82 VLAG(IV)=20.0
      GO TO 84
81 VLAG(IV)=SEPART(JJ)/TLAG(I)
84 CONTINUE
85 CONTINUE
86 CONTINUE
C
      WRITE (IRT) NCH,ND,NU,NAUT2,NCRT,NCHID,NEWIC,NCTBL,NCRPC,NACH,NCD
      WRITE (IRT) DELTAT,TAUMIN,TAUMAX,DELTAU,NLAGS,NLAGS,VELLAG
      WRITE (IRT) (U(I),I=1,MISC)
C      INITIALIZE FOR LOOP
      IPR=I
      IP=1
      KS=KP
      KESM=MISC+1
      KEAM=KESM+NCH
      KEASM=KEAM+NCH
      KASM=KEASM+NCH
      KMA=KASM+NCH
      KRA=KMA+NCH
      KPSM=KRA+NCH
      KMP=KPSM+NCH
      KRP=KMP+NCH
      KTP=KRP+NCH
C      CLEAR AREA
      NST=MISC+1
      DO 30 I=NST,KS
30 U(I)=0.0
C
      KK=KS+1
C      MAIN LOOP
C
C      SET NUMBER OF PIECES
C
60 TP=ST2P(IP)-START(IP)
   KNP=1
   NF =TP/DELTAT +.5
   NUMST=TP/ESMPP +.5
C

```

```

C      CHECK SPECIFICATIONS
C
NDIF =NUMST-NF*MSUB
IF(NDIF) 10,11,12
10 CALL ERROR( 1,24H TIME SPAN AND PIECE OFF ,NUMST,NDIF)
NUMST=(NF-1)*MSUB
GO TO 11
12 CALL ERROR( 1,24H TIME SPAN .GTR. PIECES ,NUMST,NDIF)
NUMST=NF*MSUB
11 CONTINUE
NTOT=(STOT(IP)-START(IP))/ESMPP+.5
KD=KK
NPP=NPT
NF=(NTOT-MAXINC)/MSUB
IF(NF.LT.NPP) NPP=NF
NUMS=NPP*MSUB+MAXINC
IF (NF .GE. 1) GO TO 20
WRITE (6,9002) NF,NUMS,SPOT
9002 FORMAT (36H INSUFFICIENT DATA IN TIME INTERVAL. ,2I6,F10.5)
GO TO 25
20 CALL READR(U(KD),NUMS,NCH,NCHID,INC,IDAT,NBLK,NWPR,TEMP,START(IP),
1 TEND)
KPP=1
21 CONTINUE
CALL ACCPC(U(KMP),U(KRP),U(KPSM),U(KK),MSUB,NCH)
IF (NFLAG) 181,181,182
182 CALL REMOVE (U(KMP),U(KK),MSUB,NCH)
181 CONTINUE
DO 17 J=1,ND
L=NCD(J)
NT=NCRPC(J)
DO 170 K=1,NT
KR=KTP+(K-1+NACR(J))*(NTLAGS+1)
NCU=NCTBL(J,K)
CALL CORLAT(U(KK),U(KK),NLAGS,TLAG(NLAGS+1),ESMPP,L,NCU,NCH,MSUB,
1U(KR))
WRITE (6,9000) KP,MISC,KK,KR,NT,J,NCRPC(J)
WRITE (6,9001) L(KR)
IA=KR+NTLAGS
DO 50 I=1,NTLAGS
II=KR+I-1
50 U(IA)=U(IA)+U(II)
U(IA)=U(IA)/FLZAT(NTLAGS)
IF (L .NE. NCU) GO TO 185
CALL XKRAUS (U(KMP),U(KRP),U(KPSM),U(IA),J,XASTRK,NCH)
185 CONTINUE
170 CONTINUE
17 CONTINUE
C      ACCUMULATE PIECES
NT=3*NCH
CALL PCRC1 (U(KASM),U(KPSM),NT,KNP)
IF(KNP.EQ. 1) GO TO 71
CALL STDER1 (U(KESM),U(KASM),U(KPSM),NT,KNP)
71 CONTINUE
C
C      TEST FOR OUTPUT
C
IF(IPRINT.EQ. 0) GO TO 40
IF (IPR-IPRINT) 31,183,183
183 IPR=0
TIM=FLZAT(KNP*MSUB)*ESMPP

```

```

PCELNG=FL0AT(MSUB)*ESMPF
CALL 0UT1(6,EXPNO,SC(2),SC(1),N0PAGE,SC(3),SC(4),N0RUN)
CALL 0UT5 (6,NCH,KNP,KNP,START(1),TIM,PCELAG,NCHID,U(KMP),
1U(KRP),U(KMA),U(KEAM),U(KRA),U(KEASP))
CALL 0UT5A (6,NCH,KNP,KNP,START(1),TIM,PCELAG,NCHID,XASTRK,
1U(KPSM),U(KMA),U(KEAM),U(KASM),U(KESM))
CALL 0UT1(6,EXPNO,SC(2),SC(1),N0PAGE,SC(3),SC(4),N0RUN)
NNN=2*(NLAGS+1)
D0 32 J=1,ND
NT=NCRPC(J)
D0 132 K=1,NT
NCU=NCTBL(J,K)
KM=(K-1+NACR(J))*(NLAGS+1)
KR=KTP+KM
KAA=KR
KLA=KR+NLAGS
AREA=U(KLA)
NN=NCD(J)
N=NCHID(NN)
NUP=NCHID(NCU)
CALL 0U6A(6,NLAGS,KNP,KNP,AREA,AREA,START(1),TIM,N,NUP,PCELAG,TLAG
1,VLAG(NNN),U(KR),VELLAG)
NNN=NNN+2*NLAGS+1
C WRITE (6,9000) J,NT,KM,KR,KLA
C WRITE (6,9001) U(KR)
132 CONTINUE
32 CONTINUE
31 IPR=IPR+1
40 CONTINUE
WRITE (IRT) KNP,(U(NN),NN=KESM,KS)
C
C TEST FOR END
C
IF(KNP.EC.NF) G0 T0 25
KNP=KNP+1
IF(KPP.GE.NPP) G0 T0 23
KPP=KPP+1
KK=KK+MSLB*NCH
G0 T0 21
23 NR =NF-KNP+1
IF(NR.LT.NPT) NPP=NR
NXX1=MSUB*NCH
D0 24 I=1,MAXCIN
KI=KK+NXX1+I-1
KX=KS+I
24 U(KX)=U(KI)
KD=KS+1+MAXCIN
NUMS=NPP*MSUB
KK=KS+1
G0 T0 20
25 CONTINUE
IF(IP.GE.NSTART) G0 T0 100
IP=IP+1
G0 T0 60
100 REWIND IDAT
END FILE IRT
REWIND IRT
RETURN
END

```

\$IBFTC DECK2A

DECK0000

```

      SUBROUTINE ASSIGN(NBEAMS,MAXCR,K,NCHID,NEWID,NCTBL,NCRPC,NACR,
1 NCH,ND,NU,NAUT0,NCRT,ACD)
      SUBROUTINE READS IN CORRELATION SPECIFICATIONS
      J,K1,K2,ETC
      FORMS TABLES AND REASSIGNS CHANNEL NUMBERS
      FOR INTERNAL USE
      WRITTEN BY A. WACHOWSKI
C INPUT
C   NAME      DESCRIPTION      TYPE  DIMENSION
C   NBEAMS= MAX. NO. OF CHANNELS      I
C   MAXCR = MAX. NO. OF CORRELATIONS/CHANNEL      I
C   K      = TEMP STORAGE      I      (NBEAMS)
C OUTPUT
C   NCHID = OLD CHANNEL NUMBERS IN NEW ORDER      I      (NBEAMS)
C   NEWID = NEW CHANNEL NUMBERS IN OLD ORDER      I      (NBEAMS)
C   NCTBL = CORRELATION TABLE      I      (NBEAMS,MAXCR)
C   NCRPC = NO. CORR. PER CHANNEL      I      (NBEAMS)
C   NACR  = ACCUM. NO. OF CORR.      I      (NBEAMS)
C   NCH   = NO. OF CHANNELS USED      I
C   ND    = NO. OF DELAYED CHANNELS      I
C   NU    = NO. OF UNDELAYED CHANNELS      I
C   NAUT0 = NO. OF AUTO CORRELATIONS      I
C   NCRT  = TOTAL NO OF CORRELATIONS      I
      DIMENSION K(6) ,NCTBL(NBEAMS,MAXCR),NCHID(NBEAMS)
1,NEWID(NBEAMS),NCRPC(NBEAMS),NACR(NBEAMS),NCC(NBEAMS)
      NAMELIST/CORR/ J,K
C   INITIALIZE
      NCH=0
      NAUT0=0
      DO 1 I=1,NBEAMS
        NCHID(I)=0
        NEWID(I)=0
        NCRPC(I)=0
        NACR (I)=0
        NCC(I)=0
      DO 1 L=1,MAXCR
        NCTBL(I,L)=0
1  CONTINUE
2  READ(5,CORR)
      WRITE (6,CORR)
      IF(J.EQ.0) GO TO 10
      IF(NCH.NE.0) GO TO 17
      NCHID(1)=J
      NCH=1
17  DO 6 I=1,NCH
        IF(NCHID(I).EQ.J) GO TO 7
6  CONTINUE
      NCH=NCH+1
      NCHID(NCH)=J
7  CONTINUE
      DO 5 L=1,MAXCR
        IF(K(L).EQ.0) GO TO 15
5  CONTINUE
      LM=MAXCR
      GO TO 20
15  LM=L-1
20  CONTINUE
      DO 3 M=1,LM
        KT=K(M)
        IF(KT.EQ.J) NAUT0=NAUT0+1

```

```

      D0 4 L=1,NCH
      IF(KT.EQ.NCHID(L)) G0 T0 3
4  CONTINUE
      NCH=NCH+1
      NCHID(NCH)=KT
3  CONTINUE
      CALL SORT(K,LM,1,1)
      II=1
      D0 9 L=1,LM
      IF(K(L).NE.K(L+1)) G0 T0 8
      CALL ERR0R(1,24H DUPLICATE CHANNELS K,J,K(L))
      LM=LM-1
      G0 T0 9
8  NCTBL(J,II)=K(L)
      II=II+1
9  CONTINUE
      NCRPC(J)=LM
      G0 T0 2
10 CALL SORT(NCHID,NCH,1,1)
      D0 11 I=1,NCH
      KT=NCHID(I)
11 NEWID(KT)=I
      N=1
      D0 12 I=1,NBEAMS
      LM=NCRPC(I)
      IF(LM.EQ.0) G0 T0 12
      NCD(N)=NEWID(I)
      D0 13 L=1,LM
      NC=NCTBL(I,L)
13 NCTBL(N,L)=NEWID(NC)
      NCRPC(N)=LM
      N=N+1
12 CONTINUE
      ND=N-1
      NU=NCH-NC
      D0 14 I=1,ND
14 NAGR(I+1)=NAGR(I)+NCRPC(I)
      NCRT=NAGR(ND+1)
      RETURN
      END

```

```

$IBFTC DECK2B
SUBROUTINE SORT(IARRAY,N1,N2,N3)
DIMENSION IARRAY(1)
LOGICAL CHECK
M=(N1-1)*N2
M1=2*N2
1 CHECK=.FALSE.
DO 2 I=1,2
I1=(I-1)*N2+N3
IF(I1.GT.M) GO TO 2
DO 4 J=I1,M,M1
J1=J+N2
IF(IARRAY(J).LE.IARRAY(J1)) GO TO 4
DO 3 K=1,N2
J2=J-N3+K
J3=J2+N2
ITEMP=IARRAY(J2)
IARRAY(J2)=IARRAY(J3)
3 IARRAY(J3)=ITEMP
CHECK=.TRUE.
4 CONTINUE
2 CONTINUE
IF(CHECK) GO TO 1
RETURN
END

```

```

DECK0000
SORT0010
SORT0020
SORT0030
SORT0040
SORT0050
SORT0060
SORT0070
SORT0080
SORT0081
SORT0090
SORT0100
SORT0110
SORT0120
SORT0130
SORT0140
SORT0150
SORT0160
SORT0170
SORT0180
SORT0181
SORT0190
SORT0200
SORT0210
SORT0220

```

DECK0000

CCCCCCCCCCCC

102 =

I (NUM)

END

```

$IBFTC DECK2D
      SUBROUTINE ACCPC(S,S2,S3,CATA,N,NC)
C
C      ACCUMULATE SUM AND SUM 2F SQUARES
C      WRITTEN BY  A. WACHOWSKI
C
      DIMENSION S(NC),S2(NC),CATA(NC,N)
      DIMENSION S3(NC)
      XN=N
      DO 1 L=1,NC
      S(L)=0.0
1  S2(L)=0.0
C
      DO 3 J=1,NC
      DO 2 I=1,N
      S(J)=S(J)+CATA(J,I)
2  S2(J)=S2(J)+CATA(J,I)*CATA(J,I)
      S(J)=S(J)/XN
      S2(J)=(S2(J)/XN)
      S3(J)=S2(J)-S(J)**2
3  CONTINUE
C
4  RETURN
      END

```

```

$IBFTC DECK2E
      SUBROUTINE CHOUT(ND,NCHID,NCRPC,NCTEL,NBEAMS,MAXCR,NCC)
C
C      OUTPUT ROUTINE FOR INTERNAL CHANNEL
C      ASSIGNMENTS.
C      WRITTEN BY  A. WACHOWSKI
C
      DIMENSION NCHID(ND),NCRPC(ND),NCTEL(NBEAMS,MAXCR),NCD(ND)
      WRITE(6,1)
1  FORMAT(1H1,10X,20H CHANNEL ASSIGNMENTS / 11X,3HNEW,4X,3HOLD,4X,
1  6HN0. 2F,5X,9HUNDELAYEC/12X,2HIC,5X,2HID,4X,5HCORR.,6X,8HCHANNELS
2/)
      DO 2 I=1,ND
      KN=NCRPC(I)
      N=NCD(I)
      KC=NCHID(N)
      WRITE(6,3) I,KC,KN,(NCTEL(I,J),J=1,KN)
3  FORMAT(12X,12,5X,12,6X,12,5X,12(12,2X))
2  CONTINUE
      RETURN
      END

```

DECK0000

\$IBFTC DECK2F

DECK0000

SUBROUTINE CORLAT(X,Y,NLAGS,TLAG,ESMPP,NCD,ACU,NCH,MSUB,R)

C NAME DESCRIPTION TYPE DIMENSION

C PIECEWISE CORRELATION

C WRITTEN BY A. WACHOWSKI

C

C

C X -ARRAY FOR UNDELAYED CHANNELS R X(NCH,MSUB)

C Y -ARRAY FOR DELAYED CHANNELS R Y(NCH,MSUB)

C NLAGS -NUMBER OF LAGS I

C TLAG -TIME LAG ARRAY R TLAG(NLAGS+1)

C ESMPP -EFFECTIVE SAMPLING PERIOD R

C NCD -DELAYED CHANNEL I

C NCU -UNDELAYED CHANNEL I

C NCH -NUMBER OF CHANNELS I

C MSUB -NUMBER OF SAMPLES I

C R -CORRELATION ARRAY R

C

DIMENSION X(NCH,MSUB),Y(NCH,MSUB),TLAG(NLAGS),R(1)

NTL=NLAGS+1

XMSUB=MSUB

DO 2 L=1,NTL

K1=NLAGS+L

R(K1)=0.0

K2=NLAGS-L+2

R(K2)=0.0

II=TLAG(L)/ESMPP+.5

DO 1 I=1,MSUB

IM=II+I

R(K1)=R(K1)+X(NCD,I)*Y(ACU,IM)

IF(L.EQ. 1) GO TO 1

R(K2)=R(K2)+X(NCU,IM)*Y(ACU,IM)

1 CONTINUE

R(K1)=R(K1)/XMSUB

IF(L.EQ. 1) GO TO 2

R(K2)=R(K2)/XMSUB

2 CONTINUE

RETURN

END

SUBROUTINE READR(U,N,NCH,NCHID,IND,IDAT,NBLK,NWPR,TEMP,STAR-
1 TEND)

C
C SUBROUTINE TO READ A BLOCKED DATA TAPE AND SUPPLY
C SELECTED CHANNELS TO A SERIAL DATA ARRAY U.
C WRITTEN BY A. WACHOWSKI
C
C NAME DESCRIPTION TYPE DIM.
C
C U - DATA ARRAY R (N)
C N - NUMBER OF DATA SAMPLES REQUESTED I
C NCH - NUMBER OF SELECTED CHANNELS I
C NCHID - ARRAY OF CHANNEL NUMBERS - (NCH)
C IND - INDEX OF SAMPLE IN TEMP I
C IDAT - DATA TAPE NO. I
C NBLK - NUMBER OF RECORDS PER BLOCK I
C NWPR - NUMBER OF WORDS PER RECORD I
C TEMP - BUFFER TO HOLD BLOCKED RECORDS V (1)

C
C DIMENSION TEMP(1),NCHID(NCH),U(1)
C DATA INIT/1/
C IF (INIT.EQ.0) GO TO 1
C IND=0
C INIT=0
C NWPB=NBLK*NWPR
C 1 KP=0
C 11 IF (IND.NE.0) GO TO 2
C READ(IDAT) (TEMP(I),I=1,NWPB)
C
C TEST FOR END OF DATA
C
C 8 KT=IND*NWPR
C IF (TEMP(KT+1).LT.START) GO TO 10
C DO 7 I=1,NCH
C KC=NCHID(I)
C KB=KT+KC+1
C KS=KP+I
C U(KS)=TEMP(KB)
C 7 TEND=TEMP(KT+1)
C KP=KP+NCH
C 10 IND=IND+1
C IF (IND.GE.NBLK) IND=0
C NI=KP/NCH
C IF (NI.LT.N) GO TO 11
C IF (IND.GE.NBLK) INIT=1
C RETURN
C END

\$IBFTC DECK2K

DECK0000

SUBROUTINE PCRC1(A,X,N,NP)

C

C

C

C

C

THIS SUBROUTINE PERFORMS THE PIECEWISE
RECURSION FORMULA FOR ACCUMULATIVE AVERAGES
 $A(N+1) = (A(N) * N + X(N+1)) / (N+1)$

DIMENSION A(1),X(1)

XKNP=NP

XKNP1=NP-1

DO 1 I=1,N

1 A(I)=(A(I)*XKNP1+X(I))/XKNP

RETURN

END

\$IBFTC DECK2L

DECK0000

SUBROUTINE STDER1 (E,A,P,N,NP)

C

C

C

C

C

C

C

THIS SUBROUTINE PERFORMS THE RECURSION FORMULA
FOR THE STANDARD ERROR OF THE ACCUMULATIVE AVERAGES

$E(N+1) = E(N) * (N-1) / N + ((A(N+1) - P(N+1) - P(N+1) * 2) * (N+1) / N * 2)$

DIMENSION E(1),A(1),P(1)

IF(NP.EQ.1) GO TO 2

XNP=NP-1

XNPM1=NP-2

XNPP1=NP

DO 1 I=1,N

1 E(I)=E(I)*XNPM1/XNP+((A(I)-P(I))*2)*XNPP1/XNP**2

2 RETURN

END

\$IBFTC DECK2M LIST,NBDD,DECK

SUBROUTINE REMOVE (S,DATA,N,NC)

DIMENSION S(NC),DATA(NC,N)

DO 4 J=1,NC

DO 5 I=1,N

5 DATA(J,I)=DATA(J,I)-S(J)

4 CONTINUE

RETURN

END

```

$IBFTC DCTRND LIST,N000,DECK
      SUBROUTINE XKRAUS (S,S2,S3,RXX,J,XASTRK,NC)
      DIMENSION S(NC),S2(NC),S3(NC),XASTRK(1)
      ASTRIC=RXX-S(J)**2
      XASTRK(J)=SQRT(ABS(ASTRIC))
      IF (ASTRIC .GE. 0.0) G2 TO 9
      XASTRK(J)=-XASTRK(J)
9     CONTINUE
      S3(J)=ABS(S2(J)-RXX)
      RETURN
      END

```

```

$IBFTC DECK2H LIST,N000,DECK
      SUBROUTINE ZUT5(N,NCH,N0PCE,N0ACC,ACCST,ACCTME,PCFLNG,N0CHAN,
1      PMNVAL,PRMS,AMNVAL,STERR1,ARMS,STERR2)
      DIMENSION N0CHAN(1),PMNVAL(1),PRMS(1),AMNVAL(1),STERR1(1),ARMS(1),
1      STERR2(1)
      WRITE(N,1200) N0PCE,N0ACC,ACCST,ACCTME,PCFLNG
1200 FORMAT(1H ,43(1H+),29HAVERAGES WITH ZERO TIME DELAY,44(1H+),/2H +, FM 1200
27X,13HPiece NUMBER=,15,4H +,7X,6(1H+,12X),1H+,/22H +ACCUMLATION FM 1200
3 NUMBER=,15,4H +,7X,4CH+ PIECEWISE + PIECEWISE +ACCLMULATIVE+, FM 1200
412H MEAN ,26H+ACCLMULATIVE+MEAN-SQUARED, 1H+,/22H +ACCUMLAT
412N START=,F8.2,87H+CHANNEL+ MEAN VALUE + ROOT-MEAN + MEAN VALUE
5 + STANCARD + ROOT MEAN + STANCARD +,/22H +ACCLMLLATION TIM
6E=,F8.2,9H+ NUMBER+,12X,
714H+SQLARE-VALUE+,12X,9H+ ERROR.4X,22H+SQLARE VALUE+ ERROR,
84X,1H+,/2H +,7X,13HPiece LENGTH=,F8.2,1H+,7X,6(1H+,12X),1H+,/1H ,1 FM 1200
916(1H+)) FM 1200
      XKNP=N0PCE
      DO 1 I=1,NCH
      DUM1=SQRT(PRMS(I))
      DUM2=SQRT(STERR1(I)/XKNP)
      DUM3=SQRT(ARMS(I))
      DUM4=SQRT(STERR2(I)/XKNP)
1     WRITE(N,1220) N0CHAN(I),PMNVAL(I),CLM1, AMNVAL(I),DUM2,
1     DUM3,DUM4
1220 FORMAT(2H +,28X,3H+ ,12,4H +,6(F12.8,1H+)) FM 1220
      WRITE(N,1230)
1230 FORMAT(1X,116(1H+))
      RETURN
      END

```

```

$IBFTC DECK2J . LIST,N0DC,DECK                                DECK0000
  SUBROUTINE OUT5A(N,NCH,N0PCE,N0ACC,ACCST,ACCTME,PCELNG,N0CHAN,
1      PMNVAL,PRMS,AMNVAL,STERR1,ARMS,STERR2)
  DIMENSION N0CHAN(1),PMNVAL(1),PRMS(1),AMNVAL(1),STERR1(1),ARMS(1),
1      STERR2(1)
  WRITE(N,1200) N0PCE,N0ACC,ACCST,ACCTME,PCELNG
1200  FORMAT(1H ,43(1H+),29HAVERAGES WITH ZERO TIME DELAY,44(1H+),/2H +, FM 1200
27X,13HPiece NUMBER=,15,4H +,7X,6(1H+,12X),1H+,/22H +ACCUMLATION FM 1200
3  NUMBER=,15,4H +,7X,4CH+ PIECEWISE + PIECEWISE +ACCLMULATIVE+, FM 1200
412H MEAN ,26H+ACCUMULATIVE+ STANDARC ,1H+,/22H +ACCUMLAT
410N START=,F8.2,87H+CHANNEL+ D C TRENC + STANDARC + MEAN VALUE
5 + STANDARC + STANDARD + ERR0R +,/22H +ACCUMLATION TIM
6E=,F8.2,9H+ NUMBER+,12X,
714H+ DEVIATION +,12X,5H+ ERR0R,4X,24H+ [EVIATION + VARIANCE,
82X,1H+,/2H +,7X,13HPiece LENGTH=,F8.2,1H+,7X,6(1H+,12X),1H+,/1H ,1
916(1H+))
  XKNP=N0PCE
  DO 1 I=1,NCH
  DUM1=SQRT(PRMS(I))
  DUM2=SQRT(STERR1(I)/XKNP)
  DUM3=SQRT(ARMS(I))
  DUM4=SQRT(STERR2(I)/XKNP)
1  WRITE(N,1220) N0CHAN(I),PMNVAL(I),DUM1, AMNVAL(I),DUM2,
1      DUM3,DUM4
1220  FORMAT(2H +,28X,3H+ ,12,4H +,6(F12.8,1H+))                                FM 1220
  WRITE(N,1230)
1230  FORMAT(1X,116(1H+))
  RETURN
  END

```

```

$IBFTC DECK2I                                                DECK0000
  SUBROUTINE OU6A(N,NLAG,N0PCE,N0ACC,ARV02R,ACAC0R,ACCST,ACCTME,
1      ICHAN1,ICHAN2,PCELNC,TD,VELDLY,
2      PCEC0R,LLAG)
  DIMENSION TD(1),VELDLY(1),PCEC0R(1)
  WRITE(N,1260) N0PCE,N0ACC,ARV02R,ACCST,ACCTME,PCELNG,ICHAN1
1, ICHAN2
1260  FORMAT(1H ,25(1H+),24HAVERAGES WITH TIME DELAY,23(1H+),/9X,1H+,7X, FM 1260
213HPiece NUMBER=,15,14H +AREA UNDER,14X,1H+,/9X,21H+ACCUMLATION FM 1260
3  NUMBER=,15,16H +CORRELATION=,F12.8,1H+,/9X,21H+ACCUMLATION ST FM 1260
4ART=,F8.2,11H+CORRELATED,14X,1H+,/9X,21H+ACCUMLATION TIME=,F8.2 FM 1260
5,9H+CHANNELS,16X,1H+,/9X,1H+,7X,13HPiece LENGTH=,F8.2,2H+ ,12,5H A FM 1260
6ND ,12,14X,1H+,/9X,55(1H+),/13X,43H+ TIME [ELAY++VELOCITY DELAY++ P FM 12
7  PIECEWISE +,/13X,1H+,12X,2H++,14X,14H++CORRELATION+,/13X,43(1H+)) FM 12
  NTLAGS=2*NLAG+1
  DO 1 I=1,NTLAGS
  IF (LLAG .NE. 0) GO TO 4
  DUMMY=0.0
  GO TO 3
4  DUMMY=VELDLY(I)
3  WRITE (N,1300) TD(I),DUMMY,PCEC0R(I)
1300  FORMAT(13X,1H+,F11.5,6H ++ ,F7.3,4X,2H++,F11.5,1H+)
1  CONTINUE
  WRITE(N,1310)
1310  FORMAT(1X,74(1H+))
  RETURN
  END

```

```

$ORIGIN          ALPHA,SYSUT2,REW
$INCLUDE          SCLV,MXYV,M00V,LNRV,NLAV,PNBD,GRID,L0LV
$INCLUDE          PLOT,CMRV,PRTV,SYMV,GRAX,CX0Y,LHNV,ERLV
$INCLUDE          ERNV,EPKV,HL0V,LINV,STCV,STPV,FLZG
$INCLUDE          MRGN,NBLK,PLTV,APLV,APRV,F2LV,CULP,GUKV
$INCLUDE          CLEN,FSCN,FATN
$IBFTC DECK1     LIST,N00D,DECK
                SUBROUTINE LINK3(KRUN)

C
C      OUTPUT LINK OF MATCHER PROGRAM
C
C      THIS LINK FORMS MODIFIED CORRELATIONS FROM
C
C      SELECTED PIECEWISE STATISTICS
C
C      REGIONS FOR CHANNEL ASSIGNMENTS
C
C      DIMENSION NCHID(12),NEWID(12),NCTBL(12, 6),ACRPC(12)
C      1,NACR(13),ACD(12)
C
C      TEMPORARY STORAGE
C
C      COMMON START(5),STOP(5),SC(53),VALMIN(12),VALMAX(12)
C      1,SLOPE(12),BIAS(12),LTH,NBEAMS,NFLSKC,NFLSKC,MVMEAN,
C      2TINSAM,NWDS,IFREQ,MSFC,AFILE,RMS(12),VMEAN(12),ACRUN,EXPND,ADPAGE,
C      4AVGMIN(12),AVGMAX(12),NBLK,NSTART,
C      3CONTIM
C      DIMENSION UI(8000),IU(2000)
C      DIMENSION COM(1)
C      DIMENSION NUMPCE(10),PRECD(12)
C      DIMENSION BEAC(5),ENAC(5)
C      EQUIVALENCE (START(1),COM(1))
C      EQUIVALENCE (U,IU)
C      EQUIVALENCE (TINSAM,ESMPP)
C      LOGICAL NORM, OTREND, IEAD,          PREPLT,PL0T
C      NAMELIST/DELETE/NUMPCE,APRINT
C      NAMELIST /MODIFY/ NORM,OTREND,      PREPLT,PL0T
C      NAMELIST/ACCUM/NO,BEGIN,END

C
C      NUMCOM=175
C      NBEAM=12
C      SYSTEM  PARAMETERS
C
C      NAME      DESCRIPTION      TYPE  DIMENSION
C
C      MAXSTR - MAXIMUM STORAGE OF DATA PZEL      I
C      MAXCHN - MAXIMUM NUMBER OF CHANNELS          I
C
C
C      MAXSTR= 8000
C      MAXCHN=12
C      MAXCR = 6
C      IRT=11
C
C      INTERMEDIATE TAPE
C
C      REWIND IRT
C      READ (IRT) (COM(I),I=1,NUMCOM)
C      READ (IRT) NCH,NO,NU,NAUTZ,NCRT,NCHID,NEWID,NCTBL,NCRPC,NACR,NCD
C      NTP=C

```

```

      IEND=.FALSE.
2  READ(5,DELETE)
   WRITE(6,DELETE)
      DO 1 I=1,10
      IF(NUMPCE(1).EQ.0) GO TO 3
      NTP=NTP+1
      IU(NTP)=NUMPCE(I)
1  CONTINUE
   GO TO 2
3  NST=NTP+1
C      SET TIME LAGS AND VELOCITY LAGS
   READ(IRT) DELTAT,TAUMIN,TAUMAX,DELTAU,NLAGS,NTLAGS,LLAG
   NTT=NST+NTLAGS+NCRT*(NTLAGS+LLAG)-1
   READ(IRT) (U(I),I=NST,NTT)
C      SET MODIFICATIONS
   READ(5,MODIFY)
   WRITE(6,MODIFY)
   INAC=0
40  READ(5,ACCUM)
   WRITE(6,ACCUM)
   IF(N0.EC.0) GO TO 41
   INAC=INAC+1
   BEAC(INAC)=BEGIN
   ENAC(INAC)=END
   STOP(NSTART)=STOP(NSTART)-TAUMAX
   IF (ENAC(INAC) .LT. STOP(NSTART)) GO TO 42
   NNTMP=STOP(NSTART)/DELTAT
   ENAC(INAC)=FLMAT(NNTMP)*DELTAT
42  CONTINUE
   GO TO 40
41  NAC=INAC
   READ(5,6) PR0CD
   WRITE(6,6)      PR0CD
6  FORMAT(12A6)
C
C      STORAGE POINTERS
C
   LTLAG=NST
   LVLAG=LTLAG+NTLAGS
   ICHA=LVLAG+NCRT*NTLAGS+LLAG
C      ERR2PS
   IESM=ICHA+ND
   IEAM=IESM+NCH
   IEASM=IEAM+NCH
   IEXP=IEASM+NCH
C
C      ACCUMULATION
C
   LASM=IEXP+NCRT*(NTLAGS+1)
   LMA=LASM+NCH
   LRA=LMA+NCH
   LXA=LRA+NCH
C      MODIFIED CORRELATIONS
   MPSM=LXA+NCRT*(NTLAGS+1)
   MMP=MPSM+NCH
   MRP=MMP+NCH
   MTP=MRP+NCH
C
   KEXP=MTP+NCRT*(NTLAGS+1)
   KATP=KEXP+NCRT*(NTLAGS+1)
C      PIECE WISE CORRELATIONS (FROM LINK2)

```

```

C      KESM=KATP+NCRT*(NTLAGS+1)
      KEAM=KESM+NCH
      KEASM=KEAM+NCH
      KASM=KEASM+NCH
      KMA=KASM+NCH
      KRA=KMA+NCH
      KPSP=KRA+NCH
      KMP=KPSP+NCH
      KRP=KMP+NCH
      KTP=KRP+NCH

C      KD=KESM
      KDM=KTP+NCRT*(NTLAGS+1)-1
      INAC=1
C      SET NEW
C      ACCUMULATION
C
29  ASTAR=REAC(INAC)
      ASTOP=ENAC(INAC)
      ACCTIM=C.0
      KNPT=0
      IPR=C
      NTC=MAXSTR-ICHA
      CALL CLEAR(NTC,U(ICHA))
      GET NEXT PIECE
C
C
30  READ(IRT) KNP,(U(I),I=KC,KDM)
      DO 31 I=1,NTPT
      IF(KNP.EC.IU(I)) GO T2 30
31  CONTINUE
      KNPT=KNPT+1
      IPR=IPR+1
      ACCTIM=ACCTIM+DELTAT
      CALL MOVEIU(KPSM),U(MPSM),3*NCH)
C
C      FORM MODIFIED CORRELATIONS
C
      DO 10 J=1,ND
      L=NCD(J)
      NT=NCRPC(J)
      DO 10 K=1,NT
      KRI=(K-1+NACR(J))*(NTLAGS+1)
      KR=KTP+KRI
      MT=MTP+KRI
      KRT=KR+NTLAGS
      MRT=MT+NTLAGS
      NCU=NCTBL(J,K)
16  IF(NCRM) GO T2 11
      SX=1.0
      SY=1.0
      GO T2 14
11  LX=MPSM-1+L
      LY=MPSM-1+NCU
      SX=SQRT(U(LX))
      SY=SQRT(U(LY))
      SXSX=SX*SX
14  AVLAG=C.0
      SAVLAG=C.0
      DO 15 I=1,NTLAGS
      LL=KR+I-1

```

```

PAVLAG=PAVLAG+U(LL)
15 CCONTINUE
U(KRT)=PAVLAG/FL0AT(NTLAQS)
IF (DTREND) G0 T0 13
TR=0.0
G0 T0 650
13 TR=U(KRT)
650 CCONTINUE
D0 651 I=1,NTLAGS
MM=NT+I-1
LL=KR+I-1
IF (PREMLT) G0 T0 600
XTAU=1.0
G0 T0 601
600 NNN=LTLAQ+I-1
XTAU=U(NNN)
601 CCONTINUE
U(MM)=(U(LL)-TR)/SXSY*XTAU
AVLAG=AVLAG+U(MM)
651 CCONTINUE
U(MRT)=AVLAG/FL0AT(NTLAQS)
10 CCONTINUE

C
C
C
C    ACCUMULATE PIECE
NTA=NCRT*(NTLAGS+1)
CALL PCRC(U(KATP),U(KTP),NTA,KNPT)
CALL STDERR(U(KEXP),U(KATP),U(KTP),NTA,KNPT)

C
NTT=3*NCF+NCRT*(NTLAGS+1)
CALL PCRC(U(LASM),U(MPSM),NTT,KNPT)

C
C
C    DETERMINE ERRORS
CALL STDERR(U(IESM),U(LASM),U(MPSM),NTT,KNPT)

C
C
C    FORM STATISTICAL AND CHANNEL AVERAGES
D0 60 J=1,ND
MIC=ICHA+J-1
U(MIC)=0.0
NT=NCRPC(J)
D0 61 K=1,NT
KRI=(K-1+NACR(J))*(NTLAGS+1)
KK=IEXP+KRI
KKK=KEXP+KRI
MMM=KKK+NTLAGS
MM=KK+NTLAGS
CALL AVERGE (U(MMM),U(KKK),NTLAGS)
CALL AVERGE (U(MM),U(KK),NTLAGS)
U(MM)=SQRT(U(MM)/FL0AT(KNPT))
U(MMM)=SQRT(U(MMM)/FL0AT(KNPT))
U(MIC)=U(MIC)+U(MM)
61 CCONTINUE
U(MIC)=U(MIC)/FL0AT(NT)
60 CCONTINUE

C
C
C
C    OUTPLT SELECTION

```

```

      IF (.NOT.IEND) GO TO 702
      GO TO 703
702  IF (NPRINT-IPR) 704,703,704
703  CONTINUE
      IPR=0
      CALL SUB1(6,EXPNO,SC(2),SC(1),NRPAGE,SC(3),SC(4),NORUN)
      DO 32 J=1,ND
      NT=NCRPC(J)
      MIC=ICHA+J-1
      DO 32 K=1,NT
      NCU=NCTBL(J,K)
      KRI=(K-1+NACR(J))*(NTLAGS+1)
C
C      SET CORRELATION POINTERS
C      PIECEWISE
      KR=KTP+KRI
      KAP=KATP+KRI
      KER=KEXP+KRI
C
C      MODIFIED
      MR=MTP+KRI
      MAP=LXA+KRI
      MER=IEXP+KRI
C
C      AVERAGE LAG POINTERS
      NI=NTLAGS
      KPAL=KR+NI
      KAPAL=KAP+NI
      MPAL=MR+NI
      MAPAL=MAP+NI
      KERP=KER+NI
      ARVCOR=U(KPAL)
      ACACOR=U(KAPAL)
      AMDCOR=U(MPAL)
      ACMCOR=U(MAPAL)
      MERP=MER+NI
      NN=NCI(J)
      N=NCI-ID(NN)
      NUP=NCI-ID(NCU)
      CALL SUB6(6,NLAGS,KNPT,INAC,ARVCOR,ACACOR,START(1),ACCTIM,
1N, NUP,DELTAT,AMDCOR,ACMCOR,U(LTLAG),U(LVLAT),U(KR),U(MR),
2U(KAP), U(KER),U(MAP),U(MER),LLAG)
      IF (PL0T) GO TO 607
      GO TO 608
607  CONTINUE
      I1=KDN+1
      CALL PL0TA(U(LTLAG),U(KR),U(MR),U(KAP), U(KER),U(MAP),U(MER),SC,
1KNPT,INAC,ACCTIM,N,NUP,DELTAT,NTLAGS,TIMSAF,U(I1),U(KERP),U(MERP))
      CALL CLEAR(NTLAGS,U(I1))
608  CONTINUE
      NCA=NCAY+J
      AVTIM=ACCTIM-START(1)
33  CALL SUB7(6,N, NUP,INAC,PR0CC,ACCTIM,START(1),AVTIM,U(KERP),
1U(MERP),U(MIC))
      IF(.NOT.IEND) GO TO 35
      MXM=LMA+NN-1
      MYM=LMA+NCU-1
      MXR=LASM+NN-1
      MYR=LASM+NCU-1
      CALL SUB8(6,NLAGS,N, U(MXM),NUP,U(MXR),U(MYM),U(MYR),U(LTLAG),
1U(LVLAT),U(KAP),U(MAP),LLAG)

```

```

      IF (KRLN .NE. 4) GO TO 701
      CALL LINK4(NLAGS,NTLAGS,DELTAU,U(LTLAG),U(MAP),U(11))
701  CONTINUE
    35  CONTINUE
    32  CONTINUE
704  CONTINUE
      A=ACCTIM-ASTAR
      B=ASTOP
      C=ACCTIM+DELTAT
      IF (C .GE. ASTOP) IEND=.TRUE.
      IF (A .GE. B) GO TO 101
      GO TO 30
101  IF(INAC.EQ.NAC) GO TO 100
      IEND=.FALSE.
      INAC=INAC+1
      GO TO 29
100  CONTINUE
      IF (PL0T) GO TO 605
      GO TO 606
605  CONTINUE
      CALL CLEAN
606  CONTINUE
      REWIND IRT
      RETURN
      END

```

\$IBFTC DECK2

DECK0000

```

      SUBROUTINE PCRC(A,X,N,NP)
C
C      THIS SUBROUTINE PERFORMS THE PIECEWISE
C      RECURSION FORMULA FOR ACCUMULATIVE AVERAGES
C       $A(N+1) = (A(N)*N + X(N+1)) / (N+1)$ 
C
      DIMENSION A(1),X(1)
      XKNP=NP
      XKNP1=NP-1
      DO 1 I=1,N
1  A(I)=(A(I)*XKNP1+X(I))/XKNP
      RETURN
      END

```

*IBFTC DECK3

DECK0000

```

SUBROUTINE STDERR(E,A,P,N,NP)
C
C   THIS SUBROUTINE PERFORMS THE RECURSION FORMULA
C   FOR THE STANDARD ERROR OF THE ACCUMULATIVE AVERAGES
C
C    $E(N+1) = E(N) * (N-1) / N + ((A(N+1) - P(N+1) - P(N))^2 * (N+1) / N^3)$ 
C
  DIMENSION E(1),A(1),P(1)
  IF(NP.EQ.1) GO TO 2
  XNP=NP-1
  XNPM1=NP-2
  XNPP1=NP
  DO 1 I=1,N
1 E(I)=E(I)*XNPM1/XNP+((A(I)-P(I))^2)*XNPP1/XNP**2
2 RETURN
END

```

*IBFTC DECK6

DECK0000

```

SUBROUTINE AVERGE(A,U,N)
C   FORM S AVERAGE VALUE OF N VALUES OF U AND
C
C   STORES AVERAGE IN A
C
C
C
  DIMENSION U(1)
  XN=N
  A=C.O
  DO 1 I=1,N
1 A=A+U(I)
  A=A/XN
  RETURN
END

```

*IBFTC DECK5

DECK0000

```

SUBROUTINE MOVE(X,Y,N)
C
C   MOVE N LOCATIONS FROM X TO Y
C
  DIMENSION X(1),Y(1)
  DO 1 I=1,N
1 Y(I)=X(I)
  RETURN
END

```

\$IBFTC DECK4

DECK0000

```

SUBROUTINE CLEAR(N,X)
C
C   THIS SUBROUTINE SETS N POSITIONS OF X TO ZERO
C
  DIMENSION X(1)
  DO 1 I=1,N
1 X(I)=0.0
  RETURN
  END

```

\$IBFTC DECK7

DECK0000

```

SUBROUTINE ZUT6(N,NLAG,NOPCE,NOACC,ARVCOR,ACACOR,ACCST,ACCTME,
1 ICHAN1,ICHAN2,PCELNG,AMCOCOR,ACMCOR,TD,VELDLY,
2 PCECOR,PCMCOR,ACOR,STDER3,APCOR,STDER4,LLAG)
  DIMENSION TD(1),VELULY(1),PCECOR(1),PCMCOR(1),ACOR(1),STDER3(1),
1 AMCOR(1),STDER4(1)
  WRITE(N,1260) NOPCE,NOACC,ARVCOR,ACACOR,ACCST,ACCTME,ICHAN1,ICHAN2
1 ,PCELNG,AMCOCOR,ACMCOR
1260 FORMAT(1H ,34(1H+),24HAVERAGES WITH TIME DELAY,44(1H+),/2H +,7X,13 FM 1260
2HPiece NUMBER=,15,14H +AREA UNDER,14X,35H+ACCUMULATIVE AREA UNDE FM 1260
3R +CORRELATED ++,8X,1H+,/22H +ACCUMULATION NUMBER=,15,16H +CORRE FM 1260
4LATION=,F12.8,13H+CORRELATION=,F12.8,14H+ CHANNELS ++,8X,1H+,/22H FM 1260
5 +ACCUMULATION START=,F8.2,11H+AREA UNDER,14X,26H+ACCUMULATIVE AR FM 1260
6EA UNDER +,11X,2H++,8X,1H+,/22H +ACCUMULATION TIME=,F8.2,2(9H+M0 FM 1260
7DIFIED,16X),2H+ ,12,5H AND ,12,3H ++,9X,1H+,/2H +,7X,13HPiece LENG FM 1260
8TH=,F8.2,2(13H+CORRELATION=,F12.8),1H+,11X,2H++,8X,1H+,/1H ,102(1H FM 1260
9+),/103H +TIME DELAY++VELOCITY DELAY++ PIECEWISE ++ PIECEWISE ++AC FM 1260
10CUMULATIVE++STANDARD++ACCUMULATIVE++STANDARD+,/2H +,10X,2H++,14X,7 FM 1260
25H++CORRELATION++ MODIFIED ++CORRELATION ++ ERRORR ++ MODIFIED FM 1260
3++ ERRORR +,/2H +,10X,2H++,14X,2H++,11X,15H++CORRELATION++,12X,2H+ FM 1260
4+,8X,16H++CORRELATION ++,8X,1H+,/1H ,102(1H+)) FM 1260
  NTLA=2*NLAG+1
  DO 1 I=1,NTLA
  DUMMY1=SQRT(STDER3(I)/FLOAT(NOPCE))
  DUMMY2=SQRT(STDER4(I)/FLOAT(NOPCE))
  IF (LLAG .NE. 0) GO TO 4
  DUMMY=C.0
  GO TO 3
  DUMMY=VELDLY(I)
3 WRITE(N,1300) TD(I),DUMMY, PCECOR(I),PCMCOR(I),ACOR(I),DUMMY1,
1 AMCOR(I),DUMMY2
1300 FORMAT(2H +,F9.5,6H ++ ,F7.3,4X,2(2H++,F11.5),2(2H++,F12.5,2H++, FM 1300
2F8.5),1H+)
1 CONTINUE
  WRITE(N,1310)
1310 FORMAT(1X,102(1H+))
  RETURN
  END

```

\$IBFTC DECK8

DECK0000

```

SUBROUTINE OUT7(N, ICHANA, ICHANB, N0ACC, PR0CD, ENACC, ACCST,
1 AVETME, CSTERR, CMSTER, CHERR)
DIMENSION AVETME(1), CSTERR(1), CMSTER(1), CHERR(1), PR0CD(1)
WRITE(N, 1320) ICHANA, ICHANB, N0ACC, (PR0CD(I), I=1, 5), ENACC, (PR0CD(I)
1, I=6, 10), ACCST, (PR0CD(I), I=11, 12)
1320 FORMAT( //1H ,42(1H+), 25HINTEGRATED ERROR ANALYSIS, 35(1H+), /24H + FM 1320
2 CORRELATED CHANNELS=, 12, 5H AND , 12, 1H+, 11X, 13H+ MODIFIED +, 11X, 2 FM 1320
39H+ ACCUMULATION PROCEDURE USED=, 4X, 1H+, /24H + ACCUMULATION NUMBER FM 1320
4=, 15, 4X, 37H+ CORRELATION+ CORRELATION+ CHANNEL +, 5(A6), 2X, 1H+, /24H FM 1320
5 + END OF ACCUMULATION=, F9.2, 38H + STATISTICAL+ STATISTICAL+ STATIST FM 1320
6ICAL+, 5(A6), 2X, 1H+, /24H + START OF ACCUMULATION=, F8.2, 38H + ERROR FM 1320
7 + ERROR + ERROR +, 2(A6), 20X, 1H+, /1H , 102(1H+), /1H , 17(1
8H+), 14HAVERAGING TIME, 71(1H+)
WRITE(N, 1340) AVETME, CSTERR, CMSTER, CHERR
1340 FORMAT(2H +, 15X, 2H+ , F9.3, 5X, 3(1H+, F11.7), 1H+, 32X, 1H+) FM 1340
WRITE(N, 1360)
1360 FORMAT(1H , 102(1H+)) FM 1360
RETURN
END

```

\$IBFTC DECK9

DECK0000

```

SUBROUTINE OUT8(N, NLAG, IC0RCH, XMNVAL, JC0RCH, XRMS, YMNVAL, YRMS, TD,
1VELDLY, C0R, MC0R, LLAG)
XRMS=SQRT(XRMS)
YRMS=SQRT(YRMS)
WRITE(N, 1380) IC0RCH, XMNVAL, XRMS, JC0RCH, YMNVAL, YRMS
DIMENSION TD(1), VELDLY(1), C0R(1), MC0R(1)
1380 FORMAT( //1H ,34(1H+), 27HFINAL TABULATION OF RESULTS, 24(1H+), /39H FM 1380
2+ CORRELATED + MEAN VALUE+ ROOT MEAN +, 11X, 1H+, 10X, 2(1H+, 11X), 1H+, FM 1380
3/14H + CHANNELS +, 11X, 14H+ SQUARE VALUE+, 11X, 12H+ VELOCITY +, 11X, 1 FM 1380
43H+ MODIFIED +, /2H +, 4X, 12, 5X, 2(1H+, F11.7) , 49H + TIME DELAY+ FM 1380
5 DELAY + CORRELATION+ CORRELATION+, /2H +, 4X, 12, 5X, 1H+, F11.7, 2H+ , F FM 1380
611.7, 1H+, 11X, 1H+, 10X, 2(1H+, 11X), 1H+, /1H , 85(1H+)
NTLAGS=2*NLAG+1
DO 1 I=1, NTLA GS
IF (LLAG .NE. 0) GO TO 4
DUMMY=0.0
GO TO 3
4 DUMMY=VELDLY(I)
3 WRITE(N, 1500) TD(I), DUMMY, C0R(I), MC0R(I)
1500 FORMAT(2H +, 36X, 2H+ , 2(F10.6, 1H+), 2(F11.7, 1H+)) FM 1500
1 CONTINUE
WRITE(N, 1520)
1520 FORMAT(1H , 85(1H+)) FM 1520
RETURN
END

```

\$IBFTC PLT1

PLT10000

```

SUBROUTINE PL0TA (TD,PCEC0R,PCMC0R,AC0R,STDER3,AMC0R,STDER4,SC,
1KNPT,INAC,ACCTIM,N,NUP,PCELNG,NLAGS,TIMSAM,TEMP,ERR1,ERR2)
DIMENSION TD(1),PCEC0R(1),PCMC0R(1),AC0R(1),STDER3(1),
1AMC0R(1),STDER4(1),SC(1),TEMP(1)
DIMENSION INF0R(6)
DIMENSION TITLE1(9),TITLE2(9),TITLE3(3),TITLE4(9)
DIMENSION ALAB1(12),0LAB1(12),0LAB2(12),0LAE3(12),0LAB4(12)
DATA NFLAG/0/
DATA TITLE1/54H
PIECEWISE CORRELATION
1 /
DATA TITLE2/54H
PIECEWISE MODIFIED CORRELATION
1 /
DATA TITLE3/54H
ACCUMULATIVE CORRELATION AND STANDARD ERROR
1 /
DATA TITLE4/54H
ACCUMULATIVE MODIFIED CORRELATION AND STANDARD ERROR
1R /
DATA ALAB1/72H
1TIME DELAY /
DATA 0LAB1/72H
CORRELATIONS
1 /
DATA 0LAB2/72H
MODIFIED CORRELATIONS
1 /
DATA 0LAB3/72H
CORRELATION AND STANDARD ERROR
1 /
DATA 0LAB4/72H
MODIFIED CORRELATION AND STANDARD ERROR
1 /
DATA INF2R/36H470960 BIN 309 63 JAYR0E R. COCC00/
IF (NFLAG) 11,10,11
10 CONTINUE
CALL SCL0CK (INF2R(6),TIME,ESEC,E60SEC)
CALL CAMRAV(9)
NFLAG=1
11 CONTINUE
XKNPT=KNPT
NX=-1
CALL PL0TB1 (ALAB1,0LAB1,TITLE1,TD,PCEC2R,NLAGS,INAC,N,NUP,
1TIMSAM,ACCTIM,PCELNG,SC,NX,INF0R(6),KNPT,ERR1,ERR2)
NX=-1
CALL PL0TB1 (ALAB1,0LAB2,TITLE2,TD,PCMC2R, NLAGS,INAC,N,NUP,
1TIMSAM,ACCTIM,PCELNG,SC,NX,INF0R(6),KNPT,ERR1,ERR2)
NX=-1
CALL PL0TB1 (ALAB1,0LAB3,TITLE3,TD,AC0R, NLAGS,INAC,N,NUP,
1TIMSAM,ACCTIM,PCELNG,SC,NX,INF0R(6),KNPT,ERR1,ERR2)
D0 12 I1=1,NLAGS
TEMP(I1)=SQRT(STDER3(I1)/XKNPT)
12 CONTINUE
NX=0
CALL PL0TB2 (TD,TEMP, NX)
NX=-1
CALL PL0TB1 (ALAB1,0LAB4,TITLE4,TD,AMC0R, NLAGS,INAC,N,NUP,
1TIMSAM,ACCTIM,PCELNG,SC,NX,INF0R(6),KNPT,ERR1,ERR2)
D0 13 I2=1,NLAGS
TEMP(I2)=SQRT(STDER4(I2)/XKNPT)
13 CONTINUE
NX=0
CALL PL0TB2 (TD,TEMP, NX)
RETURN
END

```

\$IBFTC PLT2

PLT20000

```

SUBROUTINE PLOTB1(AARRAY,ZARRAY,TITLE,XDATA,YDATA,NP,INAC,N,NUP,
ITIMSAM,ACCTIM,PCELNG,SC,NXX,DATE,KNPT,ERR1,ERR2)
DIMENSION SC(1),AARRAY(1),ZARRAY(1),TITLE(1),XDATA(1),YDATA(1)
INTEGER TEMPX1,TEMPX2,TEMPY1,TEMPY2
FINAC=INAC
FKNPT=KNPT
FNN=N
FNUP=NUP
N1=N+7
N2=NUP+7
NPP=-NP
CALL FRAMEV
CALL PRINTV (54,TITLE,345,1017)
CALL PRINTV (15,15HUNMOD ERROR =,200,984)
CALL LABLV (ERR1,328,984,-6,1,1)
CALL PRINTV (8,8HPUN NZ.=,200,1000)
CALL PRINTV (6,SC(1),272,1000)
CALL PRINTV (10,10HMOD ERROR=,200,968)
CALL LABLV (ERR2,290,968,-6,1,1)
CALL PRINTV (16,16HTOTAL PIECE NZ.=,200,952)
CALL LABLV (FKNPT,336,952,6,1,5)
CALL PRINTV (11,11HACCUM. NZ.=,200,936)
CALL LABLV (FINAC,296,936,6,1,5)
CALL PRINTV (21,21HCORRELATION CHANNELS=,200,920)
CALL LABLV (FNN,344,920,6,1,5)
CALL PRINTV (3,3HAND,400,920)
CALL LABLV (FNUP,408,920,6,1,5)
CALL PRINTV (5,5HDATE=,600,1000)
CALL PRINTV (6,DATE, 648,1000)
CALL PRINTV (18,18HFINAL SAMPLE RATE=,600, 984)
CALL LABLV (TIMSAM,752, 984,-6,1,1)
CALL PRINTV (11,11HACCUM TIME=,600, 968)
CALL LABLV (ACCTIM,696, 968,-6,1,1)
CALL PRINTV (13,13HPIECE LENGTH=,600,952)
CALL LABLV (PCELNG,712,952,-6,1,5)
CALL PRINTV (13,13HORGANIZATION=,600,936)
CALL PRINTV (6,SC(2),712,936)
CALL PRINTV (13,13HCHANNEL VARIABLES=,600,920)
CALL PRINTV (6,SC(N1), 752,920)
CALL PRINTV (3,3HAND,800,920)
CALL PRINTV (6,SC(N2), 832,920)
CALL PRINTV (72,AARRAY,C,0)
CALL APRNTV (C,-14,72,ZARRAY,0,1010)

```

```

C      *
C      SET MIN AND MAX LIMITS
C      *
      XR=XDATA(NP)
      XL=XDATA(1)
      TEMPI=YDATA(1)
      J=1
      DO 10 I=2,NP
      IF (TEMPI .LE. YDATA(I)) GO TO 10
      TEMPI=YDATA(I)
      J=I
10    CONTINUE
      YB=YDATA(J)
      TEMPI=YDATA(1)
      J=1
      DO 11 I=2,NP
      IF (TEMPI .GE. YDATA(I)) GO TO 11

```

```

      TEMP1=YDATA(I)
      J=I
11  CONTINUE
      YT=YDATA(J)
      CALL SETMIV(24,0,24,124)
      CALL DXCYV (1,XL,XR,DX,NX,IX,NNX,IC.,IERR,24,124)
      CALL DXCYV (2,YB,YT,DY,NY,IY,NNY,IC.0,IERR,24,124)
      CALL GRID1V(2,XL,XR,YB,YT,DX,DY,NX,NY,IX,IY,NNX,NNY,24,124)
      TEMPX1=NXV(XDATA(1))
      TEMPY1=NYV(YDATA(1))
      DO 12 J=2,NP
      TEMPX2=NXV(XDATA(J))
      TEMPY2=NYV(YDATA(J))
      CALL LINEV (TEMPX1,TEMPY1,TEMPX2,TEMPY2)
      TEMPX1=TEMPX2
      TEMPY1=TEMPY2
12  CONTINUE
      RETURN
      ENTRY PLQTB2 (XDATA,YDATA,NXX)
      TEMPX1=NXV(XDATA(1))
      TEMPY1=NYV(YDATA(1))
      DO 13 J=2,NP
      TEMPX2=NXV(XDATA(J))
      TEMPY2=NYV(YDATA(J))
      CALL LINEV (TEMPX1,TEMPY1,TEMPX2,TEMPY2)
      TEMPX1=TEMPX2
      TEMPY1=TEMPY2
13  CONTINUE
      RETURN
      END

```

APPENDIX C

DETRENDING AND NORMALIZATION OF CORRELATION FUNCTIONS

Correlation analysis is strictly valid only for stationary data, i.e., experiments, where the test environment is time invariant. In the piecewise correlation techniques the condition of stationarity is approximated by integrating only over a short piece of a record of length ΔT since it is hoped, that in this short integration time the environment does not change. However, corrections for DC shifts and trends as well as drifting gains can not be avoided, if more accurate averages have to be obtained by averaging over many pieces.

Detrending procedures and normalization are thus necessary for each piece, to correct for the change of environment, prior to averaging over more than one piece.

Detrending procedures are developed by assuming, that DC shifts and trends may be linearly approximated, over a record piece of length ΔT . The recorded two signal time histories, x and y , may then be expressed by their stationary components $x_1(t)$ and $x_2(t)$ and the associated trends in the following way:

$$\left. \begin{aligned} x(t) &= x_1(t) + a_1 t + b_1 \\ y(t) &= x_2(t) + a_2 t + b_2 \end{aligned} \right\} \quad \text{for } (i-1) \Delta T \leq t \leq i \Delta T \quad (1)$$

The condition of stationarity means, that the mean values of the stationary components should vanish.

$$\begin{aligned}\bar{x}_{1_i} &= \frac{1}{\Delta T} \int_{(i-1)\Delta T}^{i\Delta T} x_1(t) dt = 0 \\ \bar{x}_{2_i} &= \frac{1}{\Delta T} \int_{(i-1)\Delta T}^{i\Delta T} x_2(t) dt = 0\end{aligned}\tag{2}$$

Furthermore the correlation function of the stationary components

$$R_{12}(\tau)_i = \frac{1}{\Delta T} \int_{(i-1)\Delta T}^{i\Delta T} x_1(t) x_2(t + \tau) dt\tag{3}$$

should give zero area, when averaged over the time lag range of interest, ($|\tau| \leq \tau_m$), otherwise the stationary components could not be expressed by superposition of harmonic waves. This condition leads to

$$\tau R_{12} = \frac{1}{2\tau_m} \int_{-\tau_m}^{+\tau_m} R_{12}(\tau) d\tau = 0\tag{4}$$

The detrending problem may now be defined by determining, what the corrections of the actual correlation:

$$(R_{xy}(\tau))_i = \frac{1}{\Delta T} \int_{(i-1)\Delta T}^{i\Delta T} x(t) y(t + \tau) dt\tag{5}$$

are necessary to retrieve the correlation $(R_{12}(\tau)_i)$ of the stationary components. These corrections will now be derived. All over bars and the subscripts, are dropped from the correlation function, as a matter of convenience.

The relation between the accessible correlation, $R_{xy}(\tau)$ and the desired correlation, $R_{12}(\tau)$, may be expressed as

$$\begin{aligned}
 R_{xy}(\tau) = & \overline{(x_1(t) + a_1 t + b_1)(x_2(t+\tau) + a_2 t + a_2 \tau + b_2)} = \overline{x_1(t) x_2(t+\tau)} \\
 & + (b_2 + a_2 \tau) \overline{x_1(t)} + b_1 \overline{x_2(t+\tau)} + a_2 \overline{t x_1(t)} + a_1 \overline{t x_2(t+\tau)} \\
 & + \frac{1}{\Delta T} \int_{(i-1)\Delta T}^{i\Delta T} (a_1 t + b_1) (a_2 t + b_2 + a_2 \tau) dt \quad (6)
 \end{aligned}$$

The first average on the right hand side is the desired correlation, $R_{12}(\tau)$ for positive values of τ . The second and third averages vanish according to equation (2). The fourth and fifth averages may also be neglected, as may be shown, through an integration by parts. To illustrate, we take the fourth average:

$$\overline{(t x_1(t))} = \Delta T \overline{(x_1(\tau))} - \frac{1}{\Delta T} \int_{t=(i-1)\Delta T}^{t=i\Delta T} \int_{t'=0}^{t'=t} (x_1(t) dt') dt \quad (7)$$

In this equation the first term vanishes because of equation (2). The inner integrand of the second integral is an accumulative mean, which approaches zero in an oscillatory manner. This term might, therefore, also be neglected as stated above. The sixth term in equation (6) is an integral that may be given in closed form:

$$\begin{aligned}
 & \frac{1}{\Delta T} \int_{t=(i-1)\Delta T}^{t=i\Delta T} (a_1 t + b_1) (a_2 t + b_2 + a_2 \tau) dt \\
 & = b_1 (b_2 + a_2 \tau) + [a_1 b_2 + a_2 (a_1 \tau + b_1)] \frac{\Delta T}{2} + a_1 a_2 \frac{\Delta T^2}{3} \quad (8)
 \end{aligned}$$

Substituting all these averages into equation (6) one obtains

$$R_{xy}(\tau) = R_{12}(\tau) + a_1 a_2 \frac{\Delta T^2}{3} + (a_1 b_2 + a_2 b_1) \frac{\Delta T}{2} + b_1 b_2 + a_2 \left(\frac{a_1 \Delta T}{2} + b_1 \right) |\tau| \text{ for } \tau \geq 0 \quad (9)$$

Negative values of τ are introduced in the correlation programs by switching channels

$$R_{xy}(-\tau) = \overline{(x(t+\tau) y(t))_i} = R_{12}(-\tau) + a_1 a_2 \frac{\Delta T^2}{3} + (a_1 b_2 + a_2 b_1) \frac{\Delta T}{2} + b_1 b_2 + a_1 \left(\frac{a_2 \Delta T}{2} + b_2 \right) |\tau| \text{ for } \tau > 0 \quad (10)$$

The equations (9) and (10) are now reduced to experimentally accessible terms in two steps. The first step is to eliminate the constant b_1 and b_2 . This is accomplished by employing the piecewise averages.

$$\overline{x(t)} = \overline{x_1(t) + a_1 t + b_1} = \frac{a_1 \Delta T}{2} + b_1 \quad (11)$$

and

$$\overline{y(t)} = \overline{x_2(t) + a_2 t + b_2} = \frac{a_2 \Delta T}{2} + b_2 \quad (12)$$

The constant in equation (9) and (10) may then be transformed as

$$(a_1 b_2 + a_2 b_1) \frac{\Delta T}{2} + b_1 b_2 = \frac{a_1 \Delta T}{2} \left(\overline{y} - \frac{a_2 \Delta T}{2} \right) + \frac{a_2 \Delta T}{2} \left(\overline{x} - \frac{a_1 \Delta T}{2} \right) + \left(\overline{x} - \frac{a_1 \Delta T}{2} \right) \left(\overline{y} - \frac{a_2 \Delta T}{2} \right) = \overline{x} \overline{y} - a_1 a_2 \frac{\Delta T^2}{4} \quad (13)$$

Substitution of equation (11) through (13) into equations (9) and (10) gives

$$R_{12}(\tau) = R_{xy}(\tau) - \bar{x} \bar{y} - \frac{1}{12} a_1 a_2 \Delta T^2 - a_2 \bar{x} |\tau| \quad \text{for } \tau > 0 \quad (14)$$

and

$$R_{12}(\tau) = R_{xy}(\tau) - \bar{x} \bar{y} - \frac{1}{12} a_1 a_2 \Delta T^2 - a_1 \bar{y} |\tau| \quad \text{for } \tau < 0 \quad (15)$$

Both expressions differ only with respect to the last term. The second step, the elimination of a_1 and a_2 , is accomplished by expressing the left hand side of equations (14) and (15) in terms of the covariance function.

$$C_{xy}(\tau) = \overline{(x(t) - \bar{x})(y(t+\tau) - \bar{y})} = \overline{x(t) y(t+\tau)} - \bar{x} \overline{y(t+\tau)} \quad (16)$$

substituting

$$\overline{y(t+\tau)} = \overline{x_2(t) + a_2(t+\tau) + b_2} = a_2 \left(\frac{\Delta T}{2} + \tau \right) + b_2 = \bar{y} + a_2 \tau \quad (17)$$

gives

$$C_{xy}(\tau) = R_{xy}(\tau) - \bar{x} \bar{y} - a_2 \bar{x} \tau \quad \text{for } \tau > 0 \quad (18)$$

Comparing equation (18) with equation (14) gives an expression

$$R_{12}(\tau) = \overline{(x(t) - \bar{x})(y(t+\tau) - \bar{y})} - \frac{a_1 a_2}{12} \Delta T^2 \quad (19)$$

that is valid for positive and negative time delays. The last term may be calculated by integrating equation (19) over the time delay range and by applying equation (4).

$$\frac{1}{2\tau_m} \int_{-\tau_m}^{+\tau_m} R_{12}(\tau) d\tau = \tau C_{xy} - \frac{a_1 a_2}{12} \Delta T^2 = 0 \quad (20)$$

This gives the final result:

$$\overline{R_{12}(\tau)}_i = \overline{\left[(x(t) - \bar{x}_i)(y(t+\tau) - \bar{y}_i) \right]}_i - \tau (\overline{C_{xy}})_i \quad (21)$$

DC shifts and trends are thus eliminated from piecewise correlation functions by the following procedure:

- (a) Subtract the piecewise means $\bar{x}_i$ and $\bar{y}_i$ prior to the product integration.
- (b) Shift the resulting correlation curve by the amount $\tau (\overline{C_{xy}})_i$ until the area under the shifted covariance curve vanishes.

Several important conclusions follow:

- (1) The usual transformation between correlation and covariance functions

$$C_{xy} = R_{xy} - \bar{x} \bar{y} \quad (22)$$

is not valid in the presence of DC shifts and trends. The translation $\bar{x} \bar{y}$ must be replaced with the area shift τC_{xy} and the shape of C_{xy} and R_{xy} may be quite different, as indicated by the shape corrections $a_2 \bar{x}(\tau)$ in equation (18).

- (2) The area shift of the autocovariance curves

$$\tau (\overline{C_{xx}})_i = \frac{1}{2\tau_m} \int_{-\tau_m}^{+\tau_m} \left[\overline{(x(t) - \bar{x}_i)(x(t+\tau) - \bar{x}_i)} \right]_i d\tau \quad (23)$$

gives a direct estimate of the DC trends.

$$\frac{a_1 \Delta T}{2} = \sqrt{3_\tau (\overline{C_{xx}})_i} \quad (24)$$

$$\frac{a_2 \Delta T}{2} = \sqrt{3_\tau (\overline{C_{yy}})_i}$$

which, in turn, may be used to calculate the DC shifts

$$b_1 = \bar{x}_i - \sqrt{3_\tau (\overline{C_{xx}})_i} \quad (25)$$

$$b_2 = \bar{y}_i - \sqrt{3_\tau (\overline{C_{yy}})_i}$$

- (3) The amplitudes of the stationary components are given by the shifted covariance at zero time lag. The usual practice of normalizing with the unshifted covariance curve is incorrect.

$$\sigma_{x_i} = \overline{R_{11}(0)}_i^{1/2} = \sqrt{\left[(C_{xx}(\tau)) \right]_i (\tau=0) - \tau (C_{xx})_i} \quad (26)$$

and

$$\sigma_{y_i} = \overline{R_{22}(0)}_i^{1/2} = \sqrt{\left[(C_{yy}(\tau)) \right]_i (\tau=0) - \tau (C_{yy})_i} \quad (27)$$

- (4) The accumulation over many pieces can correct for environmental variations of the gain factors as well as for DC shifts and trends, by normalizing with the amplitudes of the stationary components. This gives the accumulative normalized covariance function

$$C_{12}(\tau) = \frac{1}{m} \sum_{i=1}^m \frac{\overline{C_{xy}(\tau)}_i - \tau(\overline{C_{xy}})_i}{\overline{\sigma_x}_i \overline{\sigma_y}_i} . \quad (28)$$

- (5) The assumption of linear trends may be checked by calculating the ratio

$$\frac{\sqrt{\tau(\overline{R_{xx}})_i \tau(\overline{R_{yy}})_i}}{\tau(\overline{R_{xy}})_i} = \alpha. \quad (29)$$

Within the assumptions made in equations (1) through (4), this ratio should be equal to one.

- (6) Additional checks are offered by the accumulative statistical error of the RMS amplitudes. The curves $\Delta(\overline{\sigma_x})_m (T^{-1/2})$ and $\Delta(\overline{\sigma_y})_m (T^{-1/2})$ should be more stationary, after removing the DC trends through equations (26) and (27).

APPENDIX D

EXPECTED ERROR CURVES

Let $()$ denote a statistic, i.e., a first or second order two channel product mean value between signals X and Y. The piecewise average of this statistic is defined as

$$(\overline{})_i = \frac{1}{\Delta T} \int_{(i-1)\Delta T}^{i\Delta T} () dt. \quad (1)$$

The expected error curves of these piecewise averages are derived from the hypothesis that all pieces of a long record belong to the same population of experiments, i.e., that they have been recorded in the same environment. In this case, the accumulative mean

$$(\overline{})_m = \frac{1}{m} \sum_{i=1}^m (\overline{})_i \quad (2)$$

should asymptotically approach a limiting or true value, $E [(\overline{})]$, with increasing record length $T = m\Delta T$. This true mean deviates from the sample mean $(\overline{})_m$, by the amount $(\overline{})_m - E [(\overline{})]$. If the experiment could be repeated many times, each repeated run would give a different deviation of the accumulative mean, $(\overline{})_m$, from the true mean, $E [(\overline{})]$. Then the deviation should furthermore follow the χ^2 distribution. Knowing this distribution, one can express the expected statistical error of the accumulative mean by

$$E \left[(\overline{})_m - E [(\overline{})] \right] = \sqrt{\frac{m-1}{m}} \frac{A}{\sqrt{T}}. \quad (3)$$

The calculation of the constant A will be treated later. The actual error might deviate from this expected error within a confidence interval, which is also given by the χ^2 distribution.

Only 10 percent of all repeated runs will show an error, which is smaller than the lower confidence level, subscript 0.1

$$|(\bar{\epsilon})_m - E [(\epsilon)]| \leq \frac{\chi_{0.1}^{(m)}}{\sqrt{m}} \frac{A}{\sqrt{T}} \quad (4)$$

for 10 percent of all runs. However, 90 percent of all repeated runs will show errors, which are exceeded by the upper confidence level, subscript 0.9

$$|(\bar{\epsilon})_m - E [(\epsilon)]| \leq \frac{\chi_{0.9}^{(m)}}{\sqrt{m}} \frac{A}{\sqrt{T}} \quad (5)$$

for 90 percent of all runs. The expected error curves might thus be summarized as:

$$\frac{\chi_{0.1}^{(m)}}{\sqrt{m}} \frac{A}{\sqrt{T}} \quad \text{lower confidence level.}$$

$$\sqrt{\frac{m-1}{m}} \frac{A}{\sqrt{T}} \quad \text{expected error of accumulative mean.}$$

$$\frac{\chi_{0.9}^{(m)}}{\sqrt{m}} \frac{A}{\sqrt{T}} \quad \text{upper confidence level.}$$

The functions $\chi_{0.1}^{(m)}$ and $\chi_{0.9}^{(m)}$ are available in statistical handbooks and are given in Table VII. The constant A has to be calculated from the actual error curves.

The present estimate of A is based on the observations that the relative deviation of the statistical error,

$$\frac{|(\bar{\epsilon})_m - E [(\epsilon)]| - E [|(\bar{\epsilon})_m - E [(\epsilon)]|]}{E [|(\bar{\epsilon})_m - E [(\epsilon)]|]}$$

is practically independent of the number of pieces, m , that was used in the accumulation process. For a group of i pieces, one sample of the accumulative statistical error may be calculated directly from the deviations between the piecewise averages $(\bar{})_k$ around their accumulative average $(\bar{})_i$

$$\Delta(\bar{})_i = \frac{1}{i(i-1)} \sum_{k=1}^i \left((\bar{})_k - (\bar{})_i \right)^2 \quad (6)$$

where $i = 2, 3, \dots, m$.

The expected value of this error is given by $\sqrt{(i-1)/i} A/\sqrt{i\Delta T}$, and the relative deviation between the sample $\Delta(\bar{})_i$ and this most probable error should be practically independent of the accumulation number i . The constant A , is now chosen such, that this relative deviation becomes a minimum. Choosing only the one combination of i pieces, that is available to a completely sequential data processing logic, i.e., the samples given by equation (6). The above variation principle may be expressed as:

$$\frac{\partial}{\partial A} \sum_{i=2}^m \left\{ \frac{\Delta(\bar{})_i - \sqrt{\frac{i-1}{i}} \frac{A}{\sqrt{i\Delta T}}}{\sqrt{\frac{i-1}{i}} \frac{A}{\sqrt{i\Delta T}}} \right\}^2 = 0 \quad (7)$$

Carrying out the partial differentiation of equation (7) gives an algebraic equation for A . The solution of this equation gives the desired computer estimate of A :

$$A = \sqrt{\Delta T} \frac{\sum_{i=2}^m \frac{i^2 \Delta(\bar{})_i^2}{(i-1)}}{\sum_{i=2}^m \frac{i \Delta(\bar{})_i}{\sqrt{i-1}}} \quad (8)$$

TABLE VII
EXPECTED ERROR AND CONFIDENCE LEVELS*

(a) Expected error of accumulative mean

$$E \left| \left(\overline{\cap} \right)_m - E \left[\left(\cap \right) \right] \right| = \sqrt{\frac{m-1}{m}} \frac{A}{\sqrt{T}}$$

(b) Associated confidence levels

$$\frac{\chi_{0.1}}{\sqrt{m}} \frac{A}{\sqrt{T}} \leq \left| \left(\overline{\cap} \right)_m - E \left[\left(\cap \right) \right] \right| \leq \frac{\chi_{0.9}}{\sqrt{m}} \frac{A}{\sqrt{T}}$$

(c) Confidence factors of χ^2 distribution

m	$\chi_{0.1}^2$	$\chi_{0.9}^2$	$\sqrt{\frac{m-1}{m}}$	$\chi_{0.1}/\sqrt{m}$	$\chi_{0.9}/\sqrt{m}$
2	0.0158	2.71	0.707	0.09	1.16
3	0.211	4.61	0.817	0.27	1.24
4	0.584	6.25	0.866	0.382	1.25
5	1.06	7.78	0.894	0.46	1.25
6	1.61	9.24	0.913	0.52	1.24
7	2.20	10.6	0.926	0.56	1.23
8	2.83	12.0	0.935	0.59	1.22
9	3.49	13.4	0.942	0.62	1.22
10	4.17	14.7	0.949	0.65	1.21
11	4.87	16.0	0.953	0.67	1.21
12	5.58	17.3	0.958	0.682	1.20
14	7.04	19.8	0.964	0.709	1.19
16	8.55	22.3	0.968	0.730	1.18
18	10.1	24.8	0.972	0.750	1.17
20	11.7	27.2	0.976	0.765	1.17
25	15.7	33.2	0.980	0.792	1.15
30	19.8	39.1	0.983	0.812	1.14
61	46.5	74.4	0.992	0.873	1.10
∞	m	m	1.0	1.0	1.0

*Computer estimate of A given by equation (8) Appendix C.

Table VII and equation (7) are being added to the programs CORFUN and MLTCOR such that the expected error curves

$$\frac{x_{0.1}}{\sqrt{m}} \frac{A}{\sqrt{T}} ; \sqrt{\frac{m-1}{m}} \frac{A}{\sqrt{T}} \text{ and } \frac{x_{0.9}}{\sqrt{m}} \frac{A}{\sqrt{T}}$$

may be added by the computer, to any accumulative error curve

$$\Delta(\overline{\quad})_m = f(T^{-1/2}).$$