AD-A111 923 DAYTON UNIV OH RESEARCH INST F/G 9/2

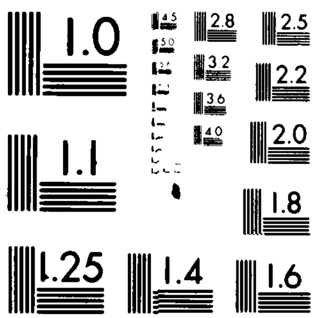
SOFEPL: A PLOTTING POSTPROCESSOR FOR 'SOFE' USER'S MANUAL.(U)

NOV 81 S H MUSICK, R E FELDMANN, J G JENSEN F33615-79-C-1770

UNCLASSIFIED AFWAL-TR-80-1109 NL

1 OF 1
AD A111923

END
DATE
FILMED
4-82
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

AFWAL-TR-80-1109

2

SOFEPL: A PLOTTING POSTPROCESSOR FOR 'SOFE'

USER'S MANUAL

UNIVERSITY OF DAYTON RESEARCH INSTITUTE
300 COLLEGE PARK AVENUE
DAYTON, OHIO 45469

November 1981

FINAL REPORT FOR PERIOD JULY 1979 - SEPTEMBER 1981

Approved for public release; distribution unlimited.

AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Stanton H. Musick

STANTON H. MUSICK
Project Engineer

Ronald L. Ringo

RONALD L. RINGO
Chief, Reference Systems Branch
System Avionics Division
Avionics Laboratory

FOR THE COMMANDER

Frank A. Scarpino

FRANK A. SCARPINO, Acting Chief
System Avionics Division
Avionics Laboratory

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AAAN-3 *, W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWAL-TR-80-1109	2. GOVT ACCESSION NO. AD-A111 923	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SOFEPL: A Plotting Postprocessor for 'SOFE', User's Manual		5. TYPE OF REPORT & PERIOD COVERED Final Report for Period July 1979 - Sept 1981 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Stanton H. Musick Richard E. Feldmann Jerry G. Jensen		8. CONTRACT OR GRANT NUMBER(s) F33615-79-C-1770
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Dayton Research Institute 300 College Park Dayton, Ohio 45469		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1206 01 20
11. CONTROLLING OFFICE NAME AND ADDRESS Avionics Laboratory (AFWAL/AAAN) Air Force Wright Aeronautical Laboratories, AFSC Wright-Patterson AFB, OH 45433		12. REPORT DATE November 1981 13. NUMBER OF PAGES 80
14. MONITORING AGENC NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Plot Postprocessor	rror Analysis
Computer Graphics	man Filter
DISSPLA	formance Analysis
Monte Carlo Simulation	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A program for performing statistical calculations and drawing line plots is developed, verified and documented. The program, named SOFEPL, is a postprocessor for SOFE, a general-purpose Monte Carlo simulation program designed to help synthesize a Kalman filter for an integrated system. SOFEPL is coded in FORTRAN and uses the DISSPLA graphics software. This report is the SOFEPL user's manual.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

BLANK PAGE

FOREWORD

This report describes the computer program SOFEPL and gives directions for using it. The report and program are products of a joint effort by Stanton H. Musick, Richard E. Feldmann, and Jerry G. Jensen.

This work was carried out at the Avionics Laboratory by S. H. Musick under Project Work Unit 1206 0120, and at University of Dayton Research Institute (UDRI) by R. E. Feldmann and J. G. Jensen under contract F33615-79-C-1770.

The authors wish to acknowledge Barbara O'Lear of UDRI for her assistance in program coding and debug.

BLANK PAGE

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

ABBREVIATIONS

ANSI	American National Standards Institute
CDC	Control Data Corporation
CRT	cathode ray tube
I/O	input/output
JCL	job control language
RMS	root mean square
SOFE	Simulation for Optimal Filter Evaluation
SOFEPL	SOFE Plotter
UDRI	University of Dayton Research Institute

KEY WORDS AND SYMBOLS

K	Number of samples in a run
N	Number of runs in a study
PLFILE	Device independent plot file
S	Standard deviation of a sample
TAPE4	SOFE/SOFEPL interface file
$(\underline{\ })$	Vector
$(\bar{\ })$	Arithmetic mean of a sample

SECTION 1
INTRODUCTION

SOFEPL is a computer program for drawing line plots of the output from SOFE, a Monte Carlo simulation program. SOFEPL (pronounced sō-fē-pē-el) stands for Simulation for Optimal Filter Evaluation Plotter. SOFEPL is flexible, efficient, and quickly accessible. This user's manual covers the function, structure and operation of the SOFEPL program. These same subjects are explained for SOFE in the SOFE user's manual [Reference 1], and are discussed only incidentally here.

Realizing the power inherent in Monte Carlo simulation requires an extensive statistical computation and plotting capability. Although SOFE offers a wide range of options for printing results, it has only rudimentary plotting facilities and no statistical computation abilities. These plotting and statistical functions are accomplished by the FORTRAN program SOFEPL, a companion postprocessor to SOFE. The interplay of SOFE and SOFEPL is illustrated in Figure 1.

This user's manual describes SOFEPL functions, documents the program's code structure, and provides instructions for and examples of SOFEPL use. Section 2 describes the problem that SOFEPL must solve and gives information defining its abilities. Section 3 is a collection of program notes describing such topics as program organization, data structures and coding conventions. Section 4 specifies the format and content of program inputs and outputs, and gives instructions for program use. In Section 5, representative samples of SOFEPL's capabilities are demonstrated for the orbit problem documented in [1]. Appendix A contains input and output for the example discussed in Section 5. Appendix B covers DISSPLA postprocessors. (DISSPLA is the

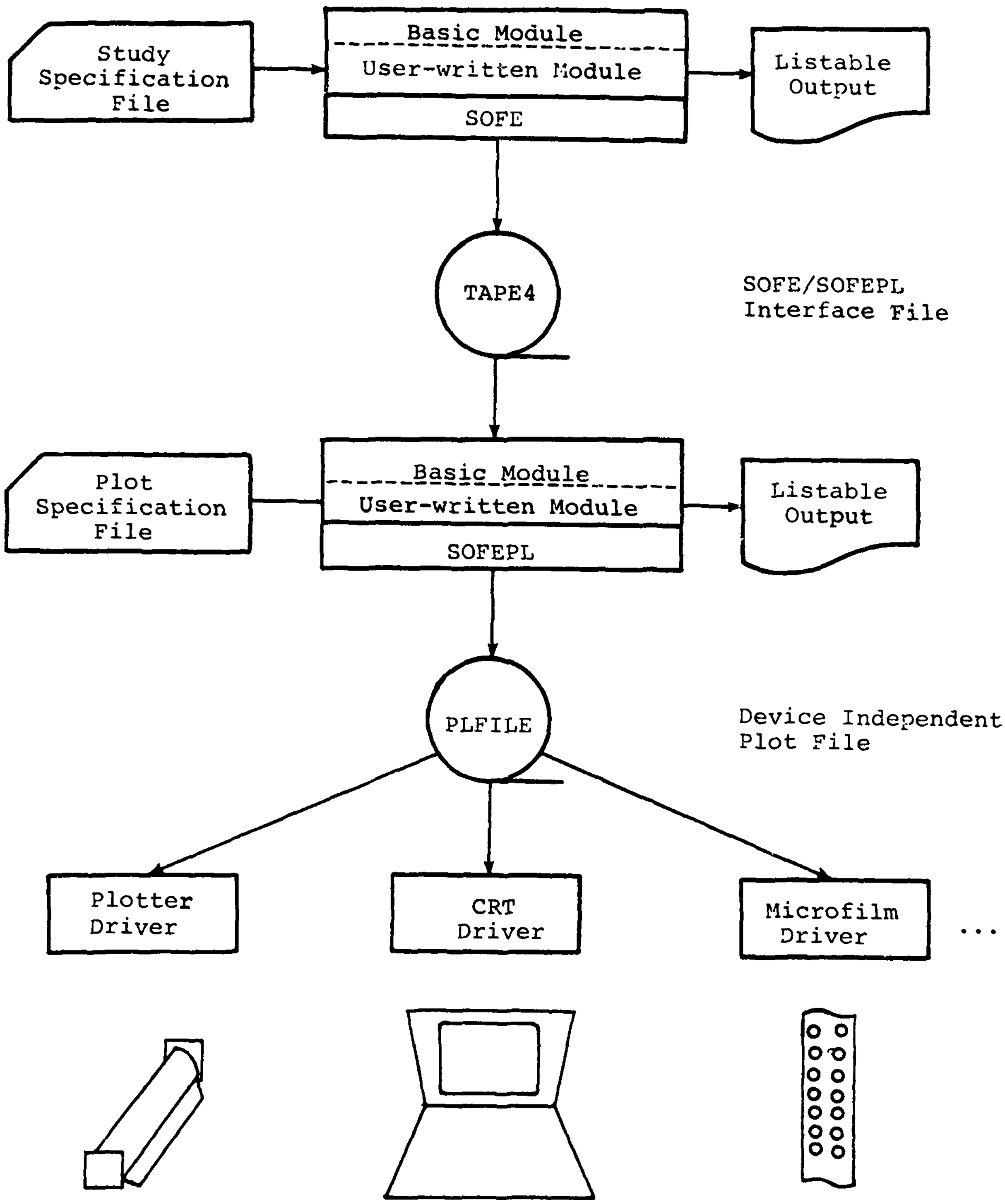


Figure 1. SOFE/SOFEPL System Diagram.

graphics software package that SOFEPL uses.) Appendix C describes two merge programs that combine multiple SOFE/SOFEPL interface files. Readers interested in exercising the program in minimum time should scan Section 2, skip Section 3, and concentrate on Sections 4 and 5.

BLANK PAGE

SECTION 2
GENERAL DESCRIPTION

SOFE is a general-purpose Monte Carlo simulation program that facilitates the synthesis of a Kalman filter for a multisensor system, and aids in the analysis of that system's performance. For a particular system configuration, which consists of both truth and filter models and measurement updates from various sensors, SOFE executes runs over a specified time interval to simulate missions.

As each simulation run develops, SOFE records it output on an unformatted interface file designated TAPE4. This file accumulates time-tagged samples of the five vector variables named in Table 1.

TABLE 1
VARIABLES RECORDED BY SOFE ON TAPE4

	Description of Vector Variable	Symbol	Dimension
1	Truth state	$\underline{X}s$	NS
2	Filter state estimate	$\underline{\hat{X}}f$	NF
3	Filter variances	$\underline{V}f$	NF
4	Measurement residuals	ZRES	M
5	Measurement residual variances	ALPHA	M

Note that Vf is the diagnoal of the filter covariance Pf. Thus each element Vf_i of Vf is the estimated error variance of the i^{th} filter state estimate $\hat{X}f_i$; i.e., $Vf_i = Pf_{ii}$.

The data of Table 1 is recorded at equally spaced intervals and at measurement update times. This sampling scheme permits the user to capture a complete history of each Monte Carlo run.

A set (ensemble) of runs, taken together, comprises a study. The statistical computations of SOFEPL are accomplished at each sample time across the ensemble of runs in the study. These statistical computations consist of simple averages and standard deviations on elements (and element differences) of the five variables listed in Table 1. SOFEPL plots these averages and standard deviations against time to facilitate analyzing the performance of the integrated system.

2.1 Average, Standard Deviation, Estimation Error

Figure 2 illustrates the average and standard deviation curves that SOFEPL draws. These curves derive from elementary statistical computations. To avoid misunderstandings concerning these computations, they are developed here.

Consider a continuous function y(t) which can have a different realization on each (SOFE) run. Figure 2 illustrates such a function for three runs, $y_1(t)$, $y_2(t)$, and $y_3(t)$. Of course, y(t) could be any element of any vector in Table 1. Assume that y(t) is sampled on each run at time t'. If there are a total of N runs in the study, the collection Y(t') of N samples at t' is

$$Y(t') = \left\{ y_1(t'), y_2(t'), \ldots, y_N(t') \right\} \tag{1}$$

where the subscripts n=1,2,...,N represent the run number. In Figure 2, N=3 runs. The arithmetic mean or average, $\bar{y}(t')$, of the quantities in the collection Y(t') is

$$\bar{y}(t') = \frac{1}{N} \sum_{n=1}^{N} y_n(t') \tag{2}$$

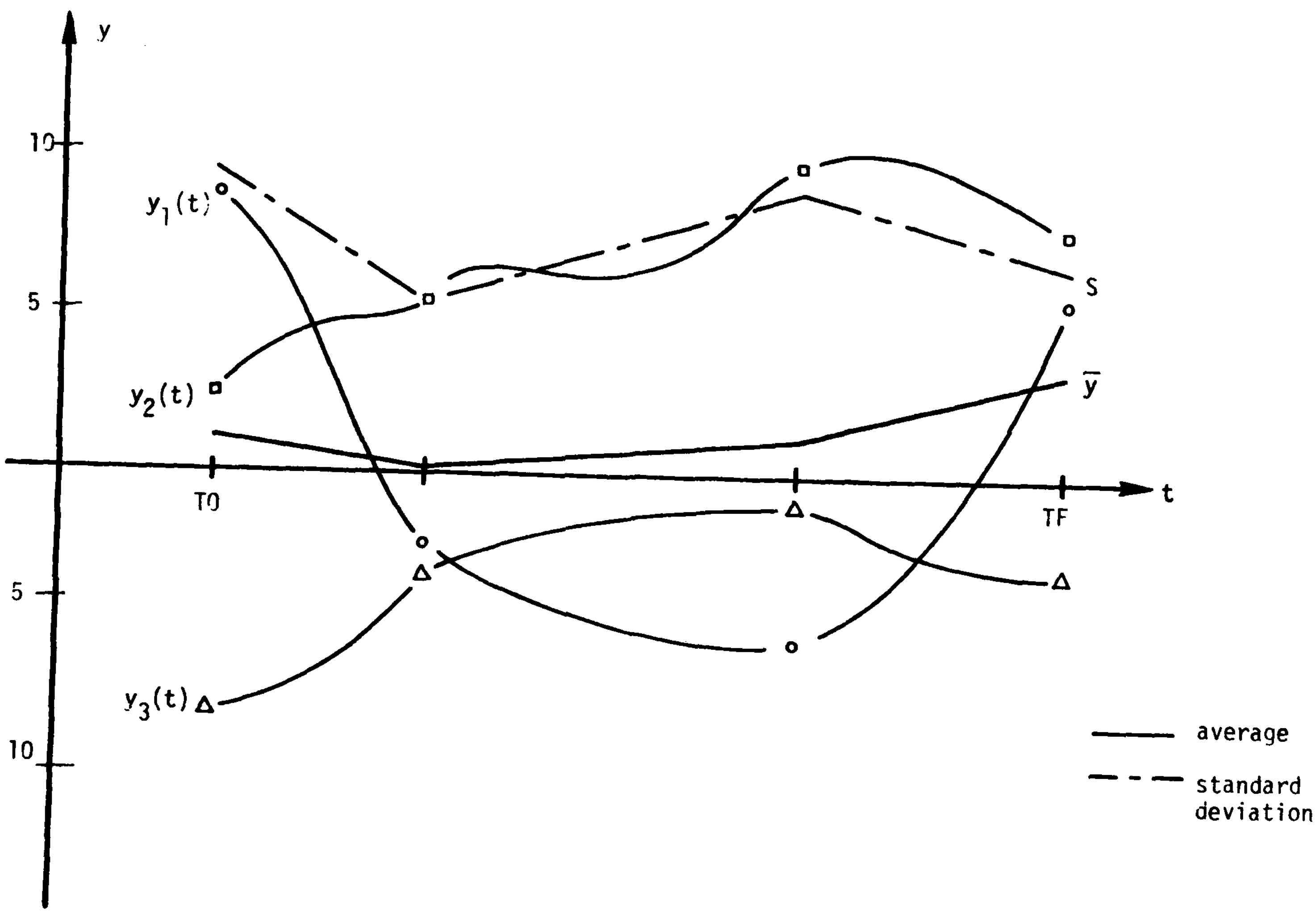


Figure 2. Average and Standard Deviation for Three Runs (○ □ △).

Define $\bar{y}(t_k)$ as the time-sequenced collection of all averages $\bar{y}(t')$ for all sample times t' in the closed interval [TO,TF] where TO and TF are the initial and final time of each run. If $\bar{y}(t_k)$ is shortened to $\bar{y}$ for notational simplicity, and if the sequence of sample times is $t_1, t_2, \ldots, t_K$, then $\bar{y}$ is

$$\bar{y} = \left\{ \bar{y}(t_1), \bar{y}(t_2), \ldots, \bar{y}(t_K) \right\} \tag{3}$$

When the point pairs $(t_k, \bar{y}(t_k))$ of (3) are computed, plotted, and connected in SOFEPL by straight line segments, they form a curve of ensemble averages of $y(t)$ samples. The dark solid line in Figure 2, connecting K=4 sample points, is such an average. In all that follows, the overbar notation will represent a time history as in (3) of averages formed over the runs in the study, each run weighted equally as in (2).

SOFEPL also computes a standard deviation, denoted here by a capital S. For the set of N samples in (1), the standard deviation is

$$S(t') = \left[1 + \frac{1}{4(N-1)} \right] \sqrt{V(t')} \tag{4}$$

where

$$V(t') = \frac{1}{N-1} \sum_{n=1}^{N} y_n^2(t') - \frac{N}{N-1} \bar{y}^2(t') \quad . \tag{5}$$

Here $V(t')$ is the familiar variance computation. Equation (4) is an approximate result that appears in Reference 2, page 76. A full derivation of the exact result appears in Reference 3, Chapter 15. The advantage of (4) is that it produces an unbiased estimate of $S(t')$, especially when $N \geq 9$ and/or when $y(t')$ is a sample from a normal or nearly normal distribution. These latter two conditions, however, need not be rigorously satisfied.

The discrete-time function $S(t_k)$, or S for short, is the time-sequenced collection of all standard deviations $S(t')$ for t' in [TO,TF]. Thus, in analogy with (3),

$$S = \left\{ S(t_1), S(t_2), \ \ldots \ , S(t_K) \right\} \tag{6}$$

The dash-dot line in Figure 2 is the S curve of y(t) for the three runs shown. Note that S is not defined for N=1 run.

Define E_{ij} as the estimation error for the j^{th} state in the estimated state vector $\hat{\underline{X}}f$. Then E_{ij} is

$$E_{ij} = Xs_i - \hat{X}f_j \tag{7}$$

where Xs_i and $\hat{X}f_j$ are the i^{th} and j^{th} elements of $\underline{X}s$ and $\hat{\underline{X}}f$, respectively. In (7), Xs_i is chosen to force the comparison with $\hat{X}f_j$ to be between the same physical quantities. SOFEPL provides a means for forming any difference E_{ij}, $1<i<NS$, $1<j<NF$.

2.2 Available Plots

Table 2 lists the 18 plots that SOFEPL can draw. All plots are time histories of the variables shown in the table, all averages are ensemble averages per (3), and all standard deviations are formed per (6). A representative set of plots from Table 2 is described in the following paragraphs.

In plot type 1, the i^{th} state of the truth state vector $\underline{X}s$ is averaged over the ensemble of runs in the study and plotted through time to produce a single curve. In type 3, the average of the i^{th} state of $\underline{X}s$ forms one curve while the average of the j^{th} state of $\hat{\underline{X}}f$ forms the other.

In type 6, the average of the difference E_{ij} of truth state i and filter state j (see (7)) forms one curve; the standard deviation SE_{ij} of this difference, as prescribed by (6), is the second curve, and $-SE_{ij}$ is the third curve. A type 9 plot is the average of the square root of the i^{th} diagonal element Vf_i of the filter covariance Pf; this is the filter's uncertainty about its

TABLE 2
AVAILABLE PLOT TYPES

Plot Type	No. of Curves	Description of Curves		
1	1	$\bar{X}s_i$		
2	1	$\hat{\bar{X}}f_i$		
3	2	$\bar{X}s_i$	$\hat{\bar{X}}f_j$	
4	3	$\hat{\bar{X}}f_i$	$\hat{\bar{X}}f_i+\overline{\sqrt{Vf_i}}$	$\hat{\bar{X}}f_i-\overline{\sqrt{Vf_i}}$
5	1	$\bar{E}_{ij}$		
6	3	$\bar{E}_{ij}$	$+SE_{ij}$	$-SE_{ij}$
7	3	$\bar{E}_{ij}$	$\bar{E}_{ij}+SE_{ij}$	$\bar{E}_{ij}-SE_{ij}$
8	3	$\bar{E}_{ij}$	$+\overline{\sqrt{Vf_j}}$	$-\overline{\sqrt{Vf_j}}$
9	1	$\overline{\sqrt{Vf_i}}$		
10	1	SE_{ij}		
11	1	$\sqrt{SE_{ij}^{2}+\bar{E}_{ij}^{2}}$		
12	2	SE_{ij}	$\overline{\sqrt{Vf_j}}$	
13	2	$\sqrt{SE_{ij}^{2}+\bar{E}_{ij}^{2}}$	$\overline{\sqrt{Vf_j}}$	
14	5	$\bar{E}_{ij}$	$\bar{E}_{ij} \pm SE_{ij}$	$\pm\overline{\sqrt{Vf_j}}$
15	1	$\overline{ZRES_i}$		
16	3	$\overline{ZRES_i}$	$+\overline{\sqrt{ALPHA_i}}$	$-\overline{\sqrt{ALPHA_i}}$
17	2	SZ_i	$\overline{\sqrt{ALPHA_i}}$	
18	NCRV	User defined using arrays P1-P10		

estimate $\hat{X}fi$. Note that standard deviations $\sqrt{Vf_i}$, not variances Vf_i, are averaged. Plot type 11 is the root mean square (RMS) value of the estimation error associated with filter state $\hat{X}f_j$.

In type 16, the average of the i^{th} of M residuals forms one curve. The remaining two curves are plus and minus the average of the square root of the i^{th} element in the vector ALPHA. $ALPHA_i$ is the filter's predicted uncertainty for measurement residual $ZRES_i$. Plot type 17 compares this predicted value to SZ_i, the actual standard deviation of the $ZRES_i$ data over the N runs in the study. SZ_i is computed using (6), as was SE_{ij}.

Plot type 18 is a user-defined plot. Whereas types 1 through 17 are entirely determined by the parameters in the user's input data, type 18 is formed in large part by FORTRAN code constructed by the user. Type 18 is discussed in detail in Subsection 4.2.

2.3 Other Attributes

SOFEPL is written in FORTRAN and was developed on Control Data Corporation (CDC) equipment where it loads in approximately 115,000 octal words of memory. A concerted effort has been made to use programming conventions, 1966 ANSI constructs, modular concepts, and informative comments that will expedite the transfer of the program to other machines. In addition, many measures have been taken to make the program efficient in its use of central memory and execution time.

SOFEPL calls subroutines in the graphics package DISSPLA for the actual plot generation. The output of this package is a compact file of plotting commands in a special format. This file, called PLFILE, is then submitted to a DISSPLA postprocessor program to make the actual plots on the desired plotting device. While it may seem redundant for a postprocessor (SOFEPL) to require a second postprocessor to complete the final product, this approach separates plot file generation from plotting, and thereby provides flexibility.

The following peripherals are required to run SOFEPL:

- Card reader or data input terminal to enter input parameters.
- A tape or disk file to supply TAPE4 data from SOFE.
- A line printer to print listable output.
- A tape or disk file to store plot commands (PLFILE) for subsequent plotting by the DISSPLA postprocessor.

DISSPLA postprocessors in turn usually require a similar set of peripherals for reading the plot commands and writing to the plotting device.

SECTION 3

PROGRAM NOTES

SOFEPL is a tool for plotting results from a series of SOFE runs. From a user's point of view, the program is flexible in the kind of plots it can make, in the choice of runs it uses, and in its presentation format. In addition, SOFEPL executes quickly and uses core efficiently. This section describes the code structure and some key features of the program design that contribute to accomplishing the goals listed above.

3.1 Organization

SOFEPL is organized into five major sections, each section representing a phase in the process of creating a plot(s) from SOFE output. These five sections are designated as follows:

(1) Program initialization,
(2) Plot selection,
(3) Data accumulation,
(4) Plot execution,
(5) Program completion.

The organization of these five tasks is shown in the macro-level flowchart, Figure 3.

Program initialization creates data and storage areas inside SOFEPL based on information from the header of the interface file, TAPE4. Plot selection reads the user's input cards which define each plot desired and sets aside the necessary storage areas in memory. The data accumulation section reads data from TAPE4 and stores that data in the appropriate storage areas. Plot execution calculates the desired curves from the stored data and constructs a file containing the requested plots. Program completion closes the plot file and ends the run.

SOFEPL accomodates plots with varying numbers of curves and points by employing a 'spooling' technique for data storage.

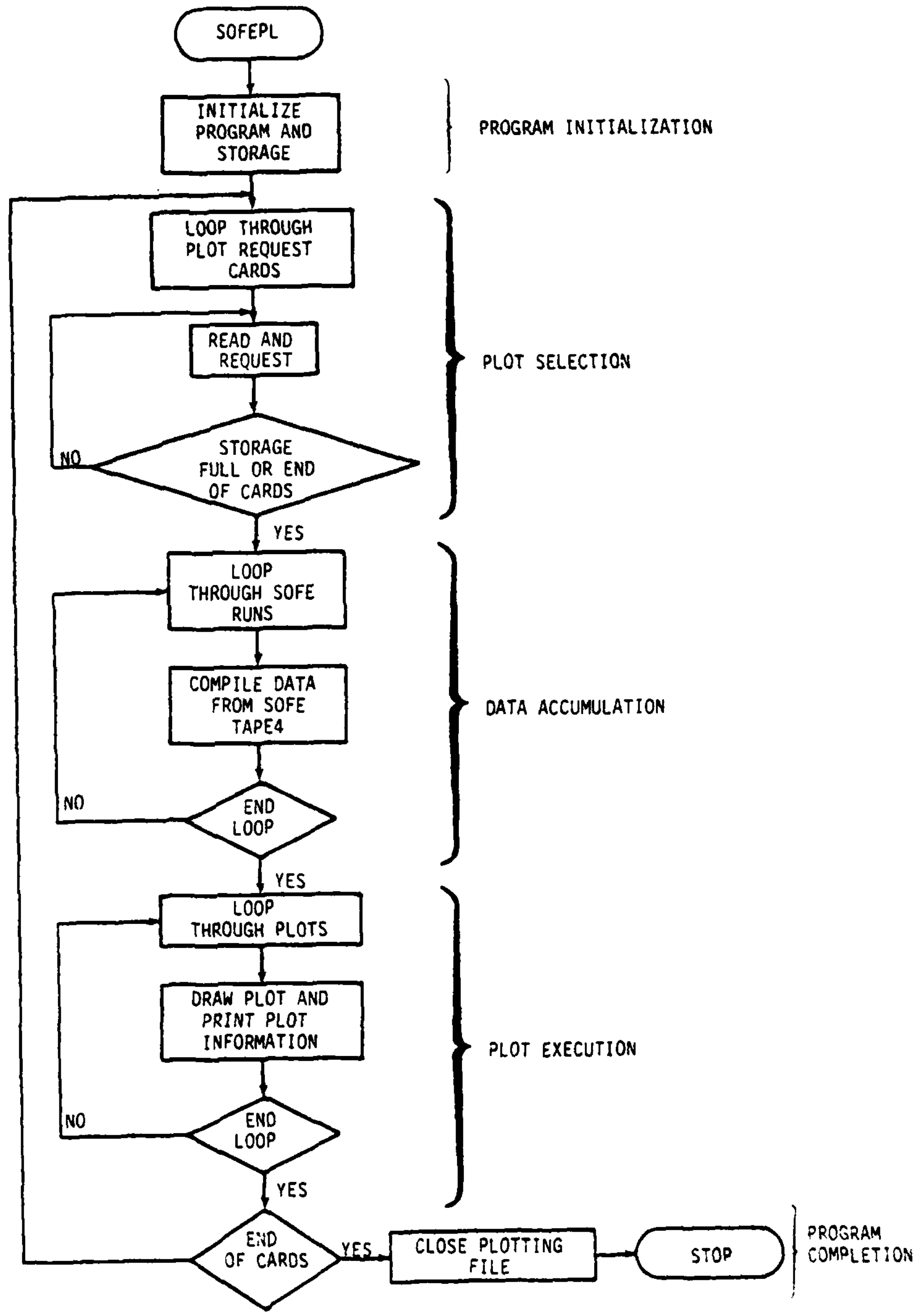


Figure 3. Macro-Level Flowchart

With this technique, the amount of storage allocated to each plot is exactly what is needed for that plot and waste of core storage is held to a minimum.

SOFEPL's spooling technique works as follows. The spool is a single linear array named 'A' of dimension 10000. As each plot request is read, it is interpreted (in light of information from TAPE4's header) to determine this plot's storage requirements. Blocks of storage are then set aside in A to hold the points for each curve as well as other necessary data. These blocks are accessed using an index of pointers. Furthermore, the block for plot n is adjacent to that for n-1 and n+1 with no unused words between them; thus loss of core storage is minimized.

When more than one plot is requested, execution time is also reduced by use of the spooling technique. This occurs because several plots are being made on each pass through TAPE4, thereby reducing the number of times TAPE4 is read and saving I/O time. Clearly, if the spool was long enough, all plots could be made in one pass through TAPE4. The tradeoff is between the cost of core storage and the cost of I/O time. This tradeoff is approximately correct for the CDC system with A dimensioned 10000. This dimension is easily changed for another system, or for plots with huge numbers of points, by altering a COMMON card and a DATA card in SOFEPL's main routine.

3.2 Code Structure

SOFEPL consists of a main executive, 18 subroutine subprograms, and two function subprograms. Additionally, SOFEPL references nine DISSPLA subroutines, seven FORTRAN library functions, and three system utility functions. The main executive sequences the program through its five sections by calling on the appropriate routines for each plot-creation phase.

Figure 4 indicates the flow of SOFEPL through its routines. This figure illustrates which routines accomplish

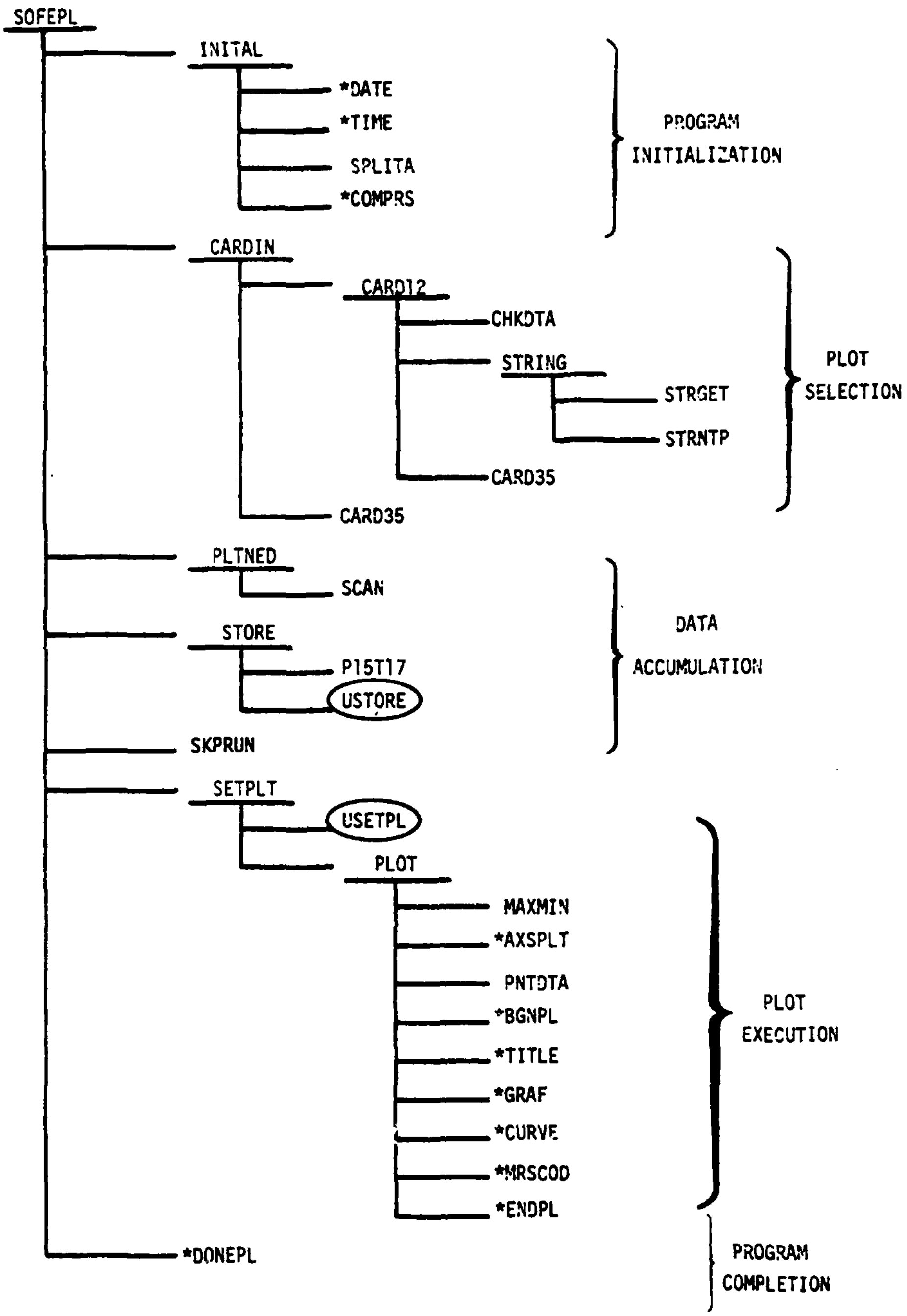


Figure 4. Subroutine Calling Sequence.

which solution phase and defines dependencies in the collection of routines. None of the FORTRAN library functions that SOFEPL uses are shown in Figure 4. A subprogram name prefixed by an asterisk indicates a DISSPLA subroutine or system utility function, while the remaining names are SOFEPL routines.

The first section, program initialization, begins with a call to INITAL where the TAPE4 header is read to acquire data that will allow arrays to be sized properly. INITAL then writes the SOFEPL header page and calls SPLITA where 'offset pointers' are assigned in A to locate quantities and data blocks. INITAL checks to be sure that the spool size of 10000 words is large enough to handle the user's problem; if not, a warning message is written and the program is STOPed. Finally, INITAL initializes the DISSPLA software by calling COMPRS.

During the second phase, plot selection, subroutine CARDIN takes over and calls on CARD12 to read cards 1 and 2, and on CARD35 to read cards 3, 4, and 5. Subroutine CHKDTA is called by CARD12 to range-check the parameter values read from card 1, the plot selection card. Subroutine STRING is called by CARD12 to acquire the desired runs from card 2, the run selection card. CARD35 reads the label data for the plot and for the y- and t-axes. The routines in this section are iterated until the spool is filled with plots or until all plot sets are read, whichever comes first.

When, in the data accumulation phase, a new run is ready for processing, subroutines PLTNED and SCAN are called to determine which plots require the data from this run. If no plots require this run's data, SKPRUN is called to skip over the run and by-pass its data entirely. Otherwise STORE is used in conjunction with USTORE and P15T17 to read and accumulate the data in the correct array locations.

The fourth section, plot execution, uses SETPLT in conjunction with USETPL to form averages and standard deviations from the accumulated data arrays. PLOT is then called to draw the requested plots. Seven of the nine calls to DISSPLA routines occur in PLOT.

DONEPL is called as the final act of SOFEPL execution to close PLFILE. This last phase and the first phase occur only once for each SOFEPL run, while the middle three phases iterate until the entire plot specification file is processed.

3.3 Coding Conventions

SOFEPL is written in FORTRAN using primarily 1966 ANSI standard constructs. Exceptions include use of quoted comments after STOPs, arithmetic expressions in array arguments, list directed input, IMPLICIT statements, quoted hollerith data in FORMATs, and a few others common to most FORTRAN compilers.

In writing SOFEPL we attempted to use clear, easily-understood code that is organized in top-down fashion. Nearly 40 percent of the statements are comments that are designed to be concise, informative, and accurate. SOFEPL uses only single precision variables of the REAL, INTEGER, and LOGICAL type. Variables are given meaningful names from the 36 alphanumeric characters. The first letter of a variable's name conforms to the FORTRAN default rule for integer and real names. (STRING, STRGET, and STRNTP are exceptions where all variables are forced to be integers through IMPLICIT statements.) Counters generally start with I or J, pointers with K, logical variables and labels with L, and sizes and dimensions with M or N. All format statements are located at the end of routines.

3.4 SOFEPL Data Structures

SOFEPL uses two major data structures. The first is the SOFE/SOFEPL interface file, TAPE4. The other is the A/IA array used to store the required data for the selected plots.

3.4.1 SOFE/SOFEPL Interface File

The SOFE/SOFEPL interface file is an unformatted sequential file written in internal binary representation by SOFE. This file, called TAPE4, contains a header record followed by IPASS subfiles, where IPASS is the number of SOFE simulation runs. Each subfile (referred to in subsequent discussion as just a "file") is separated from its neighbor by an end-of-file mark. Figure 5 shows the overall structure of TAPE4.

The header record of TAPE4 contains data pertaining to the originating SOFE run, namely the values of NS, NF, M, TO, TF, TMEASO, DTMEAS, DTCCPL, IPASS, run date, run time, and run title. This record is read and echoed in INITAL. Each file on the interface tape contains a series of variable length records. In turn each record is comprised of vector variables sampled at an event time as specified in Table 3. The records for file one are output during SOFE run one, those from file two during run two, and so on. The event times, which are determined by SOFE parameter values, are identical in every run. Records are read and processed in subroutine STORE where the read process is aided by the fact that the record's length is its first word. Note in Table 3 that t_i may equal t_k^-. When this occurs, a double sample of $\underline{X}s$, $\hat{\underline{X}}f$ and $\underline{V}f$ occurs at t_i^- thereby causing these points to be double plotted.

3.4.2 The A/IA Array

Because the A array is equivalenced to the IA array so that both integers and real values are readily accessible, the name is A/IA. A/IA was referred to earlier as the spool or spooling area. Most of the variables and arrays necessary for generation of plots are stored in this array. These variables and arrays are referenced by means of an index of pointers.

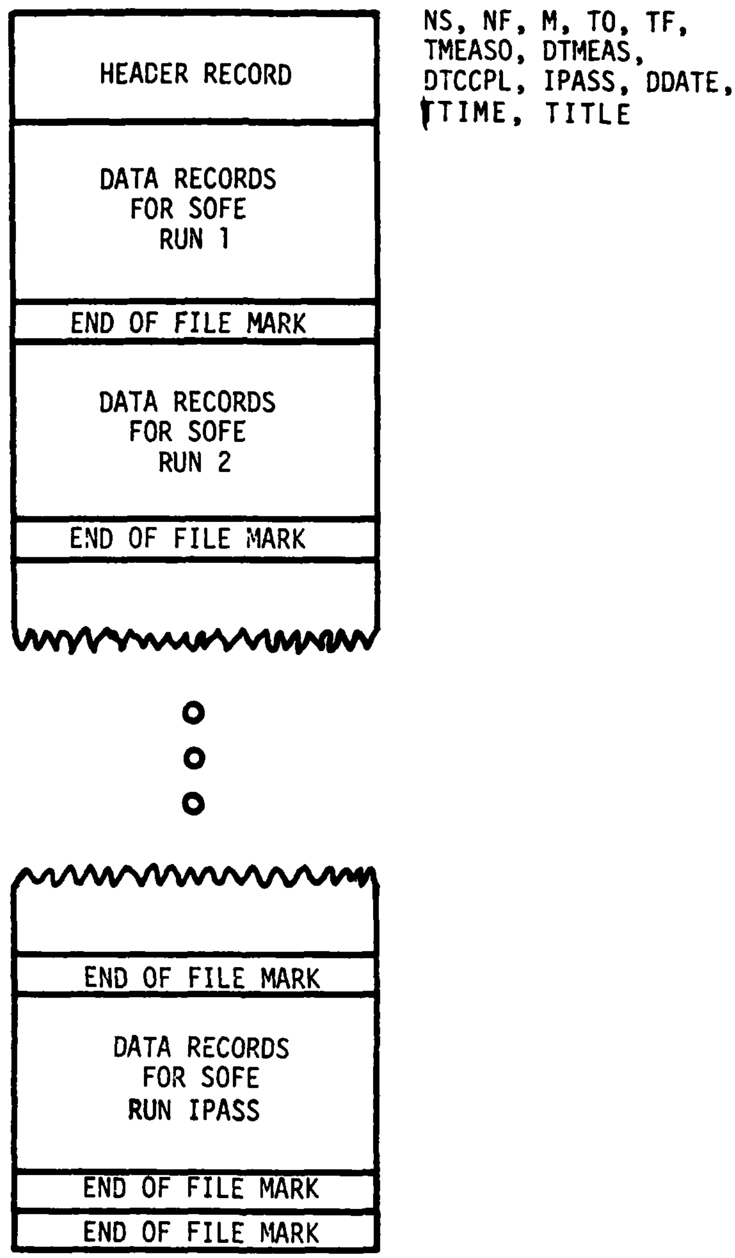


Figure 5. TAPE4 Structure.

TABLE 3

TIMING OF RECORDING OF VECTOR VARIABLES ON TAPE4

Vectors Recorded	Time of Sample		Number of Samples Per Run
	Symbol	Description	
$\underline{Xs}$ $\underline{\hat{X}f}$ $\underline{Vf}$	TO	Initial time	1
$\underline{Xs}$ $\underline{\hat{X}f}$ $\underline{Vf}$	t_k	Fixed sampling interval (=DTCCPL)	$\cong \frac{(TF-TO)}{DTCCPL}$
$\underline{Xs}$ $\underline{\hat{X}f}$ $\underline{Vf}$	t_i^-	Before measurement incorporation	$\cong \frac{(TF-TMEASO)}{DTMEAS}$
$\underline{Xs}$ $\underline{\hat{X}f}$ $\underline{Vf}$	t_i^+	After measurement incorporation	"
$\underline{Xs}$ $\underline{\hat{X}f}$ $\underline{Vf}$ ZRES ALPHA	t_i^{+c}	After measurement incorporation and feedback correction	"
$\underline{Xs}$ $\underline{\hat{X}f}$ $\underline{Vf}$	TF	Final time	1

Figure 6 shows the general structure of the A/IA array. The number of plots for which data can be stored is determined by the dimension of A/IA, by the curves per plot, and by the number of points per curve. If there is more storage available in A/IA than necessary, then only one pass through the interface file is made. If more plots are requested than can be accomodated in the A/IA array on one pass, then plot data for as many plots as will fit is stored. When plot set reading is complete, the program makes a pass through the interface file to generate the actual plots. When the first set of plots is finished, the data in A/IA for these plots is 'dumped' so SOFEPL can reuse this storage space for the next set of plots, and so on until all plot requests are processed.

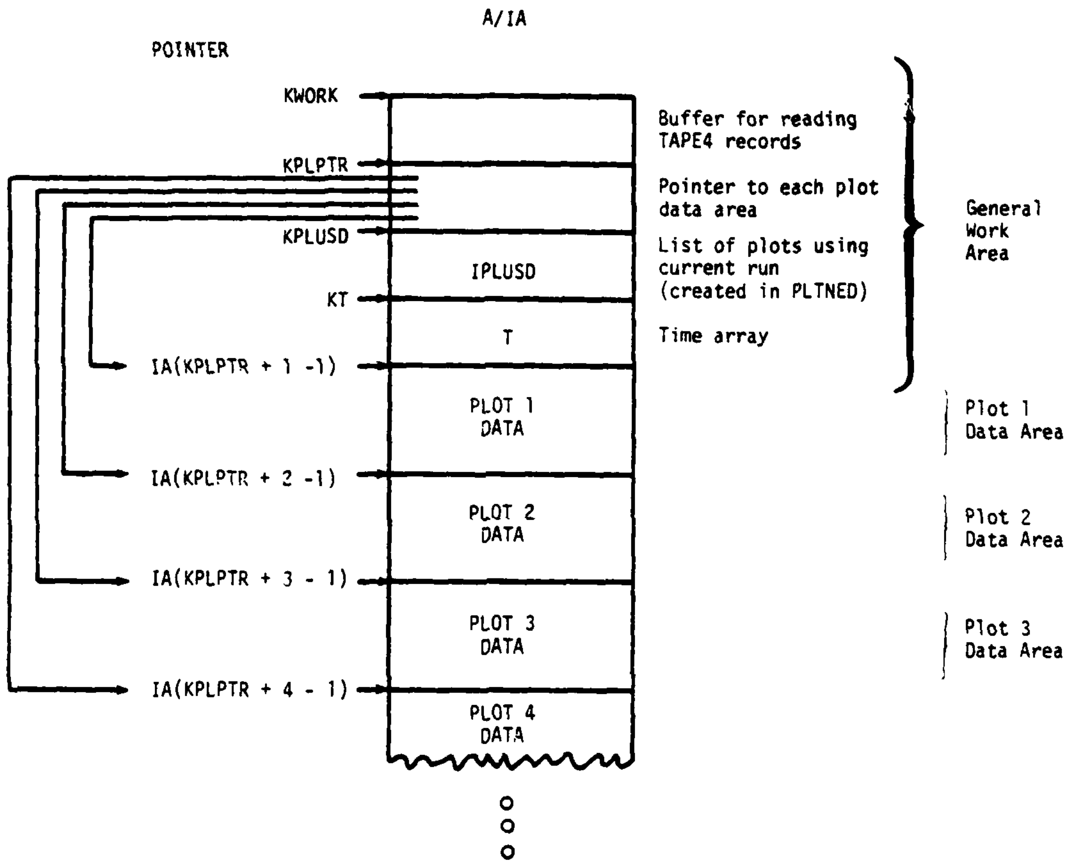


Figure 6. Structure of the A/IA Array.

The first section of the A/IA array is the general work area. Figure 6 shows the internal structure of the general work area and its relationship to the rest of the array. Figure 7 shows the structure within a plot data area. It should be noted in Figure 7 that the arrays Y1-Y5 and P1-P10 are allocated storage only for the plot types for which they are needed. Storage is allocated by adjusting the array pointers in subroutine CARDIN. For example, suppose a plot type 4 is requested, requiring three Y arrays, Y1, Y2, Y3. Suppose that after allocating storage for the preceding variables, the storage for the Y arrays starts at A/IA(1000). This means that the last value in ISETS1/ISETS2 is at location A/IA(999). Also suppose that a run on the interface tape contains 150 data points. Since the storage for the Y arrays begins at A/IA(1000), SOFEPL would place 1000 in the variable KY1, 1150 in KY2 and 1300 in KY3. KY1, KY2, and KY3 now point to the locations of Y1, Y2, and Y3 in the A/IA array. Values of 1300 would also be placed in KY4, KY5, KP1, KP2,...,KP10, thus implying that these arrays contain zero storage. If, for example, SOFEPL needed to access the tenth point in Y2, the following code would be used:

$$\text{(value of point \#10 in Y2)} = \text{A}\left(\text{IA(KLOCAT + KY2)} + 10 - 1\right)$$

where

$$\text{KLOCAT} = \text{KPLPTR} + \text{IPLOT} - 1.$$

The complexity of the pointers is the price paid for program versatility and efficiency.

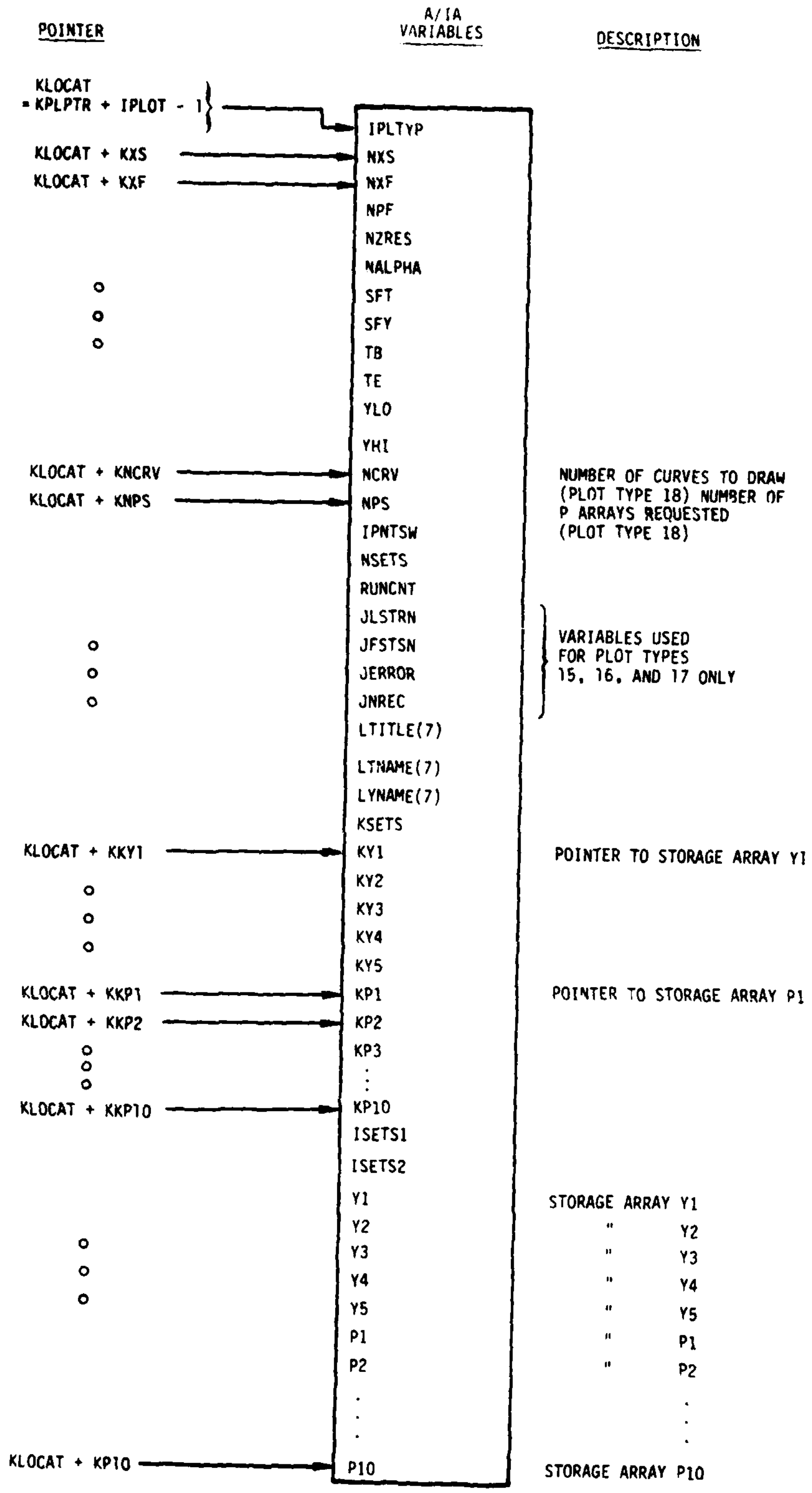


Figure 7. Detailed Structure of a Plot Data Area.

SECTION 4

MAKING PLOTS WITH SOFEPL

The procedure for obtaining SOFEPL plots is illustrated in Figure 1 and summarized here:

(1) Run SOFE with control parameter LCC made true and save TAPE4, the SOFE/SOFEPL interface file.

(2) Construct a SOFEPL 'plot specification file'.

(3) Write user routines USTORE and USETPL if a type 18 plot is desired.

(4) Run SOFEPL using TAPE4 and the plot specification file as inputs. Save the output file PLFILE.

(5) Make plots on a plotting device using PLFILE as input to the DISSPLA postprocessor for that device.

Step 1, the process of running SOFE and saving TAPE4, is discussed thoroughly in [1]. Since TAPE4 can be a very large file, the user must exercise good judgement in his/her choice of media for saving it. Magnetic tape is usually least expensive while disk is most expensive. Steps 2, 3, and 4 of the above procedure are discussed next in Subsections 4.1, 4.2, and 4.3. Step 5 instructions are given in Appendix B.

4.1 The SOFEPL Plot Specification File

The SOFEPL plot specification file determines which plots are produced by SOFEPL. This file takes the form of user-supplied data cards, or card images. The plot specification file consists of a run title card followed by any number of five-card groups (called plot sets) which specify each particular plot. That is, SOFEPL produces one plot for each plot set submitted by the user through the plot specification file. Figure 8 shows the overall structure of the plot specification file. As this figure

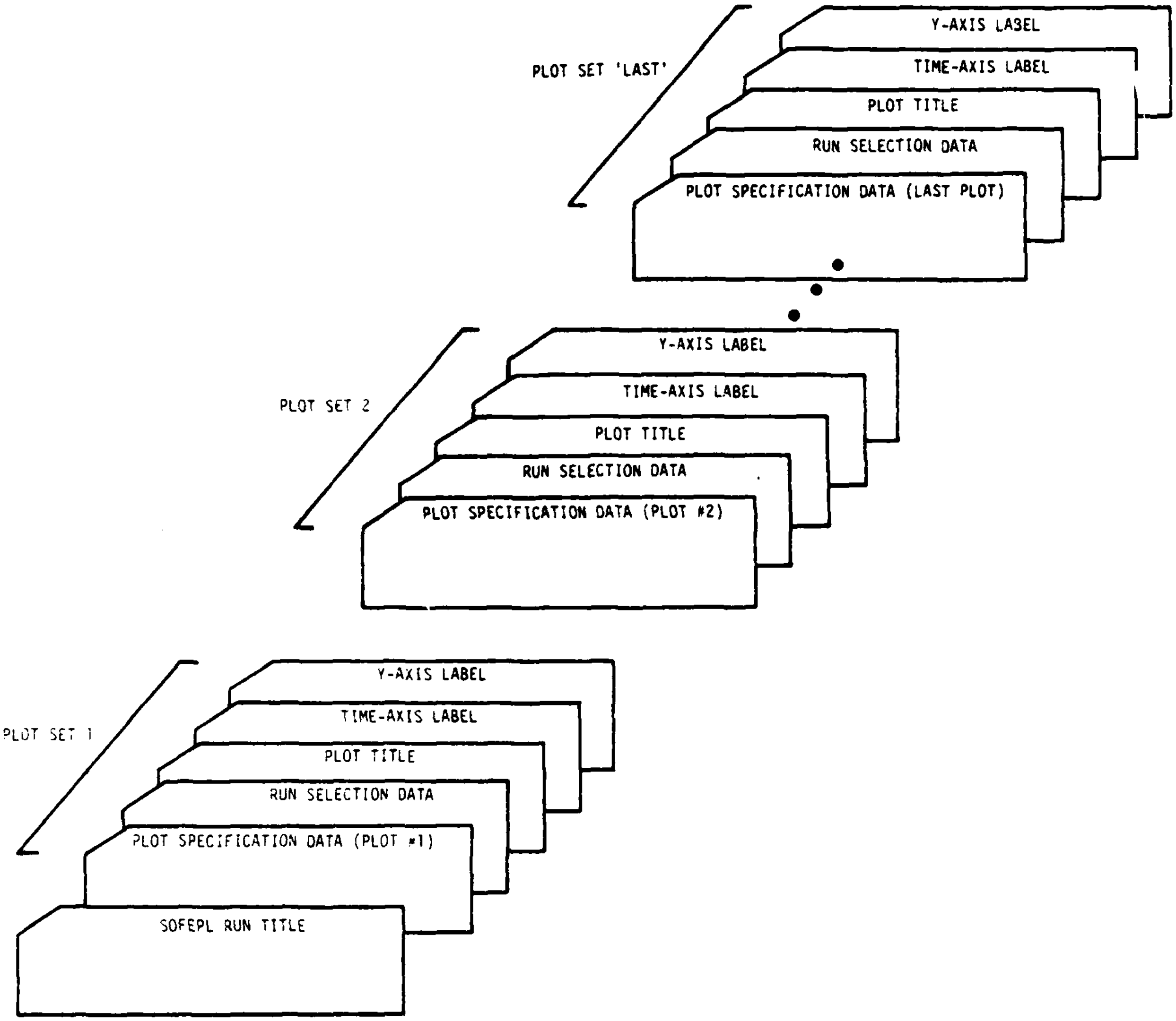


Figure 8. Structure of SOFEPL Plot Specification File.

shows, the plot specification file contains these six types of data cards:

- Run Title Card
- Plot Specification Card
- Run Selection Card
- Plot Title Card
- Time Axis Label Card
- Y-Axis Label Card.

The next four subsections describe these cards in detail.

4.1.1 SOFEPL Run Title Card

The first card of the plot specification file is the run title card. This card is used to input a title that characterizes the upcoming SOFEPL run. This title appears only on the printed output provided by SOFEPL. Thus this title can be used to match the printed output with a plot specification file. The run title card can contain up to 80 characters. This card has no format or character type restrictions.

4.1.2 Plot Specification Card

The first card of a plot set is the plot specification card. This card is used to input the 15 parameters that define the particular plot to be drawn. These parameters are listed in Table 4; their functions will be illustrated in Section 5.

The parameter values on this card are read by SOFEPL using CDC list directed format. Values input under this CDC free-form convention must be in proper order (that shown in Table 4) but need not reside in preassigned columns on the card. In addition, multiple quantities may be entered on a single card separated only by commas or blanks. To illustrate, one could enter values of 14,1,3,3 for IPLTYP, NXS, NXF, NVF in any of these ways:

```
14,1,3,3/     or     14 1 3 3/     or
14,,2*3/      or     14, ,3,3/     or
14,,3 3/      or     ...
```

TABLE 4

PLOT SPECIFICATION CARD PARAMETERS

Position on Card	Variable Name	Variable Definition	Limits	Default Value	Data Type
1	IPLTYP	Plot Type (Table 2)	1-18	none	Integer
2	NXS	Truth State Element Number	1-NS	1	Integer
3	NXF	Filter State Element Number	1-NF	1	Integer
4	NVF	Covariance Matrix Diagonal Element Number	1-NF	1	Integer
5	NZRES	Measurement Residual Number	1-M	max(1,NALPHA)	Integer
6	NALPHA	Measurement Residual Variance Number	1-M	NZRES	Integer
7	IPNTSW	Print Switch (1 = print on)	0 or 1	0	Integer
8	SFT	T-Axis Scale Factor	none	1.0	Real
9	SFY	Y-Axis Scale Factor	none	1.0	Real
10	TB	T-Axis Min. Value	none	min t	Real
11	TE	T-Axis Max. Value	none	max t	Real
12	YLO	Y-Axis Min. Value	none	min y	Real
13	YHI	Y-Axis Max. Value	none	max y	Real
14	NCURV	Number of Curves to Be Drawn (Plot Type 18 Only)	1-5	0	Integer
15	NPS	Number of P Arrays Used (Plot Type 18 Only)	NCRV-10	0	Integer

Note that a pair of commas separated by an optional blank(s) has the effect of skipping the reading of the parameter at that location; thus this parameter is assigned its default value which is 1 for NXS (see Table 4). Also note the repetition factor in the third version. The slash at the end of each version instructs the program to discontinue reading the plot specification card; the parameters remaining in the input list then acquire their default values. Real numbers are entered using any valid FORTRAN format. The user can consult his/her FORTRAN manual for additional list directed information but the examples above, coupled with those for the sample in Section 5, should be adequate for most purposes.

A few things should be mentioned about the scale factor and axis limit parameters (SFT, SFY, TB, TE, YLO, YHI). When time limits (TB and/or TE) are specified, the points lying outside the specified range do not appear on the plot. These points are, however, transfered to user-defined subroutine USTORE if a type 18 plot is in process. The same is true for plots with y-axis limits. All points are used in the various SOFEPL calculations. The out-of-bounds points are deleted by the DISSPLA plotting routines which are the last called. A list of the out-of-bounds points is generated in SOFEPL printed output. When entering both a scale factor and axis limits, the axis limits should be in the units of the rescaled data.

4.1.3 The Run Selection Card

The second card of a plot set is the run selection card. This card supplies a list of the SOFE simulation runs that are to be used in forming averages and standard deviations for this plot. This card contains a list of data items separated by commas. These items can either be integers repereseting particular runs, or integer pairs, separated by a dash, representing an inclusive set of SOFE runs. The letter 'L' can also be used in place of the integer value of the last SOFE run. For example, if there were 17 runs on TAPE4, a card reading 1, 5-9, 11, 12, 15-L would instruct SOFEPL to use SOFE runs 1, 5, 6, 7, 8, 9, 11, 12, 15, 16, and 17. In this case N, the number of runs in the study, is 11. Note that the list of data items must be contained on one card and must increase positively across the card. Blanks on this card are ignored.

4.1.4 Plot, Time-Axis, and Y-Axis Title Cards

The final three cards of the five-card plot set are the plot title, the time-axis label, and the y-axis label. Like the run title card, these cards are read with a standard FORTRAN formatted read. The titles on these cards may be up to 60 characters long and should be composed from the standard alphanumeric characters, A-Z and 1-9, plus the following nine special characters: ' , . + - * / ( ). A $ placed after any title will cause the title to be centered in the appropriate space.

4.2 Using Plot Type 18

Plot type 18 enables the user to construct new variables from the elements of $\underline{X}s$, $\hat{\underline{X}}f$ and $\underline{V}f$, and then to draw time histories of up to five such variables on a single plot. In the example to be presented in Section 5, polar coordinates are converted to rectangular coordinates through the sine and cosine relationships. The range of possibilities for plot type 18 is obviously very broad. It is hoped that this capability will satisfy most of the unusual needs not covered by types 1-17.

The user defines the variables that SOFEPL will plot by writing subroutines USTORE and USETPL. These routines are only stubs in SOFEPL's code until the user inserts executable code in them. USTORE operates as the variable computation routine. Each time new samples of $\underline{X}s$, $\hat{\underline{X}}f$, and $\underline{V}f$ are read from TAPE4, USTORE is called with those three vector samples as formal parameters. K such calls to USTORE occur for each selected run and N*K calls occur for a study. (Note that K and N are denoted KREC and RUNCNT in the stubs for USTORE and USETPL.) In USTORE, the user computes the variables he needs (up to ten) and saves them in appropriate arrays. Often he will be saving sums and sums-squared of these variables in order to form averages and standard deviations in the final plots.

After all N runs have been processed through USTORE, SOFEPL calls USETPL, thereby affording the user an opportunity for some final computations and/or rearrangements before plotting begins. As stated above, this is where averages and standard deviations should be computed by applying (2) and (4) at each sample time. Since all of SOFEPL's abilities for axis scaling and windowing remain in effect for plot type 18, USETPL is unburdened from these duties.

SOFEPL provides the type 18 user with up to ten scratch arrays (P1, P2, ... , P10). The actual number of arrays provided is NPS, an integer entered on the plot specification card especially for plot type 18. These arrays are dimensioned K long, large enough to hold one value for every sample time in a SOFE run. Curves can be drawn from the first five of these P arrays. The actual number of curves drawn is NCRV, an integer also entered on the plot specification card, especially for this plot type. Additional storage arrays can be obtained via additional DIMENSION statements within the USTORE and USETPL routines.

The definitions for the formal parameters in the argument lists of USTORE and USETPL are given in Tables 5 and 6. It should be noted that all ten P arrays are named in the argument lists of both USTORE and USETPL even though the user has requested only NPS of them (NPS $\leq$ 10). NPS is used to determine which of the P arrays are allocated storage. Arrays P1, P2, ... , PNPS are allocated storage, whereas the arrays P(NPS+1), P(NPS+2), ... , P10 are not. Thus this later set of P arrays must not be used.

4.3 JCL for Running SOFEPL

The job control language (JCL) constructs required to run SOFEPL on the CDC system at Wright-Patterson Air Force Base will

TABLE 5

DEFINITIONS OF PARAMETERS TRANSFERRED TO USTORE

PARAMETER	DEFINITIONS
IPLOT	THE CURRENT PLOT NUMBER
IRUN	THE CURRENT RUN NUMBER
IREC	THE CURRENT TAPE4 RECORD NUMBER
KREC	THE DIMENSION OF THE ARRAYS 'TIME' AND P1 THROUGH P-NPS. KREC IS THE NUMBER OF RECORDS (DATA POINTS) IN EACH SOFE RUN.
NCRV	THE NUMBER OF CURVES TO BE DRAWN FROM DATA STORED IN THE P ARRAYS. THE ARRAYS P1 THROUGH P-NCRV WILL BE PLOTTED AS CURVES ONE THROUGH NCRV. $0<NCRV<6$.
NPS	THE NUMBER OF P ARRAYS ALLOCATED FOR USER PLOTTING AND SCRATCH STORAGE OF DATA. NPS MUST BE GREATER THAN OR EQUAL TO NCRV AND LESS THAN OR EQUAL TO 10. NOTE THAT ALTHOUGH THE ARRAYS P1 THROUGH P10 ARE PASSED TO THIS ROUTINE AS FORMAL PARAMETERS, THERE IS NO STORAGE ALLOCATED TO THE ARRAYS P-(NPS+1) THROUGH P10.
NS	THE NUMBER OF ELEMENTS IN THE TRUTH STATE VECTOR
NF	THE NUMBER OF ELEMENTS IN THE FILTER STATE VECTOR, ALSO THE NUMBER OF ELEMENTS IN VF.
XS	TRUTH STATE VECTOR
XF	FILTER STATE VECTOR
VF	STATE VARIANCES FROM DIAGONAL OF THE COVARIANCE MATRIX PF
TIME	TIME ARRAY, DIMENSIONED KREC. ALL VALUES UP TO AND INCLUDING TIME(IREC) HAVE BEEN DEFINED WHEN USTORE IS CALLED.
P1-P10	LINEAR ARRAYS USED TO STORE DATA FOR PLOTTING AND SCRATCH PURPOSES. CURVES WILL BE DRAWN FROM VALUES IN P1 THROUGH P-NCRV. ARRAYS P1 THROUGH P-NPS HAVE BEEN ALLOCATED KREC WORDS OF STORAGE. ARRAYS P-(NPS+1) THROUGH P10 HAVE NO STORAGE ALLOCATED TO THEM AND THUS SHOULD NOT BE USED. THE P ARRAYS FOR WHICH STORAGE IS ALLOCATED ARE INITIALIZED TO ZERO BY SOFEPL.

TABLE 6

DEFINITIONS OF PARAMETERS TRANSFERRED TO USETPL

PARAMETER	DEFINITIONS
IPLOT	THE CURRENT PLOT NUMBER
RUNCNT	THE NUMBER OF RUNS PROCESSED TO CREATE THIS PLOT, AS SPECIFIED ON THE 'RUN SELECTION' CARD. RUNCNT IS USEFUL FOR COMPUTING AVERAGES AND STANDARD DEVIATIONS.
KREC	THE DIMENSION OF THE ARRAYS 'TIME' AND P1 THROUGH P-NPS. KREC IS THE NUMBER OF RECORDS (DATA POINTS) IN EACH SOFE RUN.
NCRV	THE NUMBER OF CURVES TO BE DRAWN FROM DATA STORED IN THE P ARRAYS. THE ARRAYS P1 THROUGH P-NCRV WILL BE PLOTTED AS CURVES ONE THROUGH NCRV. $0<NCRV<6$.
NPS	THE NUMBER OF P ARRAYS ALLOCATED FOR USER PLOTTING AND SCRATCH STORAGE OF DATA. NPS MUST BE GREATER THAN OR EQUAL TO NCRV AND LESS THAN OR EQUAL TO 10. NOTE THAT ALTHOUGH THE ARRAYS P1 THROUGH P10 ARE PASSED TO THIS ROUTINE AS FORMAL PARAMETERS, THERE IS NO STORAGE ALLOCATED TO THE ARRAYS P-(NPS+1) THROUGH P10.
TIME	TIME ARRAY, DIMENSIONED KREC.
P1-P10	LINEAR ARRAYS USED TO STORE DATA FOR PLOTTING AND SCRATCH PURPOSES. CURVES WILL BE DRAWN FROM VALUES IN P1 THROUGH P-NCRV. ARRAYS P1 THROUGH P-NPS HAVE BEEN ALLOCATED KREC WORDS OF STORAGE. ARRAYS P-(NPS+1) THROUGH P10 HAVE NO STORAGE ALLOCATED TO THEM AND THUS SHOULD NOT BE USED. THE P ARRAYS FOR WHICH STORAGE IS ALLOCATED ARE INITIALIZED TO ZERO BY SOFEPL.

be discussed briefly in Section 5 for an example problem. The Section 5 example illustrates the essentials of acquiring the appropriate files, operating on them to create an executable load module, and then running the program to produce a PLFILE. The example JCL is easily modified to accomodate a user's special needs, such as substituting magnetic tape for disk input of TAPE4. The procedures for such modifications are discussed in the appropriate CDC and/or ASD Computer Center manuals (eg. [4] and [7]) and are not covered here except to note the following. Communications into and out of SOFEPL are accomplished via the data files listed on the PROGRAM card, the first card in the SOFEPL source deck. This card is a special CDC construction that reads as follows:

```
PROGRAM SOFEPL (TAPE5 = 64/80, TAPE4, OUTPUT, TAPE6 = OUTPUT,
PLFILE = 0)
```

TAPE5 is the plot specification file, TAPE4 the SOFE/SOFEPL interface file, and PLFILE the device independent plot file. OUTPUT and TAPE6 are included on the PROGRAM card to accomodate printed output requirements.

SECTION 5
EXAMPLE RUN

This section demonstrates SOFEPL's capabilities with an example. Seven plots are created for the SOFE satellite orbit example discussed in [1]. Understanding the example presented here does not require a detailed knowledge of the SOFE example, but it will be helpful to recall these facts. The orbit problem is expressed in polar coordinates with truth and filter state vectors of dimension four. The state vector quantities are, in order, range, range rate, angle, and angle rate. These facts are reflected in plot titles and in SOFEPL control parameter choices. SOFEPL's printed output and all seven plots appear in Appendix A.

5.1 Example Input

Figure 9 shows the JCL commands, the Update directives, and the FORTRAN code required to run SOFEPL for the example problem. The first line of JCL in Figure 9 is the job card that adheres to a format established in [4]. The next three lines are comments that are echoed in the dayfile. In the fifth line, the SOFEPL source code is ATTACHed (brought to the job from disk storage as a local file named OLDPL). The SOFEPL source code (on the file OLDPL) is maintained in a compressed format using CDC's Update utility [5]. The sixth command is an Update control statement that expands OLDPL according to the Update directives placed after the *EOR in Figure 9. This expansion creates a file named COMPILE that is amenable to FORTRAN compilation. The COMPILE file so created contains all the subroutines of SOFEPL, with USTORE and USETPL modified by the FORTRAN code following the two *INSERT... directives. These modifications are needed for the desired type-18 plot.

```
SHM,T35,I060,CM115000.   problem no.,name,phone,etc.
COMMENT.
COMMENT.   ** SAVED IN SOFEPLJCL,ID=SHM,CY=999,SN=AFAL **
COMMENT.
ATTACH,OLDPL,SOFEPL,ID=SHM,MR=1,CY=999,SN=AFAL.
UPDATE,F,C=COMPILE.
FTN,L=0,I=COMPILE.
RETURN,OLDPL,COMPILE.
ATTACH,DISSPLA,DISSPLA,ID=LIBRARY,SN=ASD,CY=999.
LIBRARY,DISSPLA.
ATTACH,TAPE4,SOFEORBITPLOTTAPE,ID=SHM,CY=1,SN=AFAL.
ATTACH,DATA,SOFEPLDATA,ID=SHM,MR=1,CY=999,SN=AFAL.
REQUEST,PLFILE,*PF.
LGO(DATA)
CATALOG,PLFILE,PLFILE,ID=SHM.
*EOR       (END OF RECORD MARK)
*ID EXAMPLE
*INSERT USTORE.62
C
C  CONVERT RANGE AND ANGLE COORDINATES
C  TO CARTESIAN COORDINATES
C
      P1(IREC) = P1(IREC) + XS(1) * COS(XS(3))
      P2(IREC) = P2(IREC) + XS(1) * SIN(XS(3))
      P3(IREC) = P3(IREC) + XF(1) * COS(XF(3))
      P4(IREC) = P4(IREC) + XF(1) * SIN(XF(3))
*INSERT USETPL.47
C
C  COMPUTE AVERAGE AT EACH POINT OVER ALL RUNS
C
      DO 10 I = 1,KREC
         P1(I) = P1(I) / RUNCNT
         P2(I) = P2(I) / RUNCNT
         P3(I) = P3(I) / RUNCNT
         P4(I) = P4(I) / RUNCNT
10       CONTINUE
```

JCL

UPDATE Directives and Code for Plot Type 18

Figure 9. JCL, UPDATE Directives, and Fortran Code for the Example Run.

In line seven of the JCL in Figure 9, the FORTRAN compiler [6] produces an object file named LGO from the source code in COMPILE. The L=O setting suppresses listing of the code. In the eighth JCL line, OLDPL and COMPILE are RETURNed in order to free disk space and reduce system overhead. In lines nine and ten, the library of DISSPLA routines is ATTACHed and formally declared a LIBRARY. In lines 11 and 12, TAPE4 and the plot specification file (called DATA) are ATTACHed from disk locations where they have previously been CATALOGed. Line 13 requests space on disk for the 'permanent' storage of PLFILE. Line 14, LGO(DATA), causes the computer to load and execute SOFEPL using the files DATA and TAPE4 for input and creating PLFILE as output. Line 15 CATALOGs PLFILE for subsequent input to a DISSPLA postprocessor.

It is beyond the scope of this document to explain further the JCL and Update commands used in Figure 9. Users needing more information are referred to the references cited above and to Reference 7 which covers the CDC operating system.

As noted above, the SOFEPL plot specification file is the local file named DATA. Figure 10 shows this file, consisting of a title and seven plot sets. Blank lines are inserted between plot sets to enhance readability. Such lines are not required, but they do no harm when used as shown. Discussion of the plots themselves is in Subsection 5.3.

5.2 Example Printed Output

Appendix A contains 11 pages of printed output from the example run. The first two pages are system responses to the control statement UPDATE, F, C=COMPILE. These two pages show the code inserted in USTORE and USETPL, and summarize the deck structure of the OLDPL file. The third page is the introduction page that marks the beginning of SOFEPL execution. This page gives the title, the run date and time, and information about the SOFE run taken from the TAPE4 header.

Pages four through six of the printed output, starting with the header SOFEPL CARD DATA INPUT AND STORAGE ALLOCATION, echo

```
SOFEPL EXAMPLE BASED ON SATELLITE ORBIT PROBLEM

3,2,2/
1-L
TRUTH STATE 2 AND FILTER STATE 2 - PLOT TYPE 3$
TIME (TIME UNITS)$
RANGE RATE (LENGTH PER UNIT TIME)$

6,2,2/
1-L
RANGE RATE ERROR, AVG. AND ACTUAL STD. DEV., PLOT TYPE 6$
TIME (TIME UNITS)$
RANGE RATE ERROR (LENGTH / UNIT TIME)$

12,1,1,1/
1-10,20-30,40-50
STD. DEV. OF ACTUAL AND PREDICTED EST. ERROR - TYPE 12 PLOT
TIME (TIME UNITS)$
RANGE STANDARD DEVIATION (LENGTH)$

14,4,4,4,,,,,57.3/
1-L
COMPOSITE ERROR PLOT, SAMPLE OF PLOT TYPE 14$
TIME (TIME UNITS)$
ANGLE RATE ERROR (DEGREES PER UNIT TIME)$

16,,,,1,1,1/
1-L
RANGE RESIDUAL, AVG. AND PREDICTED STD. DEV., PLOT TYPE 16$
TIME (TIME UNITS)
RESIDUAL (LENGTH UNITS)

18,,,,,,,,,,,,,4,4/
41-50
TRAJECTORY IN CARTESIAN COORDINATES - PLOT TYPE 18$
TIME (TIME UNITS)$
X AND Y POSITION, TRUE AND ESTIMATED (LENGTH UNITS)$

3,2,2,,,,,10.,3.,0.0,25.,-.25,+.25/
1-L
SAME AS PLOT 1 BUT WITH AXIS SCALING AND TIME WINDOWING$
TIME (TIME UNITS * 10.)$
RANGE RATE (LENGTH PER UNIT TIME * 3.)$
```

Figure 10. Plot Specification File for the Example Run.

the input data for each plot and show the starting location and memory required to produce each plot. Error messages may appear in this section. SOFEPL attempts to recover gracefully from errors, usually by skipping plots in which input errors are detected.

Pages seven through ten, starting with the header PLOTTING EXECUTION, are output after the data for each plot is generated. These pages contain a summary of important data for each plot, some of it a repeat of previously echoed input data placed here for conveninece, and some of it new data available only when all curves of a plot are generated. In this latter category are the actual number of points in each curve, min/max values on each axis, and the average of the y-data for each curve. This section of output also contains a listing of the t, y coordinates for each curve if IPNTSW=1 (see plot number 5). The last page of printed output is the job dayfile in which the JCL commands of Figure 9 appear along with the system's responses.

5.3 Example Plots

The seven plots produced by the example run are shown as the last seven pages of Appendix A. The first five plots demonstrate a representative cross-section of available SOFEPL plots, while the last two focus more on SOFEPL's special capabilities. Between the instructions and explanations in Sections 2, 4, and 5, the linkage from Figure 10 through the printed output to each plot should be clear. Note the following about the first group of five plots: in Plot 3, a selected set of 32 of the available 50 runs are processed; in Plot 4, angles are converted from radians to degrees using 57.3 for SFY; in Plot 5, both axis lables are oft-center because terminating $ signs were not used on input.

Plot 6 illustrates plot type 18, the user-defined case. In this example, the orbiting satellite's r-Θ coordinates are expressed in x-y Cartesian form. To accomplish this, the code shown in Figure 9 is inserted in USTORE and USETPL. This code forms the x- and y-positions, both estimated and true, and averages them over the last 10 of the available 50 runs. Note that each curve in Plot 6 has a different line pattern to help distinguish it from its neighbors.

Plot 7, the last plot, demonstrates the use of axis limits and scale factors. In this plot these parameters have been chosen so as to "zoom in" on the data. Plot 7 should be compared with Plot 1 to see the zooming, or windowing, effect. Both plots are examples of plot type three and both use the data from the same runs. The scale factors, axis limits, and titles are the only differences. Note that the axis limits are in the units of the data after the scale factor was applied.

Different line patterns are used in plots with multiple curves to distinguish which curve corresponds to what quantity. In plot type 18 each curve has a unique line pattern. In plot types 1 through 17, these rules apply. When only one curve is drawn, its line is solid. When two curves are drawn, the line pattern is solid for the truth quantity and broken for the filter quantity. When three curves are drawn, two of them are always symmetric, either about y=o or about the third curve; in this case the symmetric pair is broken and the solitary curve is solid. There are no four-curve plots. In the single five-curve plot, three line patterns are used: $\bar{E}_{ij}$ is solid; $\bar{E}_{ij} \pm SE_{ij}$ are broken of one pattern; and $\pm\sqrt{\overline{Vf}_j}$ are broken of another pattern. In this last case differences in symmetry help to distinguish which curves are which.

APPENDIX A

EXAMPLE RUN OUTPUT

```
/////    *IDENT EXAMPLE
*****    *INSERT USTORE.62
         C
         C  CONVERT RANGE AND ANGLE COORDINATES
         C  TO CARTESIAN COORDINATES
         C
               P1(IREC) = P1(IREC) + XS(1) * COS(XS(3))
               P2(IREC) = P2(IREC) + XS(1) * SIN(XS(3))
               P3(IREC) = P3(IREC) + XF(1) * COS(XF(3))
               P4(IREC) = P4(IREC) + XF(1) * SIN(XF(3))
*****    *INSERT USETPL.47
         C
         C  COMPUTE AVERAGE AT EACH POINT OVER ALL RUNS
         C
               DO 10 I = 1,KREC
                  P1(I) = P1(I) / RUNCNT
                  P2(I) = P2(I) / RUNCNT
                  P3(I) = P3(I) / RUNCNT
                  P4(I) = P4(I) / RUNCNT
         10       CONTINUE


         MODIFICATIONS / CONTROL CARDS


USETPL   C                                                        EXAMPLE     1    I
USETPL   C  COMPUTE AVERAGE AT EACH POINT OVER ALL RUNS           EXAMPLE     2    I
USETPL   C                                                        EXAMPLE     3    I
USETPL         DO 10 I = 1,KREC                                   EXAMPLE     4    I
USETPL            P1(I) = P1(I) / RUNCNT                          EXAMPLE     5    I
USETPL            P2(I) = P2(I) / RUNCNT                          EXAMPLE     6    I
USETPL            P3(I) = P3(I) / RUNCNT                          EXAMPLE     7    I
USETPL            P4(I) = P4(I) / RUNCNT                          EXAMPLE     8    I
USETPL   10       CONTINUE                                        EXAMPLE     9    I

USTORE   C                                                        EXAMPLE    10    I
USTORE   C  CONVERT RANGE AND ANGLE COORDINATES                   EXAMPLE    11    I
USTORE   C  TO CARTESIAN COORDINATES                              EXAMPLE    12    I
USTORE   C                                                        EXAMPLE    13    I
USTORE         P1(IREC) = P1(IREC) + XS(1) * COS(XS(3))           EXAMPLE    14    I
USTORE         P2(IREC) = P2(IREC) + XS(1) * SIN(XS(3))           EXAMPLE    15    I
USTORE         P3(IREC) = P3(IREC) + XF(1) * COS(XF(3))           EXAMPLE    16    I
USTORE         P4(IREC) = P4(IREC) + XF(1) * SIN(XF(3))           EXAMPLE    17    I
```

CORRECTION IDENTS ARE LISTED IN CHRONOLOGICAL ORDER OF INSERTION

SOFEPL	CARDIN	CARD12	CARD35	CHKDTA	INITAL	MAXMIN	PLOT
PLTNED	PNTDTA	P15T17	SCAN	SETPLT	SKPRUN	SPLITA	STORE
STRING	STRGET	STRNTP	USETPL	USTORE	EXAMPLE		

DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE

YANK$$$	SOFEPL	CARDIN	CARD12	CARD35	CHKDTA	INITAL	MAXMIN
PLOT	PLTNED	PNTDTA	P15T17	SCAN	SETPLT	SKPRUN	SPLITA
STORE	STRING	STRGET	STRNTP	USETPL	USTORE		

DECKS WRITTEN TO COMPILE FILE

SOFEPL	CARDIN	CARD12	CARD35	CHKDTA	INITAL	MAXMIN	PLOT
PLTNED	PNTDTA	P15T17	SCAN	SETPLT	SKPRUN	SPLITA	STORE
STRING	STRGET	STRNTP	USETPL	USTORE			

THIS UPDATE REQUIRED 37100B WORDS OF CORE.

```
*****************************************************************************************************

                                        S O F E P L

                   AN ENSEMBLE AVERAGING AND PLOTTING POST-PROCESSOR

                                     FOR USE WITH THE

                      SIMULATION FOR OPTIMAL FILTER EVALUTION (SOFE)



SOFEPL PROBLEM TITLE:   SOFEPL EXAMPLE BASED ON SATELLITE ORBIT PROBLEM

DATE AND TIME OF THIS SOFEPL RUN:    08/27/81  AT  11.20.53.



******************************  SOFE RUN INFORMATION (FROM SOFE TAPE4)  ******************************

SOFE PROBLEM TITLE:   SATELLITE ORBIT DETERMINATION USING AN EXTENDED KALMAN FILTER

DATE AND TIME OF SOFE RUN:    08/06/81  AT  14.12.02.

SOFE INPUT CONTROL PARAMETERS:

     NO. OF TRUTH STATES       NO. OF FILTER STATES      NO. OF MEASUREMENT UPDATES      NO. OF RUNS
     NS =  4                   NF =  4                   M =  2                          IPASS =  50

     RUN BEGIN TIME     RUN END TIME     UPDATE BEGIN TIME     UPDATE TIME INTERVAL     PLOT OUTPUT INTERVAL
     TO =  0.           TF =   5.0000    TMEASO =  0.          DTMEAS =   .50000        DTCCPL =  5.00000E-02
```

```
************************  SOFEPL CARD DATA INPUT AND STORAGE ALLOCATION  ************************************

PLOT   1 INPUT INFORMATION:

     PLOT NO.      PLOT TYPE      INDICES OF STATES/VARIANCES/RESIDUALS TO BE PLOTTED          PRINT SWITCH
     IPLOT =  1    IPLTYP =  3    NXS =  2   NXF =  2   NVF =  1   NZRES =  1   NALPHA =  1    IPNTSW = 0

     SCALE FACTORS                        TIME BOUNDS                        Y-AXIS BOUNDS
     SFT =  1.000      SFY =  1.000       TB = 0.           TE = 0.          YLO = 0.          YHI = 0.

     RUNS TO BE INCLUDED                         NSETS =  1
      1 - 50,

     PLOT TITLE    :  TRUTH STATE 2 AND FILTER STATE 2 - PLOT TYPE 3$
     T-AXIS LABEL :  TIME (TIME UNITS)$
     Y-AXIS LABEL :  RANGE RATE (LENGTH PER UNIT TIME)$

     BASE STORAGE LOCATION FOR PLOT  1 DATA  :  440
     NO. OF WORDS RESERVED FOR PLOT  1 DATA  :  330

PLOT   2 INPUT INFORMATION:

     PLOT NO.      PLOT TYPE      INDICES OF STATES/VARIANCES/RESIDUALS TO BE PLOTTED          PRINT SWITCH
     IPLOT =  2    IPLTYP =  6    NXS =  2   NXF =  2   NVF =  1   NZRES -  1   NALPHA =  1    IPNTSW = 0

     SCALE FACTORS                        TIME BOUNDS                        Y-AXIS BOUNDS
     SFT =  1.000      SFY =  1.000       TB = 0.           TE = 0.          YLO = 0.          YHI = 0.

     RUNS TO BE INCLUDED                         NSETS =  1
      1 - 50,

     PLOT TITLE    :  RANGE RATE ERROR, AVG. AND ACTUAL STD. DEV., PLOT TYPE 6$
     T-AXIS LABEL :  TIME (TIME UNITS)$
     Y-AXIS LABEL :  RANGE RATE ERROR (LENGTH / UNIT TIME)$

     BASE STORAGE LOCATION FOR PLOT  2 DATA  :  770
     NO. OF WORDS RESERVED FOR PLOT  2 DATA  :  465

PLOT   3 INPUT INFORMATION:

     PLOT NO.      PLOT TYPE      INDICES OF STATES/VARIANCES/RESIDUALS TO BE PLOTTED          PRINT SWITCH
     IPLOT =  3    IPLTYP = 12    NXS =  1   NXF =  1   NVF =  1   NZRES =  1   NALPHA =  1    IPNTSW = 0

     SCALE FACTORS                        TIME BOUNDS                        Y-AXIS BOUNDS
     SFT =  1.000      SFY =  1.000       TB = 0.           TE = 0.          YLO = 0.          YHI = 0.

     RUNS TO BE INCLUDED                         NSETS =  3
      1 - 10,  20 - 30,  40 - 50,

     PLOT TITLE    :  STD. DEV. OF ACTUAL AND PREDICTED EST. ERROR - TYPE 12 PLOT
     T-AXIS LABEL :  TIME (TIME UNITS)$
     Y-AXIS LABEL :  RANGE STANDARD DEVIATION (LENGTH)$

     BASE STORAGE LOCATION FOR PLOT  3 DATA  : 1235
     NO. OF WORDS RESERVED FOR PLOT  3 DATA  :  469
```

```
PLOT   4 INPUT INFORMATION:

     PLOT NO.       PLOT TYPE       INDICES OF STATES/VARIANCES/RESIDUALS TO BE PLOTTED              PRINT SWITCH
     IPLOT =  4     IPLTYP = 14     NXS =  4    NXF =  4    NVF =  4    NZRES =  1    NALPHA =  1    IPNTSW = 0

     SCALE FACTORS                           TIME BOUNDS                             Y-AXIS BOUNDS
     SFT =  1.000       SFY =  57.30        TB = O.            TE = O.              YLO = O.          YHI = O.

     RUNS TO BE INCLUDED                                NSETS =  1
      1 - 50.

     PLOT TITLE    :  COMPOSITE ERROR PLOT, SAMPLE OF PLOT TYPE 14$
     T-AXIS LABEL :  TIME (TIME UNITS)$
     Y-AXIS LABEL :  ANGLE RATE ERROR (DEGREES PER UNIT TIME)$

     BASE STORAGE LOCATION FOR PLOT  4 DATA  : 1704
     NO. OF WORDS RESERVED FOR PLOT  4 DATA  :  735

PLOT   5 INPUT INFORMATION:

     PLOT NO.       PLOT TYPE       INDICES OF STATES/VARIANCES/RESIDUALS TO BE PLOTTED              PRINT SWITCH
     IPLOT =  5     IPLTYP = 16     NXS =  1    NXF =  1    NVF =  1    NZRES =  1    NALPHA =  1    IPNTSW = 1

     SCALE FACTORS                           TIME BOUNDS                             Y-AXIS BOUNDS
     SFT =  1.000       SFY =  1.000        TB = O.            TE = O.              YLO = O.          YHI = O.

     RUNS TO BE INCLUDED                                NSETS =  1
      1 - 50.

     PLOT TITLE    :  RANGE RESIDUAL, AVG. AND PREDICTED STD. DEV., PLOT TYPE 16$
     T-AXIS LABEL :  TIME (TIME UNITS)
     Y-AXIS LABEL :  RESIDUAL (LENGTH UNITS)

     BASE STORAGE LOCATION FOR PLOT  5 DATA  : 2439
     NO. OF WORDS RESERVED FOR PLOT  5 DATA  :  104

PLOT   6 INPUT INFORMATION:

     PLOT NO.       PLOT TYPE           NO. OF CURVES TO BE DRAWN      NO. OF P ARRAYS USED       PRINT SWITCH
     IPLOT =  6     IPLTYP = 18         NCURV = 4                      NPS =  4                   IPNTSW = 0

     SCALE FACTORS                           TIME BOUNDS                             Y-AXIS BOUNDS
     SFT =  1.000       SFY =  1.000        TB = O.            TE = O.              YLO = O.          YHI = O.

     RUNS TO BE INCLUDED                                NSETS =  1
     41 - 50.

     PLOT TITLE    :  TRAJECTORY IN CARTESIAN COORDINATES - PLOT TYPE 18$
     T-AXIS LABEL :  TIME (TIME UNITS)$
     Y-AXIS LABEL :  X AND Y POSITION, TRUE AND ESTIMATED (LENGTH UNITS)$

     BASE STORAGE LOCATION FOR PLOT  6 DATA  : 2543
     NO. OF WORDS RESERVED FOR PLOT  6 DATA  :  600
```

```
PLOT   7 INPUT INFORMATION:

     PLOT NO.      PLOT TYPE      INDICES OF STATES/VARIANCES/RESIDUALS TO BE PLOTTED          PRINT SWITCH
     IPLOT =  7    IPLTYP =  3    NXS =  2    NXF =  2    NVF =  1    NZRES =  1   NALPHA =  1    IPNTSW = 0

     SCALE FACTORS                          TIME BOUNDS                        Y-AXIS BOUNDS
     SFT =  10.00      SFY =  3.000         TB = 0.            TE =  25.000    YLO = -.25000      YHI =  .25000

     RUNS TO BE INCLUDED                        NSETS =  1
      1 - 50,

     PLOT TITLE    :  SAME AS PLOT 1 BUT WITH AXIS SCALING AND TIME WINDOWING$
     T-AXIS LABEL :  TIME (TIME UNITS * 10.)$
     Y-AXIS LABEL :  RANGE RATE (LENGTH PER UNIT TIME * 3.)$

     BASE STORAGE LOCATION FOR PLOT  7 DATA  : 3143
     NO. OF WORDS RESERVED FOR PLOT  7 DATA  :  330
```

```
*************************************  PLOTTING EXECUTION  ****************************************************

PLOT NUMBER              :   1
PLOT TITLE               :  TRUTH STATE 2 AND FILTER STATE 2 - PLOT TYPE 3$                  $

PLOT TYPE                :   3
NUMBER OF CURVES         :   2
CURVE  1                 :  AVG( XS(   2) )
CURVE  2                 :  AVG( XF(   2) )

TIME-AXIS SCALE FACTOR   :    1.000
Y-AXIS SCALE FACTOR      :    1.000
TIME-AXIS LABEL          :  TIME (TIME UNITS)$                                               $
Y-AXIS LABEL             :  RANGE RATE (LENGTH PER UNIT TIME)$                               $
DATA PRINT SWITCH        :  OFF
SOFE RUNS AVERAGED       :    1 - 50,

POINTS PER CURVE         :  132
MINIMUM TIME             :    0.
MAXIMUM TIME             :     5.0000
MINIMUM Y VALUE          :    -.177501
MAXIMUM Y VALUE          :    4.712040E-02
CURVE 1 AVG. OVER TIME   :    0.
CURVE 2 AVG. OVER TIME   :   -4.848671E-02


PLOT NUMBER              :   2
PLOT TITLE               :  RANGE RATE ERROR, AVG. AND ACTUAL STD. DEV., PLOT TYPE 6$     $

PLOT TYPE                :   6
NUMBER OF CURVES         :   3
CURVE  1                 :  AVG( E )                    WHERE E = XS(   2) - XF(   2)
CURVE  2                 :  + SE                        WHERE E = XS(   2) - XF(   2)
CURVE  3                 :  - SE                        WHERE E = XS(   2) - XF(   2)

TIME-AXIS SCALE FACTOR   :    1.000
Y-AXIS SCALE FACTOR      :    1.000
TIME-AXIS LABEL          :  TIME (TIME UNITS)$                                               $
Y-AXIS LABEL             :  RANGE RATE ERROR (LENGTH / UNIT TIME)$                           $
DATA PRINT SWITCH        :  OFF
SOFE RUNS AVERAGED       :    1 - 50,

POINTS PER CURVE         :  132
MINIMUM TIME             :    0.
MAXIMUM TIME             :     5.0000
MINIMUM Y VALUE          :    -.387272
MAXIMUM Y VALUE          :     .387272
CURVE 1 AVG. OVER TIME   :    4.848671E-02
CURVE 2 AVG. OVER TIME   :     .167978
CURVE 3 AVG. OVER TIME   :    -.167978
```

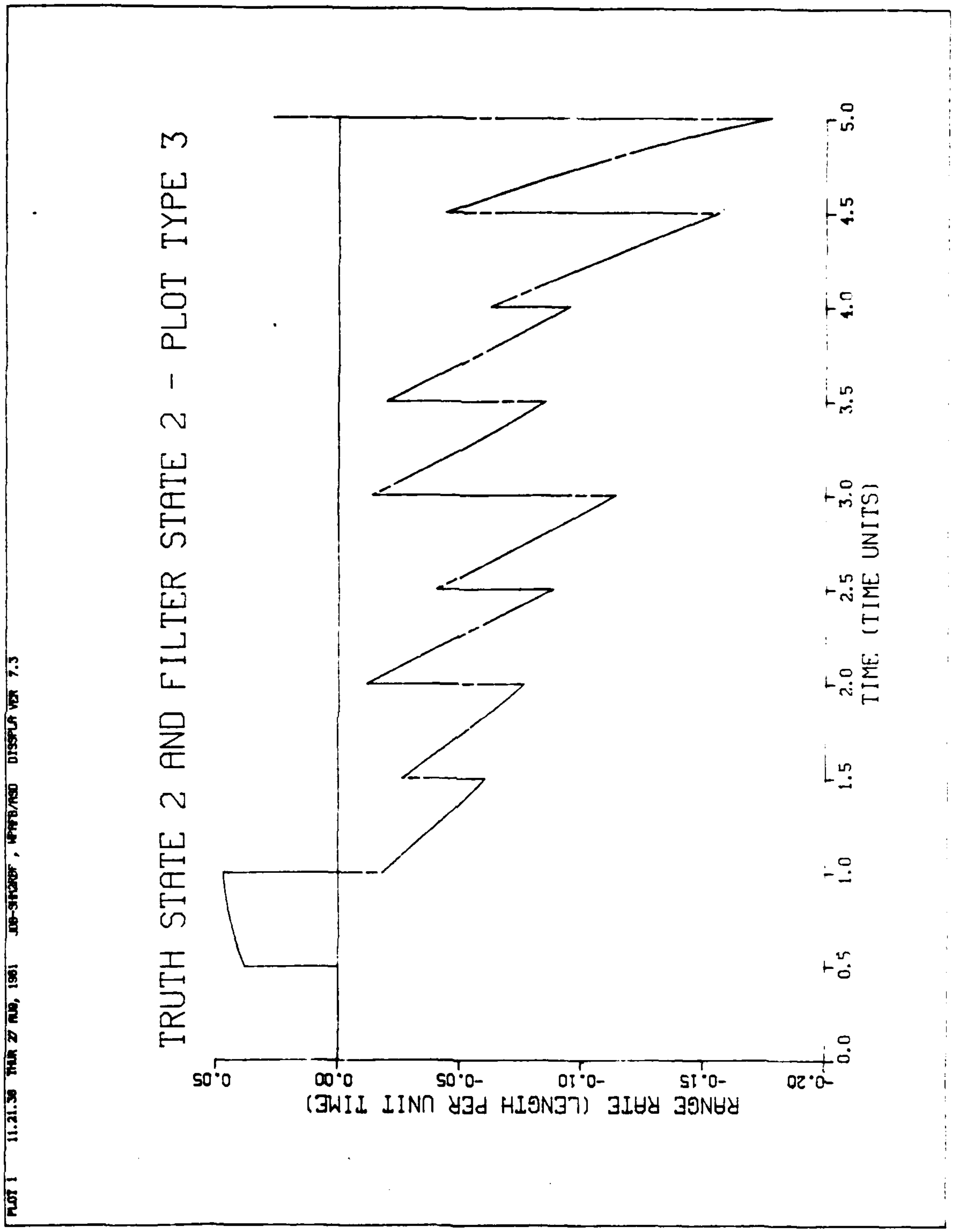
```
PLOT NUMBER              :   3
PLOT TITLE               :  STD. DEV. OF ACTUAL AND PREDICTED EST. ERROR - TYPE 12 PLOT $

PLOT TYPE                :  12
NUMBER OF CURVES         :   2
CURVE  1                 :  + SE                         WHERE E = XS(  1) - XF(  1)
CURVE  2                 :  +AVG( SQRT( VF(  1) ) )

TIME-AXIS SCALE FACTOR   :    1.000
Y-AXIS SCALE FACTOR      :    1.000
TIME-AXIS LABEL          :  TIME (TIME UNITS)$                                      $
Y-AXIS LABEL             :  RANGE STANDARD DEVIATION (LENGTH)$                      $
DATA PRINT SWITCH        :  OFF
SOFE RUNS AVERAGED       :    1 - 10,  20 - 30,  40 - 50,

POINTS PER CURVE         :  132
MINIMUM TIME             :    0.
MAXIMUM TIME             :     5.0000
MINIMUM Y VALUE          :    0.
MAXIMUM Y VALUE          :     .465651
CURVE 1 AVG. OVER TIME   :     .110218
CURVE 2 AVG. OVER TIME   :     .157521



PLOT NUMBER              :   4
PLOT TITLE               :  COMPOSITE ERROR PLOT, SAMPLE OF PLOT TYPE 14$           $

PLOT TYPE                :  14
NUMBER OF CURVES         :   5
CURVE  1                 :  AVG( E )                     WHERE E = XS(  4) - XF(  4)
CURVE  2                 :  AVG( E ) + SE                WHERE E = XS(  4) - XF(  4)
CURVE  3                 :  AVG( E ) - SE                WHERE E = XS(  4) - XF(  4)
CURVE  4                 :  +AVG( SQRT( VF(  4) ) )
CURVE  5                 :  -AVG( SQRT( VF(  4) ) )

TIME-AXIS SCALE FACTOR   :    1.000
Y-AXIS SCALE FACTOR      :    57.30
TIME-AXIS LABEL          :  TIME (TIME UNITS)$                                      $
Y-AXIS LABEL             :  ANGLE RATE ERROR (DEGREES PER UNIT TIME)$               $
DATA PRINT SWITCH        :  OFF
SOFE RUNS AVERAGED       :    1 - 50,

POINTS PER CURVE         :  132
MINIMUM TIME             :    0.
MAXIMUM TIME             :     5.0000
MINIMUM Y VALUE          :   -43.7536
MAXIMUM Y VALUE          :    73.4663
CURVE 1 AVG. OVER TIME   :    1.72196
CURVE 2 AVG. OVER TIME   :    14.6811
CURVE 3 AVG. OVER TIME   :   -11.2372
CURVE 4 AVG. OVER TIME   :    16.7488
CURVE 5 AVG. OVER TIME   :   -16.7488
```

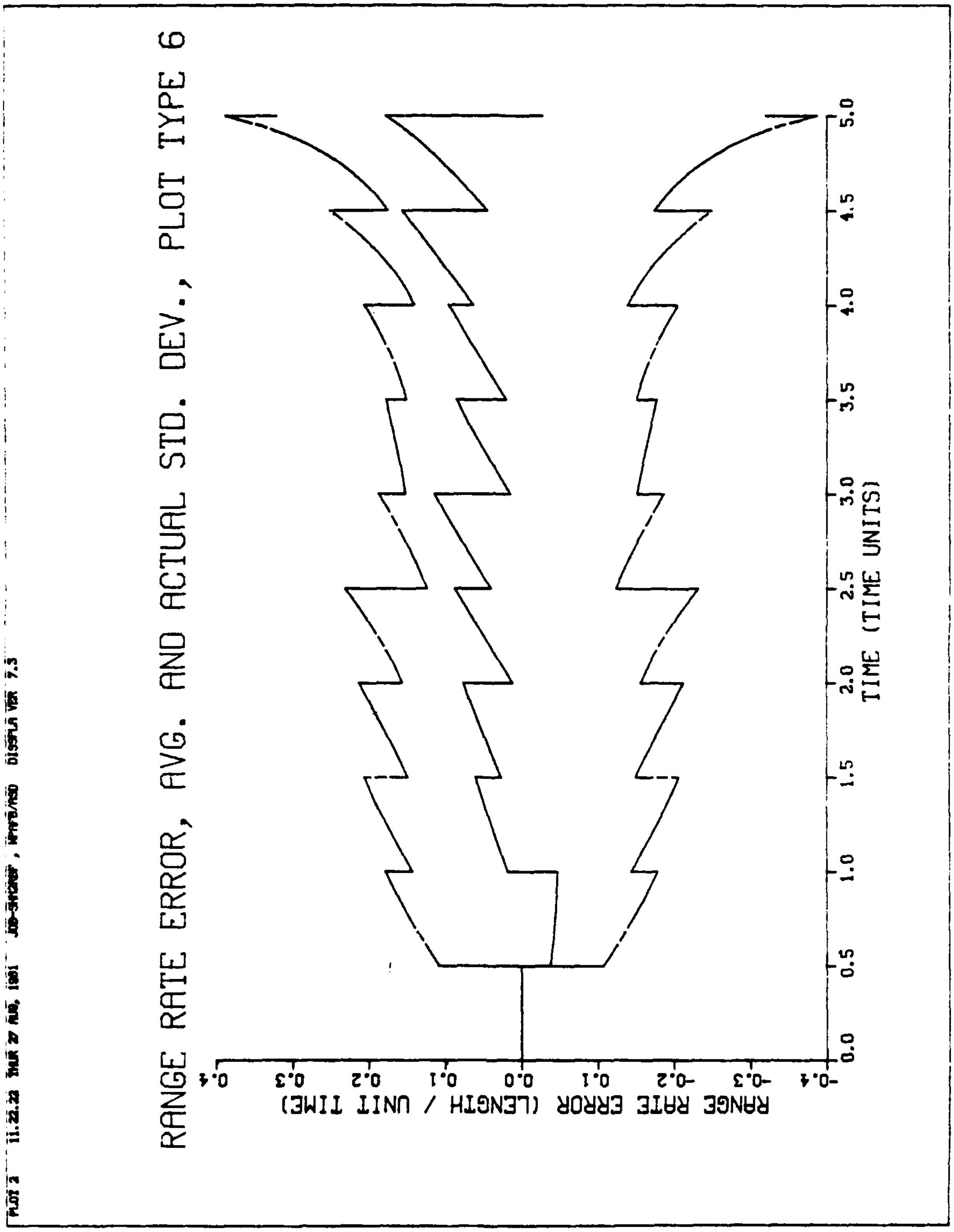
```
PLOT NUMBER              :  5
PLOT TITLE               : RANGE RESIDUAL, AVG. AND PREDICTED STD. DEV., PLOT TYPE 16$ $

PLOT TYPE                : 16
NUMBER OF CURVES         :  3
CURVE  1                 : AVG( ZRES(  1) )
CURVE  2                 : +AVG( SQRT( ALPHA(  1) ) )
CURVE  3                 : -AVG( SQRT( ALPHA(  1) ) )

TIME-AXIS SCALE FACTOR   :    1.000
Y-AXIS SCALE FACTOR      :    1.000
TIME-AXIS LABEL          : TIME (TIME UNITS)                                  $
Y-AXIS LABEL             : RESIDUAL (LENGTH UNITS)                            $
DATA PRINT SWITCH        : ON
SOFE RUNS AVERAGED       :    1 - 50,

POINTS PER CURVE         :  10
MINIMUM TIME             :     .50000
MAXIMUM TIME             :     5.0000
MINIMUM Y VALUE          :    -.476268
MAXIMUM Y VALUE          :     .476268
CURVE 1 AVG. OVER TIME   :    2.839377E-02
CURVE 2 AVG. OVER TIME   :     .244662
CURVE 3 AVG. OVER TIME   :    -.244662
```

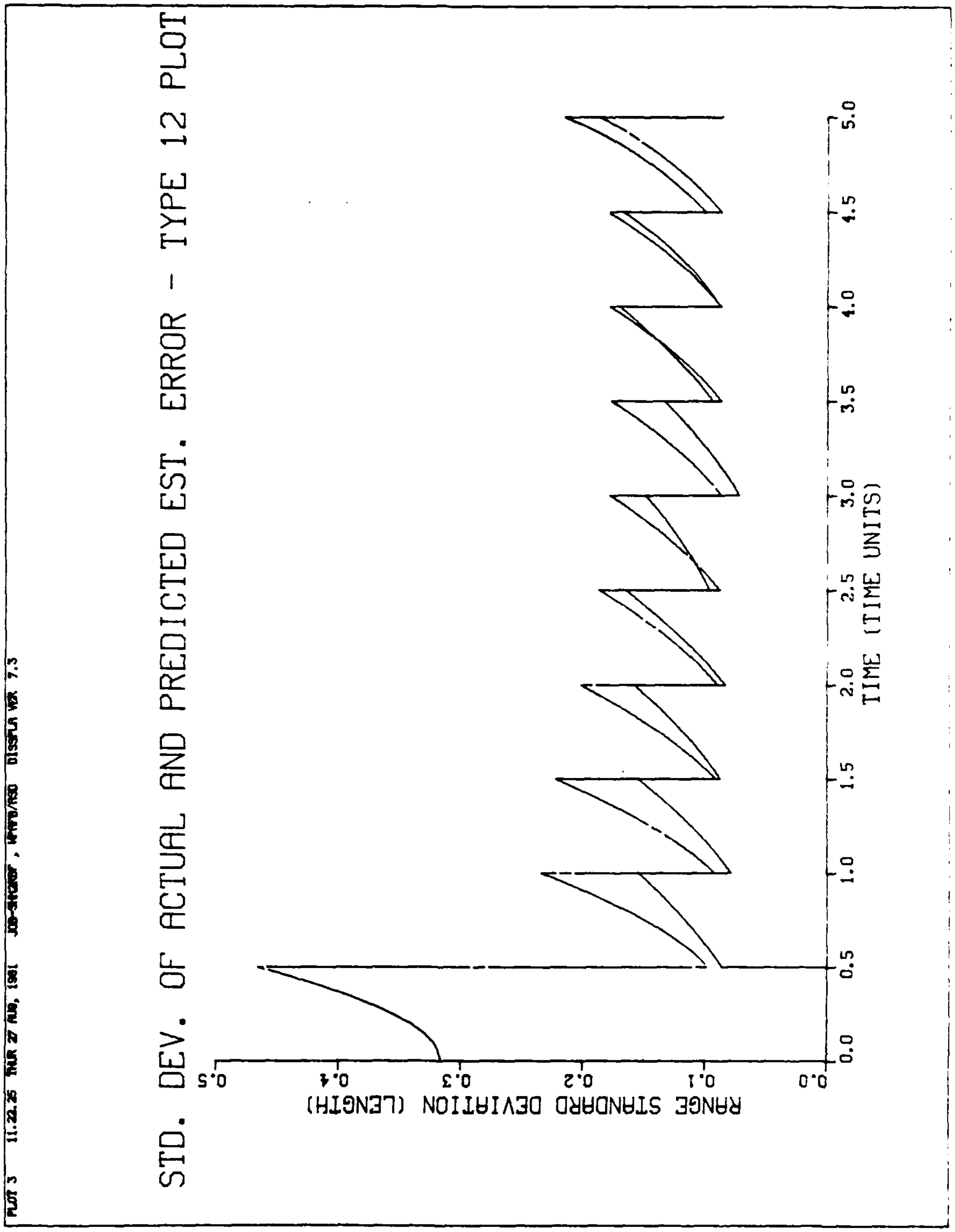
POINT	TIME	CURVE 1	CURVE 2	CURVE 3
1	.50000	3.114281E-02	.476269	-.476268
2	1.0000	-4.914512E-02	.254721	-.254721
3	1.5000	9.452269E-04	.244062	-.244062
4	2.0000	3.294121E-02	.225648	-.225648
5	2.5000	2.908494E-02	.212389	-.212389
6	3.0000	6.327496E-02	.206210	-.206210
7	3.5000	3.643897E-02	.204898	-.204898
8	4.0000	5.361472E-03	.206341	-.206341
9	4.5000	6.980489E-02	.207276	-.207276
10	5.0000	6.408839E-02	.208804	-.208804

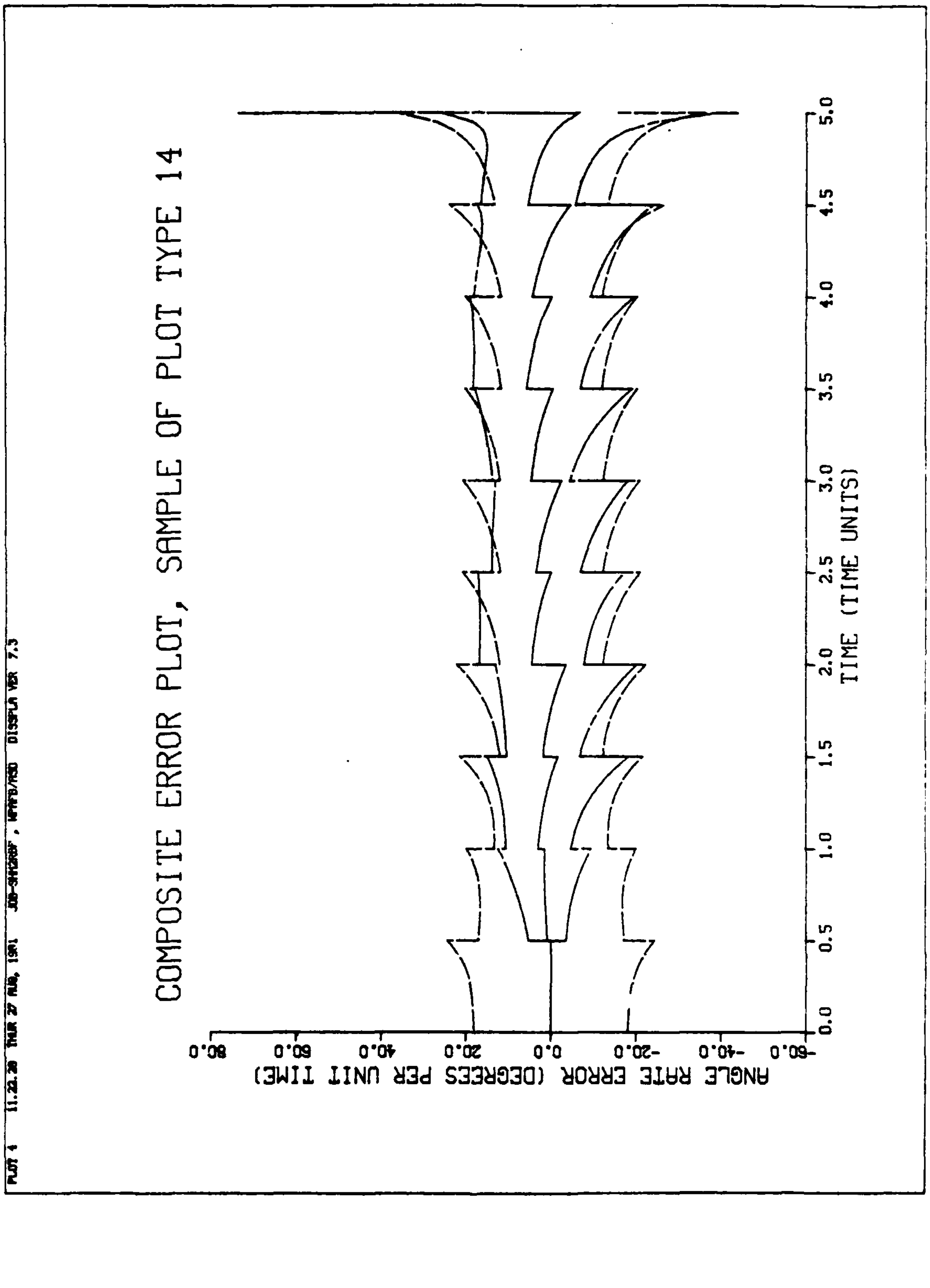
```
PLOT NUMBER             :   6
PLOT TITLE              :  TRAJECTORY IN CARTESIAN COORDINATES - PLOT TYPE 18$          $

PLOT TYPE               :  18
NUMBER OF CURVES        :   4
CURVE  1                :  USER DEFINED
CURVE  2                :  USER DEFINED
CURVE  3                :  USER DEFINED
CURVE  4                :  USER DEFINED

TIME-AXIS SCALE FACTOR  :    1.000
Y-AXIS SCALE FACTOR     :    1.000
TIME-AXIS LABEL         :  TIME (TIME UNITS)$                                            $
Y-AXIS LABEL            :  X AND Y POSITION, TRUE AND ESTIMATED (LENGTH UNITS)$          $
DATA PRINT SWITCH       :  OFF
SOFE RUNS AVERAGED      :   41 - 50,

POINTS PER CURVE        :  132
MINIMUM TIME            :    0.
MAXIMUM TIME            :     5.0000
MINIMUM Y VALUE         :    -.999965
MAXIMUM Y VALUE         :     1.01327
CURVE 1 AVG. OVER TIME  :    -.189066
CURVE 2 AVG. OVER TIME  :     .118599
CURVE 3 AVG. OVER TIME  :    -.158549
CURVE 4 AVG. OVER TIME  :     .140612


PLOT NUMBER             :   7
PLOT TITLE              :  SAME AS PLOT 1 BUT WITH AXIS SCALING AND TIME WINDOWING$     $

PLOT TYPE               :   3
NUMBER OF CURVES        :   2
CURVE  1                :  AVG( XS(  2) )
CURVE  2                :  AVG( XF(  2) )

TIME-AXIS SCALE FACTOR  :    10.00
Y-AXIS SCALE FACTOR     :    3.000
TIME-AXIS LABEL         :  TIME (TIME UNITS * 10.)$                                      $
Y-AXIS LABEL            :  RANGE RATE (LENGTH PER UNIT TIME * 3.)$                       $
DATA PRINT SWITCH       :  OFF
SOFE RUNS AVERAGED      :    1 - 50,

POINTS PER CURVE        :   66
MINIMUM TIME            :    0.
MAXIMUM TIME            :     25.000
MINIMUM Y VALUE         :    -.264176
MAXIMUM Y VALUE         :     .141361
CURVE 1 AVG. OVER TIME  :    0.
CURVE 2 AVG. OVER TIME  :   -6.139222E-02

END DISSPLA --     14144 VECTORS GENERATED IN      7 PLOT FRAMES.
```

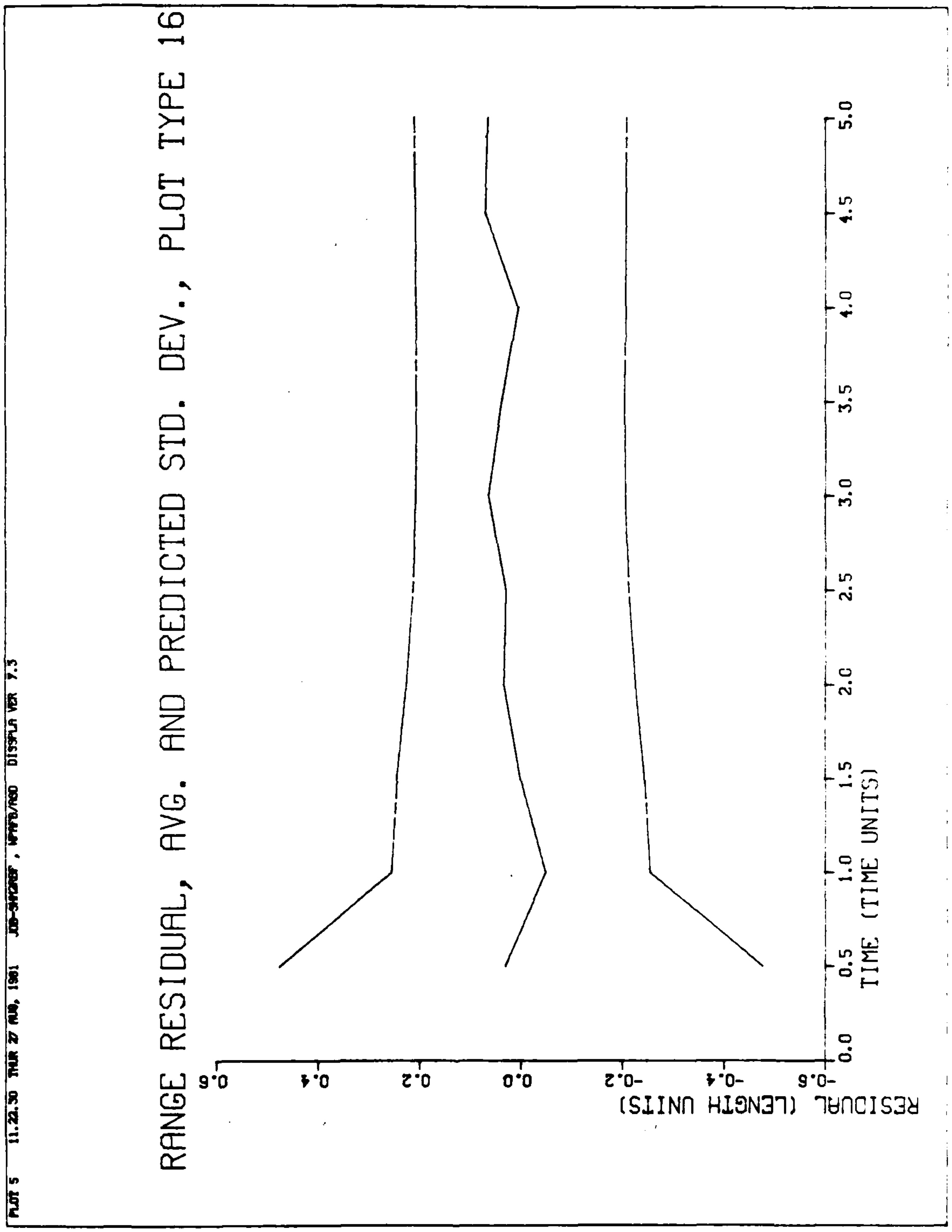
```
  CSB   NOS/BE L530C       L530C-CMR3 07/13/81
11.17.15.SHM2RBF  FROM CSA/2R
11.17.15.IP  00000256 WORDS - FILE INPUT  , DC 04
11.17.15.SHM,T35,I060,CM115000.
11.17.17.
11.17.17.  ** SAVED IN SOFEPLJCL,ID=SHM,CY=999,SN
11.17.17.=AFAL **
11.17.17.
11.17.17.ATTACH,OLDPL,SOFEPL,ID=SHM,MR=1,CY=999,S
11.17.17.N=AFAL.
11.17.26.UPDATE,F,C=COMPILE.
11.17.53. UPDATE COMPLETE.
11.17.53.FTN,L=0,I=COMPILE.
11.20.08.    10.352 CP SECONDS COMPILATION TIME
11.20.08.RETURN,OLDPL,COMPILE.
11.20.10.ATTACH,DISSPLA,DISSPLA,ID=LIBRARY,SN=ASD
11.20.10.,CY=999.
11.20.11.LIBRARY,DISSPLA.
11.20.11.ATTACH,TAPE4,SOFEORBITPLOTTAPE,ID=SHM,CY
11.20.11.=1,SN=AFAL.
11.20.12.ATTACH,DATA,SOFEPLDATA,ID=SHM,MR=1,CY=99
11.20.12.9,SN=AFAL.
11.20.24.REQUEST,PLFILE,*PF.
11.20.34.LGO(DATA)
11.20.50.     NON-FATAL LOADER ERRORS -
11.20.50.NON-EXISTENT LIBRARY GIVEN -  SYSIO
11.20.50.     NON-FATAL LOADER ERRORS -
11.20.50.NON-EXISTENT LIBRARY GIVEN -  SYSIO
11.23.14.    STOP  SOFEPL DONE
11.23.14.    113400 MAXIMUM EXECUTION FL.
11.23.14.    13.114 CP SECONDS EXECUTION TIME.
11.23.14.CATALOG,PLFILE,PLFILE,ID=SHM.
11.23.14.INITIAL CATALOG
11.23.14.RP = 008 DAYS
11.23.14.CT ID=      SHM PFN=PLFILE
11.23.14.CT CY= 001 SN=AFAL    00004544 WORDS.
11.23.14.OP  00002432 WORDS - FILE OUTPUT , DC 40
11.23.14.MS   11840  WORDS  (   47424 MAX USED)
11.23.14.CPA     27.239 SEC.       13.637 ADJ.
11.23.14.IO      52.201 SEC.       29.545 ADJ.
11.23.14.CM    1987.709 KWS.       16.139 ADJ.
11.23.14.CRUS                      59.323
11.23.14.COST               $       2.70
11.23.14.PP      77.335 SEC.     DATE 08/27/81
11.23.14.EJ  END OF JOB, 2R
```


TRUTH STATE 2 AND FILTER STATE 2 - PLOT TYPE 3
PLOT 1
11.21.38 THUR 27 AUG, 1981
DISSPLA VER 7.3
TIME (TIME UNITS)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
RANGE RATE (LENGTH PER UNIT TIME)
0.05
0.00
-0.05
-0.10
-0.15
-0.20

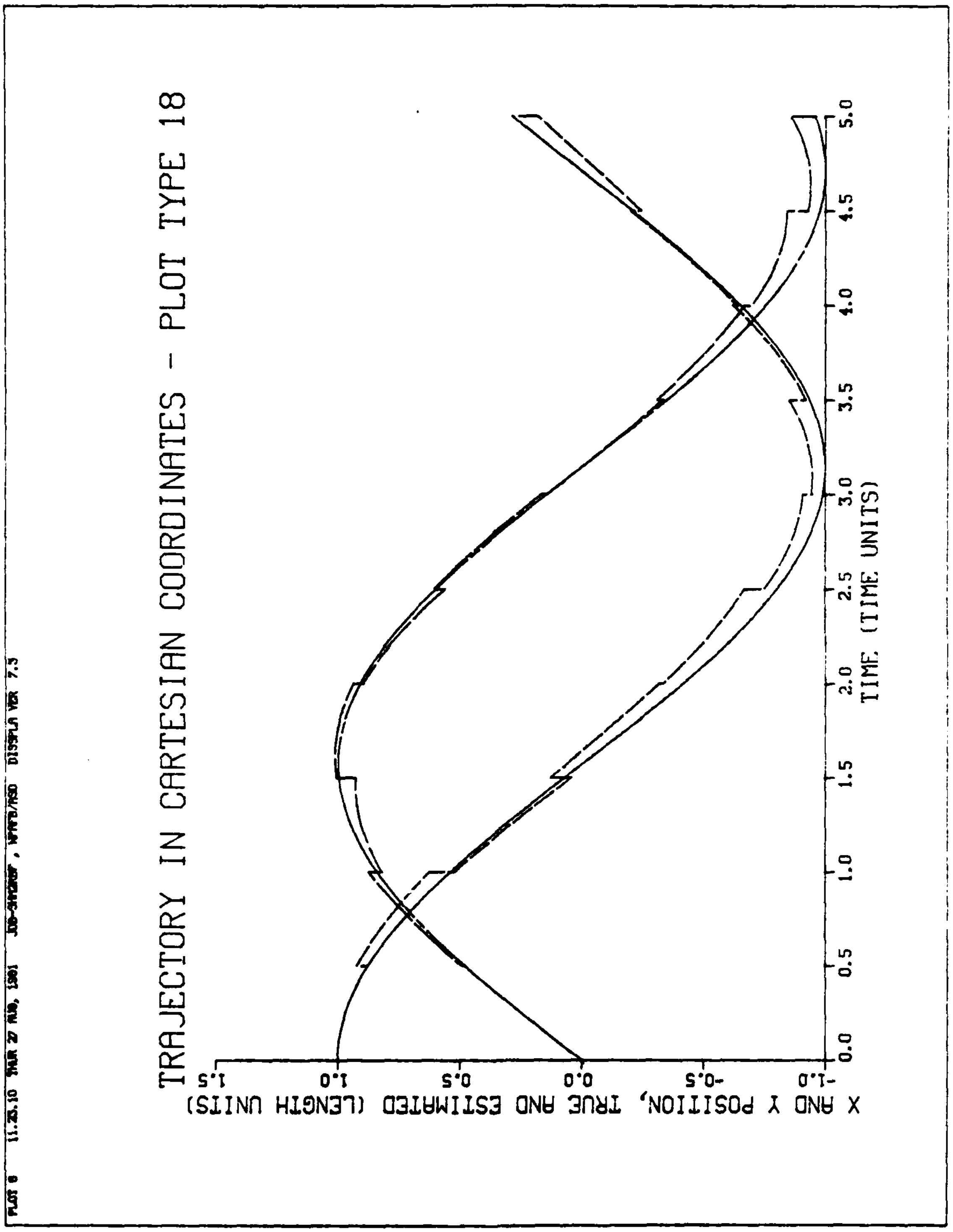

RANGE RATE ERROR, AVG. AND ACTUAL STD. DEV., PLOT TYPE 6
RANGE RATE ERROR (LENGTH / UNIT TIME)
0.4
0.3
0.2
0.1
0.0
-0.1
-0.2
-0.3
-0.4
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
TIME (TIME UNITS)


STD. DEV. OF ACTUAL AND PREDICTED EST. ERROR - TYPE 12 PLOT
RANGE STANDARD DEVIATION (LENGTH)
0.0
0.1
0.2
0.3
0.4
0.5
TIME (TIME UNITS)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


COMPOSITE ERROR PLOT, SAMPLE OF PLOT TYPE 14
ANGLE RATE ERROR (DEGREES PER UNIT TIME)
TIME (TIME UNITS)
80.0
60.0
40.0
20.0
0.0
-20.0
-40.0
-60.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

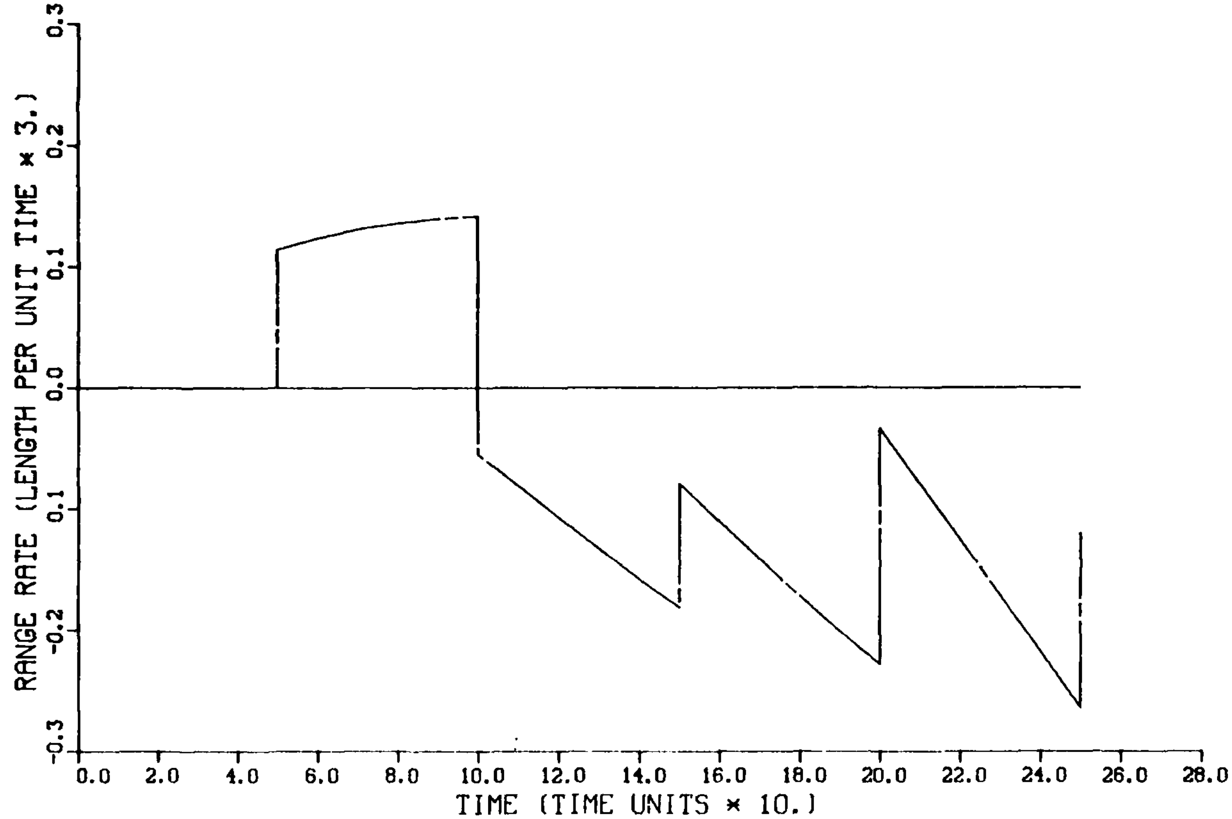


RANGE RESIDUAL, AVG. AND PREDICTED STD. DEV., PLOT TYPE 16
PLOT 5 11.22.30 THUR 27 AUG, 1981
DISSPLA VER 7.3
RESIDUAL (LENGTH UNITS)
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
TIME (TIME UNITS)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


TRAJECTORY IN CARTESIAN COORDINATES - PLOT TYPE 18
X AND Y POSITION, TRUE AND ESTIMATED (LENGTH UNITS)
TIME (TIME UNITS)
DISSPLA VER 7.3
-1.0
-0.5
0.0
0.5
1.0
1.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

PLOT 7 11.23.13 THUR 27 AUG, 1981 JOB-SHMQRBP , WPAFB/ASD DISSPLA VER 7.3

SAME AS PLOT 1 BUT WITH AXIS SCALING AND TIME WINDOWING



BLANK PAGE

60

APPENDIX B
DISSPLA POSTPROCESSORS

The DISSPLA plotting routines referenced within SOFEPL output a stream of compressed, device independent plot commands on the file PLFILE. It is the job of the DISSPLA postprocessors to transform these generalized plot commands into plot instructions for a particular plotting device. Thus the plot data from one SOFEPL run stored on PLFILE can be plotted on many different plotting devices simply by running the appropriate DISSPLA postprocessor. Two widely used postprocessors, ONLINE for an online Calcomp plotter and TEK4014 for an interactive Tektronix 4014 terminal, are reviewed here.

The ONLINE postprocessor is used to translate the DISSPLA plot commands into commands useable by an on-line Calcomp plotter. This postprocessor can be executed from any terminal or batch job. ONLINE uses the local file PLFILE as input and creates an output file called PLOT containing the Calcomp plot commands. This file is then routed to the on-line plot device. The following commands can be used to execute ONLINE.

```
ATTACH, DISSPLA, ID=LIBRARY, SN=ASD.
LIBRARY, DISSPLA.
ATTACH, PLFILE, ID=SHM.
ONLINE.
ROUTE, PLOT, DC=PU, FID=SHM, TID=AG.
```

In this use the plot file produced by the ONLINE postprocessor was routed to terminal AG to be plotted on the AG (Building 22) online Calcomp drum plotter. Upon typing the ONLINE command, the user will be prompted to enter DISSPLA postprocessor directives. These directives and their effects are explained later in this section.

The TEK4014 postprocessor is used to draw the plots directly on a Tektronix 4014 Computer Display Terminal. TEK4014 is an interactive processor executed while the user is logged in on the Tektronix terminal. The commands used to execute TEK4014 are

```
ATTACH, DISSPLA, ID=LIBRARY, SN=ASD.
LIBRARY, DISSPLA.
ATTACH,PLFILE, ID=SHM.
TEK4014.
```

The postprocessor then prompts the user for the needed directives. After the directives have been entered the postprocessor draws the plots on the screen one at a time, pausing after each plot for inspection or hard copying by the user.

Plot directives instruct the DISSPLA postprocessor to selectively plot, scale, window, and orient the ensuing plots on the actual plotting surface. The plot directives are the same for all the different DISSPLA postprocessors. The two most useful directives are the DRAW and the MODIFY commands. The DRAW command specifies which plots are to be drawn and has the form

```
DRAW=list
```

where "list" is the list of plots or series of plots to be drawn. For example:

```
DRAW=1,3,5-7,10-13,16*
```

would draw plots 1,3,5,6,7,10,11,12,13,16. The MODIFY command controls changes to the plots to be drawn and must follow the DRAW command. The MODIFY command has the form

```
MODIFY=list(spec1*spec2*...*specn)
```

where "list" is the list of plots to be modified and "spec" is a command defining a change to be made to the plots. Several specifications (spec) are available to change the size, position or portion of the plot to be drawn. Two such specifications are

SCALE and CORNER. A specification of SCALE=.75 acts to reduce the plot to 75 percent of its original size. The specification CORNER=1.5,1.5 would move the lower left corner up and to the right 1.5 inches. All postprocessors continue to read directives until a $, signifying end of directives, is encountered. A set of directives to draw the first seven plots on PLFILE, reduce their size, and recenter them is

```
DRAW=7*
MODIFY=7(SCALE=.75)*
MODIFY=7(CORNER=1.5,1.5)$
```

Note that the "*" is necessary to separate the individual commands.

A detailed discussion of all DISSPLA postprocessor commands is found in Reference 9.

BLANK PAGE

64

APPENDIX C
MERGING INTERFACE FILES

It is occasionally required that two or more 'similar' interface files (TAPE4s) be merged to form a single interface file so that SOFEPL plots can be generated from a composite data set. Two 'similar' interface files would exist, for example, if two SOFE studies were made of the same problem, one study of N_1 runs from $ISEED_1$, and a second study of N_2 runs from $ISEED_2$. Merging all runs into one master file of $N_1 + N_2$ runs produces a larger ensemble from which to draw conclusions.

In a second case, 'similar' interface files are also generated when one SOFE study is the time-continuation of a previous study. In this case, each study contains the same number of runs but the later study takes up in time where the first ends. SOFE provides this capability through its ICONT parameter. Again, benefits could derive from merging the TAPE4 outputs of these two studies into one file of longer runs.

This appendix discusses two programs for accomplishing file merging. The first is named AFFIXR and deals with merging two similar files containing runs over the same time interval. The second is named AFFIXT and deals with merging two similar files of equal numbers of runs where the second file is the time-continuation of the first.

Breaking a large study into two or more substudies may be advisable for any of these reasons: computer turn-around times for big runs are prohibitively long; a short preliminary evaluation of performance is needed before a lot of computer time is spent and possibly wasted; the analyst is unsure of the number of runs (or run lengths) required to reach statistical significance so he would prefer to monitor convergence rates as new runs are added gradually.

AFFIXR

AFFIXR (affix runs) merges two interface files that meet the following 'similarity' criteria:

- NS, NF, M same in both files
- TO, TMEASO, TF same in both files
- DTMEAS, DTCCPL same in both files

The IPASS value (number of runs) of each file may be different.

AFFIXR is designed to execute from an interactive terminal connected to the ASD CDC computer system. The formal names of AFFIXR's input files are TAPE1 and TAPE2, and that of its output file is TAPE3. If we assume that the file AFFIXR is a loadable object module, the program can be run by entering

AFFIXR, TAPE1, TAPE2.

AFFIXR checks both input TAPEs for compatibility, i.e., checks to see if they satisfy the similarity criteria listed above. If they are not compatible, AFFIXR prints diagnostic messages as to why they cannot be merged. If they are compatible, the program creates a TAPE3 output. TAPE3 is a valid data file for input to SOFEPL.

AFFIXR performs its merge operations in the following order. First, TAPEs 1 and 2 are rewound and their header records are read and compared. If the TAPEs are compatible, AFFIXR writes a new header to TAPE3 in which $IPASS_3 = IPASS_1 + IPASS_2$. This header also contains the data listed in the similarity criteria plus the run date, time and title from TAPE1. Second, all of the runs from TAPE1 are copied to TAPE3. Third, all the runs from TAPE2 are copied to TAPE3.

Repeated executions of AFFIXR can be used to merge any number of files. Suppose a user wants to merge files A, B, and C. He can do this, with the final output on E, using execution-time replacement of file names as follows:

AFFIXR, A, B, D

AFFIXR, D, C, E

AFFIXT

AFFIXT (affix time) merges two interface files that meet the following similarity criteria:

- NS, NF, M same in both files
- $TF_1 \leq TO_2$
- $\overline{IPASS}$ same in both files

The second criterion states that the two studies must not overlap in time; although studies may be separated by a time gap, the usual case will be that $TF_1 = TO_2$.

A feature of AFFIXT is that the two input files (TAPE1 and TAPE2 again) do not have to match in DTMEAS, DTCCPL or TMEASO. This feature is useful if one wants to merge mission segments of different types to obtain a composite mission. AFFIXT is executed in the same manner as AFFIXR with TAPE3 again being the output file.

AFFIXR and AFFIXT source codes can be attached from the CDC disk as follows:

```
ATTACH,AFFIXR,ID=SHM,SN=AFAL,CY=999

ATTACH,AFFIXT,ID=SHM,SN=AFAL,CY=999
```

BLANK PAGE

68

REFERENCES

1. S. H. Musick, "SOFE: A Generalized Digital Simulation for Optimal Filter Evaluation, User's Manual", AFWAL-TR-80-1108, Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, OH, October 1980.

2. W. J. Dixon and F. J. Massey, Introduction to Statistical Analysis (Second Edition), McGraw Hill Book Company, New York, 1957.

3. W. E. Deming, Some Theory of Sampling, John Wiley and Sons, New York, 1950.

4. CDC NOS/BE User's Guide, ASD Computer Center, WPAFB, OH.

5. UPDATE, Version 1, Reference Manual, Control Data Corporation publication number 60449900.

6. FORTRAN Extended Version 4, Reference Manual, Control Data Corporation publication number 60497800.

7. NOS/BE Version 1, Reference Manual, Control Data Corporation publication number 60493800.

8. DISSPLA Beginners/Intermediate Manual, Volume 1, Integrated Software Systems Corp., San Diego, California.

9. The DISSPLA Pre-processor/Post-processor User Manual", Integrated Software Systems Corp., San Diego, California.

ILMED

8