

AD-A154 995

KALMAN FILTER TIME SERIES ANALYSIS OF GAMMA-RAY DATA
FROM NAI(T1) DETECTORS FOR THE ND6620 COMPUTER(U) NAVAL
RESEARCH LAB WASHINGTON DC G W PHILLIPS 08 MAY 85

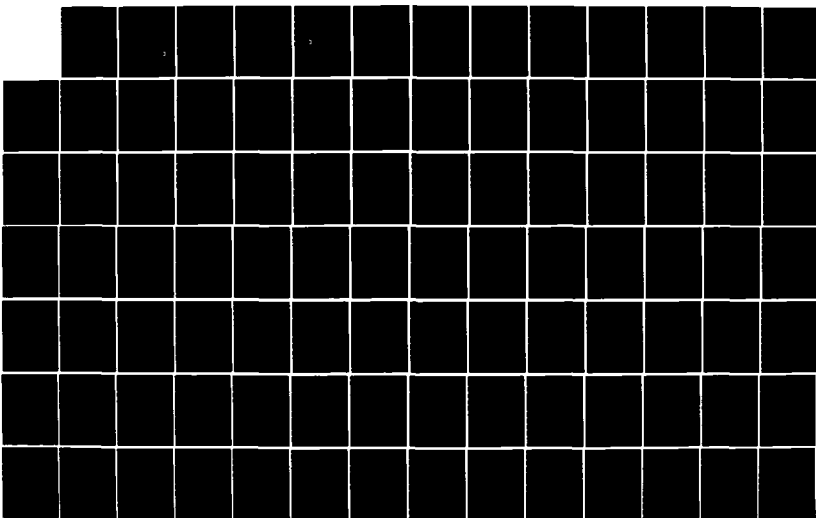
1/2

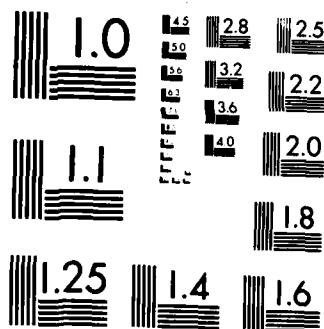
UNCLASSIFIED

NRL-MR-5541

F/G 20/6

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A154 995

2

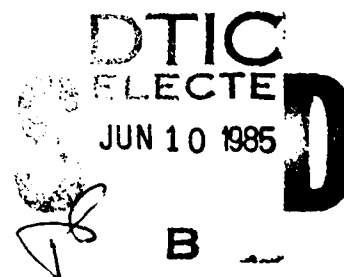
NRL Memorandum Report 5541

Kalman Filter
Time Series Analysis of Gamma-Ray Data
from NaI(Tl) Detectors for the ND6620 Computer

G. W. PHILLIPS

Radiation Detection Section
Condensed Matter and Radiation Sciences Division

May 8, 1985



DTIC FILE COPY

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release, distribution unlimited.

85 5 13 103

REPORT DOCUMENTATION PAGE				
1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b RESTRICTIVE MARKINGS	
2a SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.	
2b DECLASSIFICATION/DOWNGRADING SCHEDULE				
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NRL Memorandum Report 5541			5 MONITORING ORGANIZATION REPORT NUMBER(S)	
6a NAME OF PERFORMING ORGANIZATION Naval Research Laboratory		6b OFFICE SYMBOL (If applicable) Code 6616		7a NAME OF MONITORING ORGANIZATION
6c ADDRESS (City, State, and ZIP Code) Washington, DC 20375-5000			7b ADDRESS (City, State, and ZIP Code)	
9a NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research		8b OFFICE SYMBOL (If applicable) Code 1513		9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER ~~ND6620-116~~
8c ADDRESS (City, State, and ZIP Code) Arlington, VA 22217			10 SOURCE OF FUNDING NUMBERS	
			PROGRAM ELEMENT NO 33901N	PROJECT NO R7092
			TASK NO.	WORK UNIT ACCESSION NO. EX320-147
11 TITLE (Include Security Classification) Kalman Filter Time Series Analysis of Gamma-Ray Data from NaI(Tl) Detectors for the ND6620 Computer				
12 PERSONAL AUTHOR(S) Phillips, G.W.				
13a TYPE OF REPORT Final		13b TIME COVERED FROM TO		14 DATE OF REPORT (Year, Month, Day) 1985 May 8
15 PAGE COUNT 116				
16 SUPPLEMENTARY NOTATION				
17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP		
			NaI detectors, Time series analysis, Gamma-ray spectra, Kalman Filter.	
19 ABSTRACT (Continue on reverse if necessary and identify by block number) The Kalman Filter technique is applied to time series gamma-ray data from NaI(Tl) detectors. A set of standard spectra which contribute significantly to the data in the least-squares sense are first determined using regression analysis. The Kalman Filter computer program is then used to determine the time behavior for the intensities of the standards which contribute to the observed spectra. The program was written for the ND6620 computer in DEC RT-11 FORTRAN and is being transferred to the VAX 11/780. <i>search words include:</i>				
20 DISTRIBUTION AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS			21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL G. W. Phillips			22b TELEPHONE (Include Area Code) (202) 767-5692	22c OFFICE SYMBOL Code 6616

CONTENTS

INTRODUCTION	1
PREPARATION FOR KALMAN FILTER	1
THEORY OF OPERATION	1
FILTER TUNING	5
PROGRAM KLPREP	6
Operation	6
Language	6
Inputs	6
Outputs	6
Subroutines Called	6
Input Variables	7
Record 1	7
Record 2a,...	7
Record 3	7
Record 4	7
Record 5	7
Output Variables	8
Header	8
Data Records	8
PROGRAM KFILTR	9
Operation	9
Language	10
Inputs	10
Outputs	10
Subroutines Called	10
Variables	10
Input to KLIN	12
Input Variables from Keyboard	12
Record 1	12
Records 2a,...	12
Input to KINIT	12
Input Variables from Keyboard	12
Records 3a,...	12

Subroutines Called by KINIT	12
Output from KOUT	13
Lineprinter Output Columns	13
Disk File Output Columns	13
REFERENCES	13
APPENDIX A: Sample Outputs for File J0161	16
APPENDIX B: FORTRAN Listings, DEC RT-11	25
APPENDIX C: MIDAS System Utilities	77
APPENDIX D: IMSL* Subroutines	83

DTIC
ELECTE
JUN 10 1985
B

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Avail and/or	
Special	
A-1	

DTIC
COPY
INSPECTED
1

**KALMAN FILTER
TIME SERIES ANALYSIS OF GAMMA-RAY DATA
FROM NaI(Tl) DETECTORS FOR THE ND6620 COMPUTER**

INTRODUCTION

This program is intended for use on time series gamma-ray data from NaI(Tl) detectors. It is used in conjunction with the PAGSCN Data Screen¹ and the PREGA Regression Analysis² programs. The data consists of consecutive 256 channel gamma-ray spectra, each collected over a unit time period. Program PAGSCN screens the data for bad records and system malfunctions. The data are then summed over time into background and source spectra. Program PREGA is used to do a pivotal regression analysis of the source spectrum to a library consisting of the background plus a set of standard spectra. PREGA determines the subset of the library which gives the best fit to the source spectra in the least-squares sense. The Kalman Filter is then used to determine the time behavior for the intensities of the library spectra as components of the source spectrum.

PREPARATION FOR KALMAN FILTER

The analyst should be guided by the results of the PREGA least-squares analysis in selecting library sources for use in the Kalman Filter. In general, the background plus the sources found to be significant in the least-squares analysis will be used. Additional library sources may be included to check for possible interferences or correlations with the data. The gain and zero offset of the library spectra must be adjusted using Program GSHIFT² to match the values obtained from the energy calibration of the background. It is assumed that these values do not change between collection of the background and the source spectra.

THEORY OF OPERATION

The Kalman Filter^{1,2} provides an adaptive minimum variance estimate of the intensities of the various library spectra in the source spectrum at each 10 second record. It makes optimal use of a priori data from the results of the previous measurements and combines this with the current results to get a best estimate for the source intensities x_k at time k and their covariances P_k , given the observed spectra $y_1, y_2, \dots, y_k$.

The filter operates on a system model shown in Fig. 1 which relates the source intensities x_k at time k to the observed data y_k . The response matrix S_k describes the response of the detector system to the signal from the source. The expected output of the system is $S_k x_k$. Added to this is a system noise vector v_k which includes the random statistical variations in the detector system. The result is the observed system output

Manuscript approved January 23, 1985.

y_k . Generally the system noise v_k is assumed to be gaussian with a mean of zero and covariance R_k which is known or can be estimated.

The behavior of the source between times k and $k+1$ is assumed known and is modeled by the transition matrix H_k . The expected output is $H_k x_k$. Added to this is the input noise vector u_k which represents variations in the source intensities due to unknown effects or inadequacies in the model. The result is the vector x_{k+1} giving the source intensities at time $k+1$. The input noise is also generally assumed to be gaussian with zero mean and covariance Q_k which is known or can be estimated. Note that the input noise u_k drives the system, which is otherwise completely determined by the initial conditions x_0 and the transmission matrices $H_0, H_1, \dots, H_k$. Were it not for the input noise, our knowledge of the source vector x_k would continue to improve with each observation, and its covariance P_k would continued to decrease.

Given the above model, the Kalman Filter shown in Fig. 2 provides estimates for the source intensities $\hat{x}_{k/k-1}$ and covariance $P_{k/k-1}$ and for the expected system output $\hat{y}_k = S_k \hat{x}_k$. These are compared to the observed data y_k . The difference between observation and prediction is fed back with a gain K_k to provide a corrected estimate given by

$$\hat{x}_{k/k} = \hat{x}_{k/k-1} + K_k [y_k - \hat{y}_k].$$

The magnitude of the Kalman gain K_k depends on both the input covariance $P_{k/k-1}$ and the output covariance R_k .

At time step k , we begin with a priori estimates represented by $\hat{x}_{k/k-1}$ and $P_{k/k-1}$ based on data up to and including y_{k-1} . The updated a posteriori estimates, including the knowledge of the data y_k , are then given by

$$\hat{x}_{k/k} = [I - K_k S_k] \hat{x}_{k/k-1} + K_k y_k$$

$$P_{k/k} = [I - K_k S_k] P_{k/k-1}$$

where I is the diagonal identity matrix, and the Kalman gain

$$K_k = P_{k/k-1} S_k^T [S_k P_{k/k-1} S_k^T + R_k]^{-1}$$

where R_k is the diagonal matrix given by the Poisson variances in the data y_k .

The response matrix S_k is made up of elements $S_k(I,J)$ giving the response of the detector system in channel I to a unit source of type J. The columns of S_k thus contain the library spectra, and the elements $\hat{x}_k(J)$ contain the estimated intensities of source J at time k. We will assume that the system response does not change with time so that S_k is a constant matrix for all times k.

To obtain a priori estimates at time $k+1$, we can write

$$\begin{aligned}\hat{x}_{k+1/k} &= H_k \hat{x}_{k/k} \\ P_{k+1/k} &= H_k P_{k/k} H_k^T + Q_k\end{aligned}$$

The transition matrix H_k represents the known time behavior of the system, which we will take as constant so that $H_k = I$, the identity matrix. The input noise matrix Q_k , which we will take as diagonal, represents modeling errors and other unknown variations in the source term x_k . We will take it to be proportional to the square of a weighted mean $\bar{x}_k$ for times 1,2...,k

$$Q_k(I,J) = \delta_{IJ} q_0(J) \bar{x}_k(J)^2$$

or optionally, fix it at its value at some time k_0 ,

$$Q_k = Q_{k_0} \text{ for } k \geq k_0.$$

Currently we use $q_0(J) = 0.1$ for background and 0.3 for the rest of the library.

For a constant background and small input noise, the system will rapidly approach good estimates for x and P , regardless of the chosen initial values of x_0 and P_0 . If the signal then changes due to a real source, it will take several time steps to obtain good estimates for the new values of x and P . The magnitude of the input noise term Q relative to the output noise R determines how fast the filter can adjust to a change in the signal. A small Q leads to greater memory and thus smoother variations in the estimates x . A large Q leads to less memory resulting in a more immediate influence of a change in the observed data y and thus larger variations in the estimates $\hat{x}$. This is discussed in more detail below in the section on "Filter Tuning."

To start the filter we use initial values $\hat{x}_{0/-1}$ of 1.0 for background and equal to their estimated standard deviations for the other library members. We assume an initial error of 100 percent so that the initial variance is

$$P_{0/-1}(I,J) = \delta_{IJ} \hat{x}_{0/-1}(J)^2$$

To obtain output intensities in terms of standard deviation units, we calculate a normalized source term

$$n_k = (\hat{x}_{k/k} - \bar{x}_k) / v_k^{1/2}$$

where the weighted mean uses an iterative exponential weighting with constant slope, a , given by

$$\bar{x}_k = (a_k \bar{x}_{k-1} + \hat{x}_{k/k}) / b_k$$

where the constant in the numerator

$$a_k = \sum_{j=0}^{k-1} a^{(k-j)}.$$

In the denominator

$$\begin{aligned} b_k &= a_k + 1 \\ &= \sum_{j=0}^k a^{(k-j)}, \end{aligned}$$

so that $a_{k+1} = a b_k$. The numerator is equivalent to

$$\sum_{j=0}^k a^{(k-j)} \hat{x}_{j/j}$$

so for the constant slope, a , less than one we have exponentially declining weights as we go back in time. For large k we approach the limits

$$\lim_{k \rightarrow \infty} a_k = a/(1-a)$$

$$\lim_{k \rightarrow \infty} b_k = 1/(1-a).$$

An exponentially weighted variance, v_k is also calculated for $\hat{x}_{k/k}$, by

$$\bar{u}_k = (a_k \bar{u}_{k-1} + \hat{x}_{k/k}^2) / b_k$$

$$v_k = \bar{u}_k - (\bar{x}_k)^2.$$

FILTER TUNING

To further examine the effects of the parameters q_0 , consider the terms in the brackets for the expression for the Kalman gain

$$K_k = P_{k/k-1} S_k^T [S_k P_{k/k-1} S_k^T + R_k]^{-1}.$$

For the output noise R_k negligibly small, we note that

$$K_k \approx S_k^{-1}$$

$$\hat{x}_{k/k} \approx S_k^{-1} y_k$$

In this limit, the updated estimates depend only on the response matrix S_k and the observed data y_k at step k . All information prior to step k is ignored.

For negligibly small input noise, $P_{k/k-1}$ becomes negligibly small and the first term in the brackets can be neglected. Then

$$K_k \approx P_{k/k-1} S_k R_k^{-1} \approx 0$$

$$x_{k/k} \approx x_{k/k-1}$$

In this limit, the updated estimates depend only on the a priori estimates and the observed data y_k is ignored. The filter will tend to diverge from the data over time.

The input covariance estimate $P_{k/k-1}$ is kept from becoming negligibly small by the addition of the input noise matrix Q_k at each time step. Thus the filter is driven by the input noise. If Q is too small, the estimates will tend to diverge from the data. If Q is too large, they will depend only on the last observation, ignoring all previous data. The values of q_0 recommended in the previous section can be decreased or increased

depending on the amount of smoothing desired in the estimates as they vary with time.

PROGRAM KLPREP

Operation

This program reads the data tapes, extracts the gamma spectra, record ID, and MODE, condenses the gamma spectra from 256 to a fewer number of channels (typically sixteen) and writes the results to a disk file for use by the Kalman Filter.

Language

The program is written in DEC RT-11 FORTRAN and runs on the Nuclear Data ND6620 computer under the MIDAS operating system.

Inputs

Magtape	data files in NIAGARA format
Keyboard	logical unit (LU) 5, input in ASCII separated by blanks or commas.

Outputs

Diskfile	LU12, header and condensed spectra in format for input to program KFILTR
Lineprinter	LU6, header and condensed spectra in ASCII format. For diagnostic purposes, not normally printed out, sample output in Appendix A.

Subroutines Called

FREEFM	free field input routine (FORTRAN listing in Appendix B)
MCLI	utility to allow call of MIDAS system command from program (see Appendix C). Used to define LU12 as desired diskfile.
QANDC	utility to open diskfile on LU12 (see Appendix C)
DATIN	reads in data from the tapes (FORTRAN listing in Appendix B)
MTAPEF	tape input utility (see Appendix C)

BTIME utility to return day and date in integer format (see Appendix C)

DATOUT outputs data to LU12 (FORTRAN listing in Appendix B)

Input Variables

Record 1

NFSKIP number of files on tape to skip before beginning processing

NRSKIP number of records on tape to skip before beginning processing

NREC1 ID of first data record to process

NREC2 ID of last data record to process

Record 2a,...

NDREC(I) up to 16 bad records to delete, entered on one or more lines, separated by commas, terminated by double carriage return

Record 3

MS starting channel in input spectrum

MF final channel in input spectrum

NCH number of channels in condensed output spectrum. Each channel in the output spectrum will contain N input channels starting with MS and ending with MF, where $N = (MS - MF + NCH)/NCH$.

Record 4

FILE.ELEMENT filename for output on LU12

Record 5

IANS ASCII 'YES' or 'NO' in answer to whether to printout results on LU6

Output Variables

Header

HEADER(I)	header record
NREC1, NREC2	first and last output record
NTIME	total time spanned by data (seconds)
NCH	number of channels in output spectrum
MS,ME	starting and final channels in input spectrum

Data Records

NREC	record ID number
NID	MODE switch
I1(I)	output spectrum for Pod 1
I2(I)	output spectrum for Pod 2

470	16	-1.493	0.614	-5.50	1.40	2.05	-3.83	2.23	-2.30
471	16	-0.228	-1.11	-0.274	-1.52	1.75	-4.08	2.74	-1.69
472	16	0.475	-1.41	-7.03	-5.20	5.738E-02	-1.66	1.65	-1.52
473	16	-0.334	-3.730E-02	3.36	-4.48	-1.06	-0.564	1.40	-1.01
474	16	-3.31	1.36	-1.37	-2.84	-0.608	-0.351	4.850E-02	-2.34
475	16	-2.07	-0.229	0.116	1.15	-1.15	-0.893	-0.807	0.399
476	16	-1.25	-0.504	-5.25	5.422E-02	-2.39	-1.07	0.403	0.757
477	16	-2.39	1.05	-6.05	3.18	-1.40	-1.54	0.503	0.814
478	16	-2.63	7.754E-02	-7.21	4.33	-1.40	-1.46	-1.46	2.33
479	16	-0.210	-0.439	-3.44	0.129	-0.792	-2.02	0.824	1.72
480	16	1.793	-0.731	6.858E-02	-1.99	0.700	-1.49	-4.587E-02	-1.026E-02
481	14	1.33	-0.360	-4.73	-1.44	0.880	-2.66	0.327	0.710
482	14	0.617	-0.331	1.23	-3.01	0.551	-0.526	-6.341E-02	0.789
483	14	0.863	0.309	0.476	-3.66	2.44	-1.33	0.632	-0.580
484	14	-1.36	1.39	2.27	1.13	4.28	-1.38	2.92	-2.07
485	17	1.86	-1.17	5.98	0.289	0.172	2.30	10.6	-1.12
486	17	1.34	-0.276	7.07	-1.99	-1.46	4.59	16.2	0.470
487	17	-0.371	1.06	11.9	0.137	-3.73	7.40	22.3	0.257
488	17	-1.84	2.25	17.0	0.704	-8.60	7.46	38.7	1.78
489	17	-1.75	1.22	22.1	1.93	-12.6	10.8	43.9	2.90
490	17	-1.70	2.65	19.9	-0.574	-8.44	7.54	46.8	3.42
491	17	0.410	0.945	19.8	-2.53	-3.58	5.49	38.8	1.89
492	17	0.371	1.36	29.5	-0.975	-12.8	11.6	51.5	4.07
493	17	0.280	0.448	27.0	0.308	-9.23	9.67	47.4	2.46
494	17	2.29	-1.43	17.2	-0.484	-6.03	4.86	48.3	3.42
495	17	4.076E-02	0.136	17.7	0.558	-4.05	4.44	40.3	1.89
496	17	9.328E-02	0.963	17.0	0.137	-5.12	3.24	51.8	1.73
497	17	0.443	1.79	12.3	0.141	-6.49	4.88	50.3	3.33
498	17	-1.91	2.63	26.2	2.33	-4.19	2.28	48.9	2.18
499	17	1.80	-1.17	3.62	1.34	-1.51	7.49	5.37	-0.562
500	17	0.357	-1.13	7.78	2.35	-1.23	2.88	14.6	-1.28
501	14	-0.550	0.853	0.763	3.20	1.08	-2.08	26.0	-3.63
502	14	-4.91	4.10	-3.93	6.39	0.433	-1.10	12.5	-1.74
503	14	-0.364	-0.240	6.57	2.67	-4.84	4.47	1.53	1.30
504	14	1.07	-0.314	-3.30	-0.459	1.03	-2.16	7.10	-1.24
505	14	3.833E-02	0.136	-4.19	0.909	-2.64	3.75	0.405	0.531
506	14	1.32	-1.41	-4.22	-0.666	1.91	-1.87	4.20	0.173
507	14	-0.673	4.225E-02	0.511	1.22	2.50	-5.29	10.5	-3.11
508	14	2.06	-1.49	-2.50	-1.25	0.465	-2.70	2.32	1.49
509	14	1.36	-2.13	-5.92	-0.575	-0.745	-0.987	2.64	1.76
510	18	0.817	-0.551	3.95	2.10	-1.20	0.889	2.56	1.14
511	18	-0.913	1.58	-2.55	0.785	-1.42	0.340	4.83	1.63
512	18	-0.741	2.20	-2.31	0.506	-1.05	-1.49	25.1	0.646
513	18	-2.18	3.64	-6.92	0.513	-3.52	4.56	15.5	1.34
514	18	-0.208	1.88	10.3	-0.623	-6.37	5.53	38.5	2.32
515	18	1.50	1.70	15.0	0.249	-10.1	8.90	57.5	2.65
516	18	0.522	0.388	24.6	3.26	-3.44	6.05	46.3	3.36
517	18	0.864	0.740	20.1	-0.983	-5.30	5.71	53.3	2.18
518	18	0.813	-1.35	13.9	2.91	-4.50	7.20	23.9	1.79
519	18	-0.344	1.11	3.29	1.133E-02	-5.17	4.64	21.2	0.829
520	18	-3.16	1.53	9.93	2.59	-2.12	0.648	8.76	0.189
521	18	-1.64	0.832	2.71	1.24	0.167	-0.534	0.260	-1.61
524	1	0.276	-0.341	-3.36	0.844	1.07	-0.466	-2.61	0.839
525	1	-0.958	0.246	5.27	-1.85	2.35	-2.20	-2.98	0.543

00161.DD 14 JAN 1995 11:46:55 AM

REC.	MODE	011161	RA26P1	CO60P1	TH32P1	B12161	RA26P2	CO60P2	TH32P2
420	1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
421	1	-0.975	0.975	0.975	-0.975	-0.975	0.975	0.975	0.975
422	1	-1.33	1.36	-1.27	0.844	-1.36	1.18	1.16	-0.881
423	1	-1.39	1.51	-0.338	-0.465	1.13	-1.02	-1.42	-1.63
424	1	-0.630	0.790	1.66	-0.321	0.694	-0.782	-1.39	-1.03
425	1	1.51	-1.24	-1.13	0.665	0.539	-0.565	-0.447	-1.04
426	1	-0.900	0.100	0.734	0.673	-0.771	0.379	0.459	1.34
427	1	-0.465	0.413	-0.729	0.825	-0.669	1.15	-0.269	-0.240
428	1	0.265	-0.197	0.706	0.306	-0.746	1.41	1.37	-1.73
429	1	1.63	-1.63	-0.695	1.02	-0.687	0.673	1.92	-0.883
430	1	0.472	-0.240	-9.315E-02	-0.170	0.798	-0.597	-1.92	-1.59
431	1	-0.616	0.815	1.32	0.332	-0.169	0.218	-2.00	1.61
432	1	0.325	-0.864	0.220	0.989	0.752	0.276	-0.206	-1.28
433	1	1.45	-1.01	1.32	-1.60	0.774	-0.237	-1.30	-1.39
434	15	0.222	-0.377	-0.402	-1.91	-0.451	-0.938	0.793	0.601
435	15	-2.28	1.73	-2.41	2.62	1.65	-2.45	0.375	-4.549E-02
436	15	-1.44	0.869	-2.27	1.55	1.54	-1.94	1.24	-1.15
437	15	-1.97	0.897	-2.10	5.61	-0.690	-1.30	3.63	0.496
438	15	0.463	-1.303E-02	-2.39	-1.48	-0.631	-0.887	3.49	-1.07
439	15	1.07	-0.562	-1.54	-1.65	0.544	-1.59	3.59	-1.40
440	15	-1.12	1.11	-4.07	0.719	2.04	-1.14	0.638	-2.26
441	15	-0.369	-0.952	-2.12	1.42	-0.234	-1.05	-0.745	-1.90
442	15	-1.44	0.293	-5.28	1.89	0.258	-2.01	-1.72	-0.882
443	15	0.392	-2.12	-0.364	0.999	1.31	-2.58	0.512	-1.93
444	15	-3.52	1.33	-4.54	3.66	-1.33	-0.878	1.12	6.075E-02
445	15	-4.00	2.41	-0.397	2.10	1.11	-2.40	1.84	-0.858
446	15	-1.02	0.344	-5.48	1.14	1.68	-2.63	0.670	-0.387
447	15	1.56	-1.21	-2.05	0.145	3.25	-2.67	-1.15	-0.278
448	15	-0.721	-3.084E-02	-2.35	2.06	-0.434	-1.99	0.200	0.243
449	15	-3.70	2.09	-5.95	5.61	-0.350	-1.03	-0.777	-1.00
450	15	0.166	-0.588	-8.56	-1.87	1.54	-1.70	-1.23	-1.75
451	15	0.577	-5.295E-02	-12.0	0.803	-0.396	-0.519	-0.749	-4.165E-03
452	15	-1.48	-0.643	-0.467	5.28	-1.06	0.444	-0.679	-0.747
453	15	0.621	-1.38	-2.44	5.558E-03	0.378	-1.97	-1.41	-0.806
454	15	1.58	-3.65	0.799	0.336	-0.549	-1.55	0.530	-0.602
455	15	2.40	-3.49	0.507	-2.81	-2.42	-0.902	-0.607	0.389
456	15	-1.92	-0.170	-2.00	2.63	-2.263E-02	-1.62	-1.61	-1.08
457	15	-1.64	0.195	-0.730	0.695	-0.125	-2.49	0.522	-6.996E-02
458	15	-1.43	0.409	-7.00	0.389	0.129	-2.36	0.737	0.351
459	15	-0.851	-0.764	-4.36	0.670	0.439	-4.08	0.614	-0.191
460	15	-1.53	0.619	-5.86	1.80	-0.933	-2.19	1.09	-0.792
461	15	-2.67	1.95	-6.75	-0.692	1.43	-3.92	1.18	-1.09
462	14	-2.36	1.25	-2.79	-0.346	0.650	-3.37	1.16	0.264
463	14	-1.28	7.264E-02	-3.11	-1.32	1.34	-3.05	-0.236	-1.27
464	14	1.15	-2.46	-1.61	-3.06	3.40	-3.39	0.251	-1.84
465	14	-1.02	-2.16	9.69	-2.36	2.94	-3.35	0.541	-2.63
466	14	-1.49	-0.489	3.89	-1.65	0.220	-2.00	-2.937E-02	-1.30
467	14	1.40	-2.39	-3.49	-1.86	1.83	-4.22	0.529	-0.352
468	14	-3.053E-02	-1.15	-5.52	0.558	0.825	-2.15	1.10	-0.398
469	14	-4.61	1.34	2.97	4.85	1.42	-3.09	0.947	-1.57

518	18	4.47	1.31	0.990	0.159	-4.36E-02	1.207E-03	0.789	0.417	* 0.361	* 1.515E-02
519	18	4.47	1.31	0.988	3.872E-04	3.301E-02	1.159E-02	0.814	0.498	* 0.163	* 1.365E-02
520	18	2.16	6.57	0.939	0.187	6.634E-03	2.062E-03	0.793	0.370	* 0.145	* 4.757E-03
521	18	2.00	9.24	0.820	0.219	2.309E-02	1.053E-02	0.888	0.143	6.030E-02	* 5.105E-04
522	18	-0.05	4.90	0.084	0.166	5.204E-03	6.089E-03	0.958	7.876E-02	2.786E-03	-1.523E-02
523	0										
524	1	-0.32	1.96	0.965	7.684E-02	-9.807E-03	4.797E-03	0.986	8.242E-02	-1.665E-02	4.814E-03
525	1	2.09	-0.63	0.913	0.121	1.155E-02	4.055E-03	1.03	* -1.147E-02	-1.911E-02	2.309E-03
526	1	-0.66	1.13	0.956	4.149E-02	1.993E-02	3.064E-03	0.971	2.112E-03	1.503E-02	* 2.711E-03
527	1	1.41	2.13	0.936	4.944E-02	1.916E-02	2.507E-03	0.921	0.187	-2.024E-02	7.283E-03
528	1	0.03	2.61	0.975	5.499E-02	-2.671E-03	8.822E-04	1.03	* 2.712E-02	-2.004E-02	-7.846E-03
529	1	0.36	0.49	0.998	9.256E-03	9.134E-03	6.139E-04	1.01	3.490E-02	-7.749E-03	-1.555E-02
530	1	0.23	0.32	0.973	3.761E-02	1.588E-02	1.228E-03	0.949	9.710E-02	-1.196E-02	4.859E-03
531	1	1.90	0.59	0.976	9.286E-02	1.734E-02	8.246E-03	0.922	0.108	3.403E-03	7.604E-03
532	1	2.65	0.58	0.909	0.171	-1.247E-02	5.485E-03	0.860	0.247	* -1.706E-03	1.309E-02
533	1	0.47	0.83	0.926	7.387E-02	5.227E-03	4.636E-03	0.925	0.103	-1.489E-02	1.834E-02
534	1	1.27	1.07	0.964	0.109	-9.193E-03	1.866E-04	0.854	0.115	-5.713E-04	2.967E-03
535	1	-0.04	3.13	0.990	7.670E-02	-1.357E-02	-5.938E-03	0.922	0.192	1.222E-03	3.185E-03
536	1	1.09	0.78	1.05	* -2.810E-02	-1.462E-02	-2.537E-03	0.882	0.224	* 9.857E-03	7.452E-03
537	1	1.12	1.50	1.05	* -4.967E-02	-1.209E-02	-5.981E-03	1.04	* -0.128	1.495E-02	* 5.888E-03
538	1	0.28	2.73	1.12	* -0.140	-3.092E-05	-8.184E-03	1.01	-7.739E-02	5.097E-03	4.372E-03
539	1	1.03	1.94	1.10	* -0.178	-7.146E-03	-2.129E-03	0.941	6.558E-02	1.022E-02	-7.513E-03
540	1	1.35	1.47	0.936	9.739E-02	5.459E-03	3.267E-03	0.998	2.919E-02	-5.289E-03	-6.570E-03
541	1	0.21	-0.12	0.925	9.013E-02	4.092E-03	1.429E-03	1.05	* 6.150E-03	-1.265E-02	-1.147E-02
542	1	2.22	0.41	0.951	2.936E-02	1.502E-02	1.519E-03	0.996	6.057E-02	7.717E-03	-1.678E-02
543	1	3.16	2.04	0.872	0.210	-1.158E-02	4.279E-03	0.945	0.113	-1.148E-02	3.591E-03
544	1	-0.08	2.85	1.01	-1.746E-02	-7.434E-03	2.173E-03	1.01	-9.607E-02	2.130E-02	* 3.519E-03
545	1	0.33	0.79	0.985	-1.747E-02	-0.463E-03	3.591E-03	1.01	-7.877E-02	7.970E-03	4.725E-03
546	1	1.96	0.46	1.00	-1.293E-02	2.607E-03	-4.242E-03	1.01	-0.140	1.814E-02	* 4.314E-03
547	1	0.43	-0.35	0.985	1.007E-03	1.270E-03	1.520E-05	0.960	-4.113E-02	1.948E-02	* 1.273E-02
548	1	0.49	1.59	1.04	* -6.040E-02	9.866E-03	-1.892E-03	0.905	0.109	-4.764E-03	1.495E-02
549	1	1.10	-0.44	0.928	0.118	1.711E-02	* -6.050E-03	0.938	5.988E-02	-1.407E-02	1.497E-02
550	1	1.30	2.22	0.877	0.230	1.796E-03	-3.510E-03	0.963	0.145	-4.897E-03	-5.369E-03
551	1	2.01	0.01	1.01	2.970E-02	2.196E-03	-7.394E-03	0.995	3.449E-02	1.235E-02	-7.743E-03
552	1	3.40	0.27	0.962	0.151	-1.536E-02	-2.938E-03	0.965	4.767E-02	9.329E-03	-1.153E-02
553	1	0.50	1.26	0.995	5.525E-02	-5.850E-03	-3.107E-03	0.975	2.795E-02	3.068E-02	* -1.041E-02
554	1	1.31	1.40	0.992	2.972E-02	1.931E-03	-1.940E-03	0.968	-2.662E-03	3.329E-02	* -6.568E-03
555	1	-0.57	-0.61	1.04	* -8.450E-02	-1.539E-03	-1.187E-03	0.995	-1.691E-02	1.374E-02	2.100E-03
556	1	0.38	0.51	1.08	* -0.111	2.804E-03	-5.822E-03	1.00	7.047E-03	-5.793E-03	-4.846E-03
557	1	2.21	1.12	0.908	0.103	1.634E-02	* 3.619E-03	0.969	7.047E-03	2.625E-03	-7.841E-03
558	1	2.04	-0.33	0.936	1.925E-03	1.481E-02	3.279E-03	1.04	* 3.097E-03	-5.910E-03	-1.017E-02
559	1	0.80	0.48	0.974	-1.324E-02	2.817E-03	3.575E-03	0.989	7.929E-02	-6.260E-03	-2.387E-03
560	1	-0.41	1.31	0.953	6.555E-02	-1.538E-03	3.795E-03	0.965	7.121E-02	-4.862E-03	2.545E-03

453	14	-9.19	1.78	0.899	0.108	-9.19E-03	-2.298E-03	0.995	-5.750E-02	-5.730E-04	-1.243E-02
464	14	1.07	1.36	1.00	-8.417E-02	-5.467E-03	-8.019E-03	1.06	*7.588E-02	2.726E-03	-1.714E-02
465	14	2.81	1.63	0.910	-6.140E-02	2.002E-02	-5.729E-03	1.04	*7.409E-02	4.687E-03	-2.354E-02
466	14	2.66	0.76	0.890	6.569E-02	8.147E-03	-3.414E-03	0.960	-9.626E-04	8.263E-04	-1.274E-02
467	14	0.57	1.16	1.01	-0.117	-1.612E-02	-4.074E-03	1.01	-0.121	4.603E-03	-4.935E-03
468	14	1.14	0.80	0.949	1.523E-02	-1.519E-02	3.860E-03	0.979	-8.889E-03	8.461E-03	-5.316E-03
469	14	4.47	1.41	0.758	0.205	5.851E-03	1.795E-02	0.997	-6.008E-02	7.434E-03	-1.490E-02
470	16	1.56	0.83	0.884	0.149	-1.511E-02	6.611E-03	1.02	*9.976E-02	1.612E-02	-2.064E-02
471	16	1.65	1.07	0.922	1.856E-02	-2.071E-03	-3.393E-03	1.01	-0.114	1.959E-02	-1.591E-02
472	16	1.55	1.49	0.973	-4.957E-03	-1.888E-02	-1.507E-02	0.955	1.795E-02	1.221E-02	-1.453E-02
473	16	4.25	1.57	0.924	9.535E-02	6.585E-03	-1.269E-02	0.920	7.713E-02	1.052E-02	-1.686E-02
474	16	2.22	1.52	0.814	0.251	-4.893E-03	-7.299E-03	0.935	8.869E-02	1.333E-03	-1.878E-02
475	16	2.81	0.44	0.832	8.534E-02	-1.205E-03	5.810E-03	0.918	5.930E-02	-4.433E-03	1.217E-03
476	16	1.01	1.41	0.901	6.448E-02	-1.448E-02	2.203E-03	0.879	4.954E-02	3.752E-03	4.149E-03
477	16	1.33	-0.22	0.837	0.183	-1.846E-02	1.246E-02	0.910	2.420E-02	4.426E-03	4.617E-03
478	16	0.18	0.29	0.842	0.109	-1.934E-02	1.627E-02	0.910	2.843E-02	-8.864E-03	1.705E-02
479	16	0.47	-0.37	0.915	6.942E-02	-9.999E-03	2.447E-03	0.929	-1.921E-03	6.600E-03	1.204E-02
480	16	2.79	1.31	0.987	4.723E-02	-1.321E-03	-4.501E-03	0.975	2.667E-02	7.147E-04	-2.139E-03
481	14	-0.19	0.90	1.02	2.981E-02	-1.334E-02	-2.697E-03	0.981	-3.668E-02	3.237E-03	3.759E-03
482	14	-0.51	1.87	0.979	7.303E-02	1.670E-03	-7.881E-03	0.970	5.749E-02	5.961E-04	4.409E-03
483	14	0.22	2.08	0.981	0.126	-3.119E-04	-9.594E-03	1.03	*3.540E-02	5.298E-03	-6.804E-03
484	14	0.84	3.59	0.896	0.209	-1.130E-03	5.754E-03	1.09	*3.309E-02	2.081E-02	-1.900E-02
485	17	2.73	6.63	1.03	1.356E-02	1.332E-02	2.973E-03	0.959	0.232	*7.265E-02	-1.122E-02
486	17	0.40	4.95	1.01	8.178E-02	1.600E-02	-4.525E-03	0.908	0.357	*0.111	*1.796E-03
487	17	0.77	3.36	0.912	0.183	2.798E-02	2.475E-03	0.838	0.509	*0.152	*5.039E-05
488	17	4.32	4.19	0.875	0.274	*4.047E-02	4.337E-03	0.687	0.512	*0.263	*1.249E-02
489	17	0.55	2.65	0.879	0.196	5.322E-02	8.369E-03	0.562	0.691	*0.298	*2.169E-02
490	17	1.90	3.38	0.881	0.304	*4.785E-02	1.391E-04	0.692	0.517	*0.318	*2.599E-02
491	17	1.26	0.55	0.970	0.175	4.757E-02	-6.502E-03	0.843	0.406	*0.264	*1.340E-02
492	17	3.62	4.20	0.969	0.206	6.915E-02	-1.178E-03	0.557	0.734	*0.349	*1.614E-02
493	17	1.03	1.03	0.965	0.137	6.546E-02	2.036E-03	0.668	0.632	*0.322	*1.814E-02
494	17	2.05	1.69	1.05	*5.942E-03	4.098E-02	4.335E-04	0.767	0.371	*0.328	*2.596E-02
495	17	1.85	3.10	0.955	0.113	4.234E-02	3.860E-03	0.828	0.348	*0.274	*1.346E-02
496	17	1.58	2.88	0.957	0.176	4.061E-02	2.475E-03	0.795	0.283	*0.352	*1.212E-02
497	17	1.31	1.23	0.972	0.239	2.900E-02	2.488E-03	0.752	0.373	*0.341	*2.520E-02
498	17	1.22	1.68	0.872	0.306	*6.337E-02	9.689E-03	0.824	0.231	*0.331	*1.577E-02
499	17	6.07	19.21	1.03	1.374E-02	7.468E-03	6.418E-03	0.907	0.514	*3.733E-02	*6.660E-03
500	17	0.64	2.37	0.968	1.681E-02	1.777E-02	9.736E-03	0.915	0.264	*9.948E-02	-1.258E-02
501	14	0.72	2.43	0.930	0.168	4.111E-04	1.254E-02	0.907	-5.127E-03	0.177	*3.181E-02
502	14	3.66	6.23	0.746	0.414	*-1.123E-02	2.299E-02	0.967	4.819E-02	8.539E-02	*-1.630E-02
503	14	0.77	5.53	0.917	8.450E-02	1.478E-02	1.081E-02	0.804	0.350	*1.138E-02	8.603E-03
504	14	0.92	1.76	0.998	7.894E-02	-9.665E-03	5.158E-04	0.985	-9.224E-03	4.904E-02	-1.225E-02
505	14	-0.06	2.16	0.955	0.113	-1.187E-02	4.682E-03	0.872	0.311	*3.766E-03	2.292E-03
506	14	-0.38	1.79	1.03	-4.497E-03	-1.194E-02	-1.628E-04	1.01	6.582E-03	2.944E-02	*-6.356E-04
507	14	1.02	3.17	0.925	0.106	2.217E-05	6.027E-03	1.03	*-0.179	7.202E-02	-2.759E-02
508	14	1.06	2.80	1.04	*-1.025E-02	-7.690E-03	-2.069E-03	0.968	-3.871E-02	1.672E-02	*1.012E-02
509	14	-0.57	0.00	1.04	-5.931E-02	-1.615E-02	1.361E-04	0.930	5.961E-02	1.890E-02	*1.233E-02
510	18	3.76	-0.23	0.988	6.089E-02	8.297E-03	8.935E-03	0.916	0.156	1.831E-02	*7.240E-03
511	18	1.55	0.15	0.915	0.223	-7.801E-03	4.604E-03	0.909	0.126	3.370E-02	*1.131E-02
512	18	-0.11	10.04	0.922	0.270	*-7.209E-03	3.686E-03	0.921	2.697E-02	0.171	*3.239E-03
513	18	-1.40	3.84	0.865	0.379	*-1.861E-02	3.712E-03	0.845	0.355	*0.106	*8.940E-03
514	18	4.75	10.41	0.944	0.245	2.394E-02	-2.219E-05	0.756	0.407	*0.261	*1.696E-02
515	18	5.31	8.50	1.02	0.232	3.565E-02	2.844E-03	0.640	0.590	*0.390	*1.966E-02
516	18	2.21	2.44	0.975	0.132	5.939E-02	1.275E-02	0.847	0.436	*0.314	*2.544E-02

OUTPUT FOR FALMAN FILTER
 USING DATA FROM CHANNEL 19 TO 242
 CONDENSED TO 16 OUTPUT VECTOR CHANNELS
 THRESHOLD IS 3.00

FINAL LEARNING RECORDS, 9999 FOR POD 1, 9999 FOR POD 2
 INPUT VECTOR FOR POD1 HAS 4 VARIABLE INTENSITIES, LEAVING 12 DEGREES OF FREEDOM
 INPUT VECTOR FOR POD2 HAS 4 VARIABLE INTENSITIES, LEAVING 12 DEGREES OF FREEDOM

REC.	MODE	XS01	XS02	B11161	RA26P1	C060P1	TH32P1	B12161	RA26P2	C060P2	TH32P2
420	1	1.09	-1.54	0.974	4.844E-02	-2.277E-03	3.755E-03	0.962	4.209E-02	4.235E-03	5.512E-03
421	1	1.85	2.73	0.963	8.352E-02	-9.009E-04	-5.634E-03	0.955	0.119	6.165E-03	6.656E-03
422	1	3.44*	1.55	0.923	-4.600E-03	5.202E-03	0.879	0.196	7.858E-03	5.279E-03	5.279E-03
423	1	1.77	2.29	0.685	0.254	-3.290E-03	-1.604E-03	1.00	2.607E-02	1.323E-03	-2.639E-03
424	1	0.62	1.23	0.907	0.213	2.685E-03	-1.187E-03	0.987	2.929E-02	-9.528E-04	-1.769E-03
425	1	1.77	0.73	1.01	2.026E-02	-5.491E-03	3.048E-03	0.984	3.786E-02	1.911E-03	-3.007E-03
426	1	0.91	2.45	0.903	0.137	2.445E-05	3.469E-03	0.927	9.904E-02	4.796E-03	8.934E-03
427	1	0.19	0.95	0.915	0.165	-4.384E-03	4.426E-03	0.928	2.686E-03	2.686E-03	1.366E-03
428	1	-1.56	2.45	0.945	0.119	-2.319E-04	2.724E-03	0.921	7.950E-03	7.950E-03	8.820E-03
429	1	0.20	1.78	1.02	-2.033E-02	-3.990E-03	5.530E-03	0.924	0.200	1.262E-02	-4.717E-03
430	1	1.40	2.02	0.968	9.241E-02	-2.518E-03	1.617E-03	0.976	6.797E-02	-6.066E-03	-1.131E-02
431	1	1.53	2.38	0.920	0.181	1.373E-03	3.230E-03	0.943	0.120	-1.144E-02	1.203E-02
432	1	1.97	1.47	0.959	4.585E-02	-1.328E-03	5.503E-03	0.974	0.125	6.529E-04	-1.063E-02
433	1	2.60	1.00	1.01	2.613E-02	1.765E-03	3.226E-03	0.977	9.486E-02	-7.786E-03	-1.344E-02
434	15	1.06	0.56	0.963	7.411E-02	-2.485E-03	-4.238E-03	0.939	5.684E-02	6.386E-03	2.064E-03
435	15	4.20*	0.74	0.857	0.234	-7.465E-03	1.083E-02*	1.00	-2.489E-02	3.559E-03	-2.478E-03
436	15	-1.30	1.67	0.893	0.169	-7.099E-03	7.132E-03	1.00	2.273E-03	9.301E-03	-1.144E-02
437	15	0.97	1.85	0.887	0.171	-6.694E-03	2.047E-02*	0.932	3.723E-02	2.558E-02*	2.004E-03
438	15	1.80	2.20	0.973	0.102	-7.414E-03	-2.823E-03	0.934	5.960E-02	2.464E-02*	-1.084E-02
439	15	0.06	1.44	0.998	6.008E-02	-5.314E-03	3.403E-03	0.970	2.157E-02	2.532E-02*	-1.340E-02
440	15	0.70	1.77	0.906	0.187	-1.157E-02	4.389E-03	1.02	* 4.577E-02	5.343E-03	-2.058E-02
441	15	1.39	3.88*	0.938	3.043E-02	-6.729E-03	6.701E-03	0.944	5.070E-02	-4.017E-03	-1.600E-02
442	15	-0.30	-0.64	0.892	0.125	-1.457E-02	8.236E-03	0.961	-1.339E-03	-1.063E-02	-9.283E-03
443	15	1.23	1.88	0.970	-5.813E-02	-3.630E-03	5.309E-03	0.994	-3.228E-02	4.491E-03	-1.707E-02
444	15	2.69	1.08	0.884	0.246	-1.297E-02	1.407E-02*	0.912	6.011E-02	8.573E-03	-1.492E-03
445	15	0.33	0.38	0.784	0.286	*-3.958E-03	8.939E-03*	0.988	-2.248E-02	1.348E-02	-9.085E-03
446	15	2.29	-1.02	0.910	0.174	-1.505E-02	5.787E-03	1.01	-3.508E-02	5.554E-03	5.227E-03
447	15	1.95	1.40	1.02	1.064E-02	-6.550E-03	2.502E-03	1.05	*-3.680E-02	-6.763E-03	-4.336E-03
448	15	2.51	4.32*	0.923	9.661E-02	-7.298E-03	8.782E-03*	0.940	-8.650E-05	2.375E-03	-6.973E-05
449	15	2.39	1.36	0.797	0.262	*-1.621E-02	2.045E-02*	0.942	5.206E-02	-4.231E-03	-1.029E-02
450	15	0.85	0.79	0.960	5.810E-02	-2.267E-02	-4.125E-03	1.00	1.545E-02	-7.285E-03	-1.637E-02
451	15	3.36*	2.23	0.978	9.873E-02	-3.119E-02	4.665E-03	0.941	7.956E-02	-4.038E-03	-2.084E-03
452	15	2.22	3.56*	0.891	5.393E-02	-2.647E-03	1.936E-02*	0.920	0.132	-3.568E-03	-8.177E-03
453	15	0.08	2.00	0.979	-1.911E-03	-7.542E-03	2.043E-03	0.984	1.086E-03	-8.533E-03	-8.659E-03
454	15	2.04	1.25	1.02	-0.174	4.872E-04	3.127E-03	0.936	2.369E-02	4.612E-03	-6.983E-03
455	15	0.70	2.22	1.05	*-0.162	-2.369E-04	-7.209E-03	0.879	5.833E-02	-3.078E-03	1.134E-03
456	15	2.32	1.68	0.872	8.982E-02	-6.445E-03	1.066E-02*	0.953	1.985E-02	-9.843E-03	-1.091E-02
457	15	0.34	2.12	0.884	0.118	-3.299E-03	4.308E-03	0.949	-2.717E-02	4.553E-03	-2.626E-03
458	15	0.55	-0.24	0.893	0.134	-1.881E-02	3.304E-03	0.957	-2.034E-02	6.012E-03	8.205E-04
459	15	0.94	1.20	0.917	4.470E-02	-1.227E-02	4.227E-03	0.967	-0.113	5.179E-03	-3.623E-03
460	15	2.18	1.28	0.889	0.150	-1.600E-02	7.955E-03	0.924	-1.103E-02	8.390E-03	-8.546E-03
461	15	2.51	1.59	0.840	0.251	-1.819E-02	-2.509E-04	0.997	-0.105	9.002E-03	-1.101E-02
462	14	-0.52	0.71	0.853	0.198	-8.405E-03	8.892E-04	0.973	-7.513E-02	8.891E-03	1.051E-04

J0161.DD PROCESSED ON 14 JAN 1995 11:46:27 AM
 RECORDS 420 TO 500 1410 SECS
 STARTING CHANNEL = 19. FINAL CHANNEL = 242. CONDENSED TO 16 CHANNELS FOR FILTER
 FINAL LEARNING RECORDS FOR INPUT VARIANCE
 9999 FOR POD 1, 9999 FOR POD 2

POD 1 LIBRARY

1 PREGA.B11161.380.1.0
 CONDENSED SPECTRUM 0.183E 04 653. 427. 176. 116. 83.1 20.8
 48.3 26.6 29.4 25.8 15.9 5.24
 2 DAT.R426P1.1480.04
 CONDENSED SPECTRUM 0.106E 04 323. 268. 61.3 74.7 47.5 22.6
 26.9 4.95 7.10 0.905 0.736 0.000
 3 DAT.C060P1.380.01
 CONDENSED SPECTRUM 0.255E 04 0.118E 04 766. 508. 577. 75.2
 2.24 0.737 0.421 1.66 1.84 1.16
 4 DAT.TH32P1.1000.01
 CONDENSED SPECTRUM 0.277E 04 0.194E 04 0.144E 04 875. 896. 230. 195.
 196. 110. 152. 184. 152. 1.13

GRANDSUMS: 0.381E 04 0.203E 04 0.679E 04 0.953E 04

POD 2 LIBRARY

1 PREGA.B12161.380.1.0
 CONDENSED SPECTRUM 0.213E 04 785. 527. 223. 138. 101. 125.
 61.0 49.3 34.2 27.2 26.6 8.37
 2 DAT.R426P2.1480.04
 CONDENSED SPECTRUM 0.113E 04 375. 378. 63.4 74.9 48.0 24.9
 25.1 28.8 7.14 1.05 0.736 0.000
 3 DAT.C060P2.380.01
 CONDENSED SPECTRUM 0.275E 04 0.124E 04 781. 523. 551. 100.
 2.29 0.658 0.211 1.95 1.18 1.05
 4 DAT.TH32P2.1000.01
 CONDENSED SPECTRUM 0.292E 04 0.196E 04 0.152E 04 889. 934. 236. 192.
 202. 117. 108. 153. 184. 1.24

GRANDSUMS 0.456E 04 0.216E 04 0.711E 04 0.985E 04

INITIAL INPUT VECTORS

POD 1 0.949 0.379E-01 0.949E-02 0.949E-02
 POD 2 0.948 0.379E-01 0.948E-02 0.948E-02

Appendix A: Sample Outputs for File J0161

1. Printer File lineprinter output (LU6) from KFILTR
2. Disk File normalized intensities (LU3) from KFILTR
3. Plot (Fig. A1) plot of normalized intensity estimates

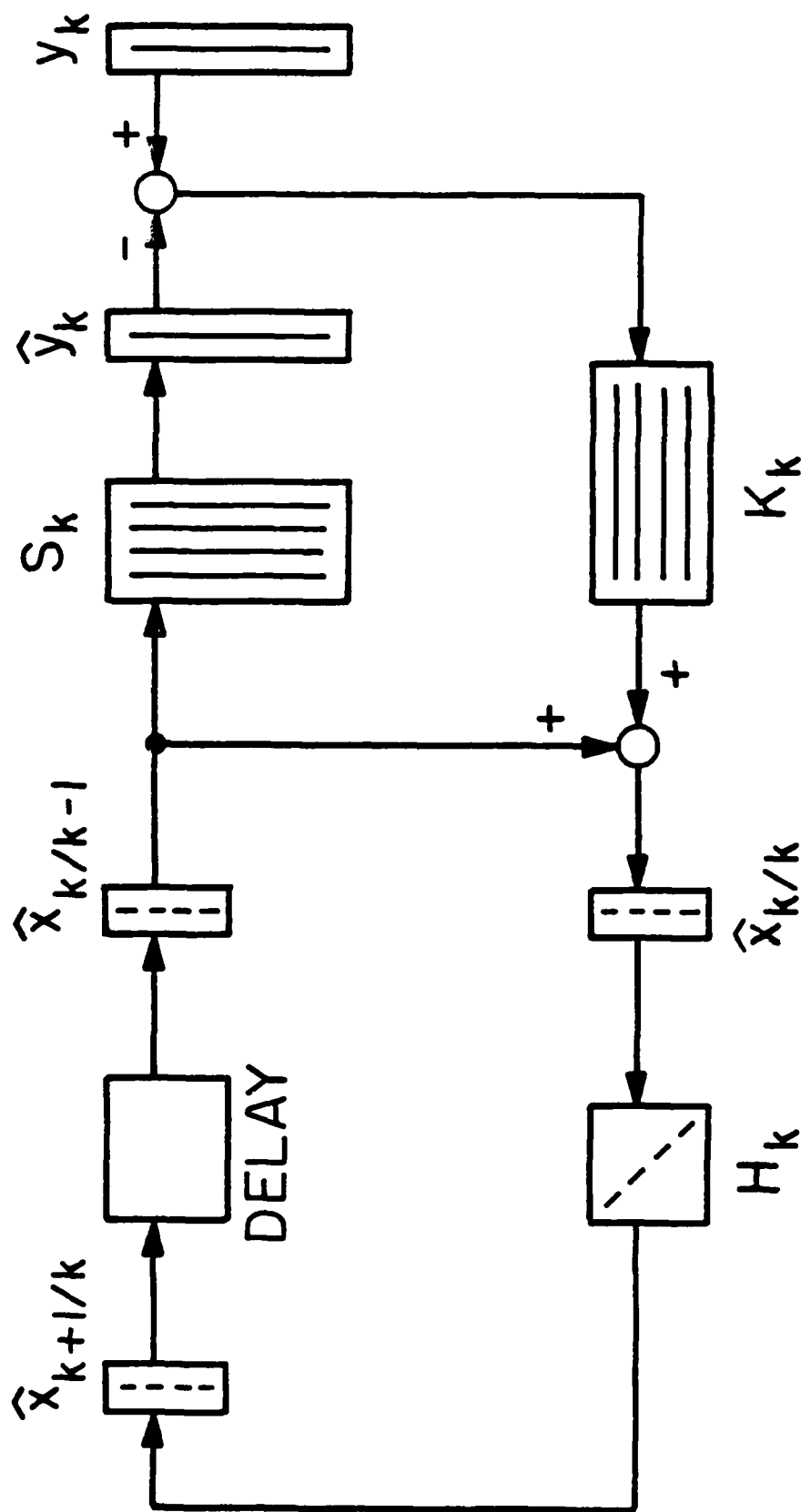


Figure 2 Diagram of the Kalman Filter for estimation of the source intensities x_k .

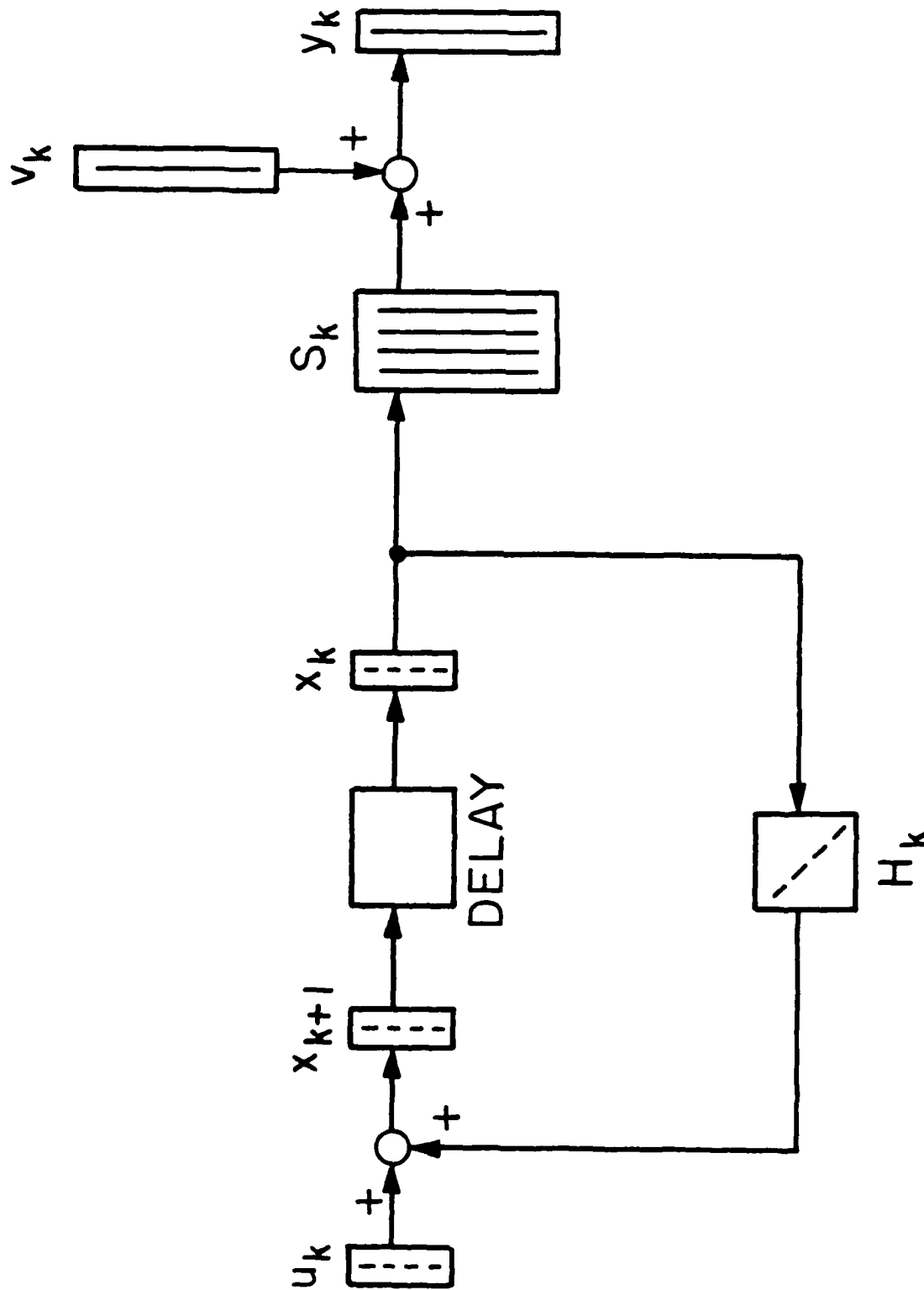


Figure 1 System model for the relationship between the source intensities x_k and the observed data y_k .

LIBIN inputs library spectra from logical unit LUF, number of
 channels M

Output from KOUT

Lineprinter LU6, lineprinter output, input information, header, and
 results

Disk File LU3, normalized intensities

Lineprinter Output Columns

REC record number

MODE data MODE switch

XSQ1, XSQ2 normalized chi-square (RSS) for pods 1 and 2

Library Elements relative intensities for pod 1 followed by pod 2.

Disk File Output Columns

REC, MODE same as above

Library Elements normalized intensities for pod 1 followed by pod 2.

References

1. G.W. Phillips and B.G. Glagola, "Program PAGSCN - Data Screen for the ND6620 Computer," NRL Memorandum Report 5269, March 1984.
2. G.W. Phillips and B.G. Glagola, "Program PREGA - Pivotal Regression Analysis of Gamma-Ray Spectra from NaI(Tl) Detectors for the ND6620 Computer, NRL Memorandum Report 5275, April 1984.
3. B.D.O. Anderson and J.B. Moore, "Optimal Filtering," Prentice-Hall, Englewood Cliffs, N.J. (1979).
4. J.L. LeMay and W.L. Brogan, "Kalman Filtering, Short Course Notes," Continuing Education Institute, Columbia, MD (1982).

L	dimension of Q on call to DDKALM
IER	error indicator from DDKALM
I,J	DO loop indices

Input to KLIN

Keyboard	LU5, on initial call only, free field format separated by commas
Data File	LU12, contains output from KLPREP

Input Variables from Keyboard

Record 1

File.Element	file name for output of KLPREP
LREC1, LREC2	optional start/stop records, defaults from KLPREP
IQ1, IQ2	optional cutoff records k0 for calculation of Q, x_m , and s, pods 1 and 2

Records 2a,...

NDREC(I)	optional bad record numbers to delete during calculation
----------	--

Input to KINIT

Keyboard	LU5, free field format separated by commas
Disk File	LU8, library spectra in Nuclear Data spectral format

Input Variables from Keyboard

<u>Records 3a, ...</u>	library spectra for Pod 1, Pod 2
------------------------	----------------------------------

File.Element	filename for spectrum
REALX(1)	counting time for spectrum
REALX(2)	initial intensity, $x_0/-1$
REALX(3)	optional fractional error, $(q_0)^{1/2}$, default = 1.0

Subroutines Called by KINIT

FREEFM	free field input subroutine (listing in Appendix B)
--------	---

Q(8)	input noise vector, diagonal elements of input noise matrix Q
V1(8,8), V2(8,8)	covariance matrices P for pods 1 and 2
Y1(16), Y2(16)	input data spectra for pods 1 and 2
R1(16), R2(16)	Poisson variances for pods 1 and 2
T1(16,16), T2(16,16)	work arrays for DDKALM
T3(16)	work vector for DDKALM
S1(16,8), S2(16,8)	library response matrices for pods 1 and 2
Q1(8), Q2(8)	vectors of input noise factor q_0 for pods 1 and 2
P1(8), P2(8)	vector of diagonal elements of covariance matrix P for pods 1 and 2
K	step index for Kalman filter, increments by one each for record
IN, IL, IS, IT	dimensions for arrays used in DDKALM
NR	record number
ICRT	logical unit number for CRT
LP	logical unit number for line printer
INIT	initially 0, set to 1 after call to KINIT
NID	MODE switch for input data
M	number of channels for condensed data
MS	starting channel for 256 channel spectra
MF	final channel for 256 channel spectra
N1, N2	number of library elements for pods 1 and 2
IDAY, IYR	Julian date, year
IHR, IMIN, ISEC	hour of day, minute, second

The cutoff record K1 is the minimum of K0, input by the operator, or the last record of the initial block of background (MODE = 1) data. There should be at least 30 records of MODE = 1 at the beginning of the data in order to get a good value for the sample mean $\bar{x}_m$ and standard deviation s.

The normalized intensities $n(x)$ are output to a disk file on LU3. The intensities $\hat{x}_{k/k}$ are listed in the printout on LU6 and are flagged by an asterisk whenever $n(x)$ or one of the three exponentially weighted averages exceeds its standard deviation by a factor THSIG currently set at 2.0 sigma. Sample outputs and plots of $\hat{x}_{k/k}$ are given for collection J0039 in Appendix A.

Language

The program is written in DEC RT11 FORTRAN and runs on the Nuclear Data ND6620 computer under the MIDAS operating system.

Inputs

Data input by KLIN. Library spectra input by KINIT.

Outputs

Data output by KOUT. Running status output to CRT (LU5).

Subroutines Called

KLIN	inputs data from disk file on LU12. (output of KLPREP). Listing in Appendix B.
KINIT	inputs library spectra from LU8, condenses to number of channels used for data from KLPREP. Listing in Appendix B.
BTIME	utility to return day and date in integer format (see Appendix C).
KSTEP	calculates the diagonal elements for the input noise matrix Q.
DDKALM	computes the updated estimates $\hat{x}_{k/k}$ and $P_{k/k}$.
KOUT	outputs the results.

Variables

X1(8), X2(8)	vector of estimated source intensities x for pods 1 and 2
H(8)	source transition vector, diagonal elements of transition matrix H

PROGRAM KFILTR

Operation

This is the main Kalman Filter program. It calls subroutines KLIN to read in the data from the file prepared by KLPREP, KINIT to read in the library spectra and condense them to the same number of channels as the data, KSREP to prepare for analyzing the next record, DDKALM to do the Kalman filter equations, and KOUT to output the results of each record. Sample output and plots are given in Appendix A for collection J0039.

The Kalman Filter subroutine DDKALM is a modification of a proprietary subroutine FTKALM, copyrighted by International Mathematical and Statistical Library, Inc. (IMSL). It in turn calls a number of proprietary IMSL routines. Listings are given in Appendix D for illustrative purposes only.

The lineprinter output gives for each Pod the normalized chisquare or residual sum of squares (RSS)

$$(1/ND) \sum_I (S_k \hat{x}_{k/k-1} - y_k)^2 / y_k$$

for each record k with ND degrees of freedom and Poisson data y_k , where the sum is over the channels I in the data spectrum. If this exceeds a threshold THRESH currently set at 3.0 sigma, the value is flagged by an asterisk. Source intensities are given for each library member and for each pod. Exponentially weighted averages are calculated for each library source for three different slopes,

$$a = 0.95, 0.80, \text{ and } 0.667.$$

A normalized intensity, $n(x)$, is also calculated, by

$$n(x) = (\hat{x}_{k/k} - x_m) / s$$

where the sample mean x_m and standard deviation s are given for $k \leq k_1$ by

$$\begin{aligned} x_m &= \bar{x}_k \\ s &= (v_k)^{1/2} \end{aligned}$$

and for $K > K_1$ by

$$\begin{aligned} x_m &= \bar{x}_{k_1} \\ s &= (v_{k_1})^{1/2}. \end{aligned}$$

526	1	5.957E-02	-0.807	8.65	-1.55	0.570	-1.95	2.07	0.582
527	1	-0.396	-0.792	8.34	0.147	-1.05	1.47	-3.14	1.14
528	1	0.522	-0.629	-0.477	-0.348	2.48	-1.49	-3.11	-0.707
529	1	1.07	-1.23	4.29	-0.429	1.88	-1.34	-1.30	-1.65
530	1	0.472	-0.858	1.24	-0.990	-0.138	-0.196	-1.92	0.844
531	1	0.541	-0.130	-6.40	1.89	-0.398	1.277E-02	0.352	1.18
532	1	-1.05	0.902	-4.44	1.05	-3.01	2.58	-0.404	1.85
533	1	-0.637	-0.380	2.71	0.791	-0.901	-0.150E-02	-2.35	2.48
534	1	0.246	7.725E-02	-3.11	-0.559	-3.22	0.210	-0.236	3.87
535	1	0.884	-0.343	-4.88	-2.42	-1.00	1.56	2.913E-02	0.640
536	1	2.33	-1.72	-5.31	-1.39	-2.30	2.15	1.31	1.16
537	1	2.34	-2.01	-4.28	-2.44	2.76	-4.34	2.06	0.967
538	1	3.86	-3.20	0.590	-3.11	1.71	-3.41	0.682	0.785
539	1	3.42	-3.70	-2.28	-1.26	-0.408	-0.777	1.36	-0.666
540	1	-0.395	-7.063E-02	2.81	0.378	1.45	-1.45	-0.933	-0.551
541	1	-0.660	-0.166	2.26	-0.181	3.10	-1.87	-2.02	-1.15
542	1	-5.025E-02	-0.966	6.67	-1.09	1.38	-0.869	0.989	-1.00
543	1	-1.92	1.41	-4.07	0.686	-0.260	0.938E-02	-1.85	0.689
544	1	1.27	-1.58	-2.40	4.496E-02	1.95	-3.76	3.00	0.680
545	1	0.765	-1.58	-2.82	0.477	1.91	-3.44	1.03	0.828
546	1	1.12	-1.52	1.66	-1.91	1.86	-4.58	2.53	0.778
547	1	0.753	-1.34	1.12	-0.612	0.217	-2.75	2.73	1.80
548	1	2.07	-2.15	4.59	-2.10	-1.57	2.602E-02	-0.856	2.08
549	1	-0.588	0.206	7.52	-2.46	-0.505	-0.882	-2.23	2.03
550	1	-1.81	1.68	1.33	-1.68	0.322	0.682	-0.875	-0.405
551	1	1.24	-0.962	1.49	-2.87	1.34	-1.35	1.67	-0.694
552	1	0.213	0.635	-5.60	-1.53	1.04	-1.11	1.32	-1.16
553	1	0.996	-0.625	-1.76	-1.56	0.700	-1.47	4.30	-1.01
554	1	0.927	-0.962	1.38	-1.21	0.478	-2.04	4.77	-0.551
555	1	2.12	-2.47	-1.933E-02	-0.977	1.35	-2.30	1.88	0.508
556	1	3.03	-2.21	1.74	-2.39	1.59	-1.91	-1.01	-0.341
557	1	-1.08	0.386E-03	7.20	-1.72	0.493	-0.687	0.237	-0.712
558	1	-0.411	-1.33	6.59	0.382	2.80	-1.93	-1.03	-0.990
559	1	0.484	-1.53	1.74	0.472	1.14	-0.524	-1.08	-4.053E-02
560	1	4.075E-03	-0.493	-5.940E-02	0.539	0.364	-0.673	-0.870	0.564

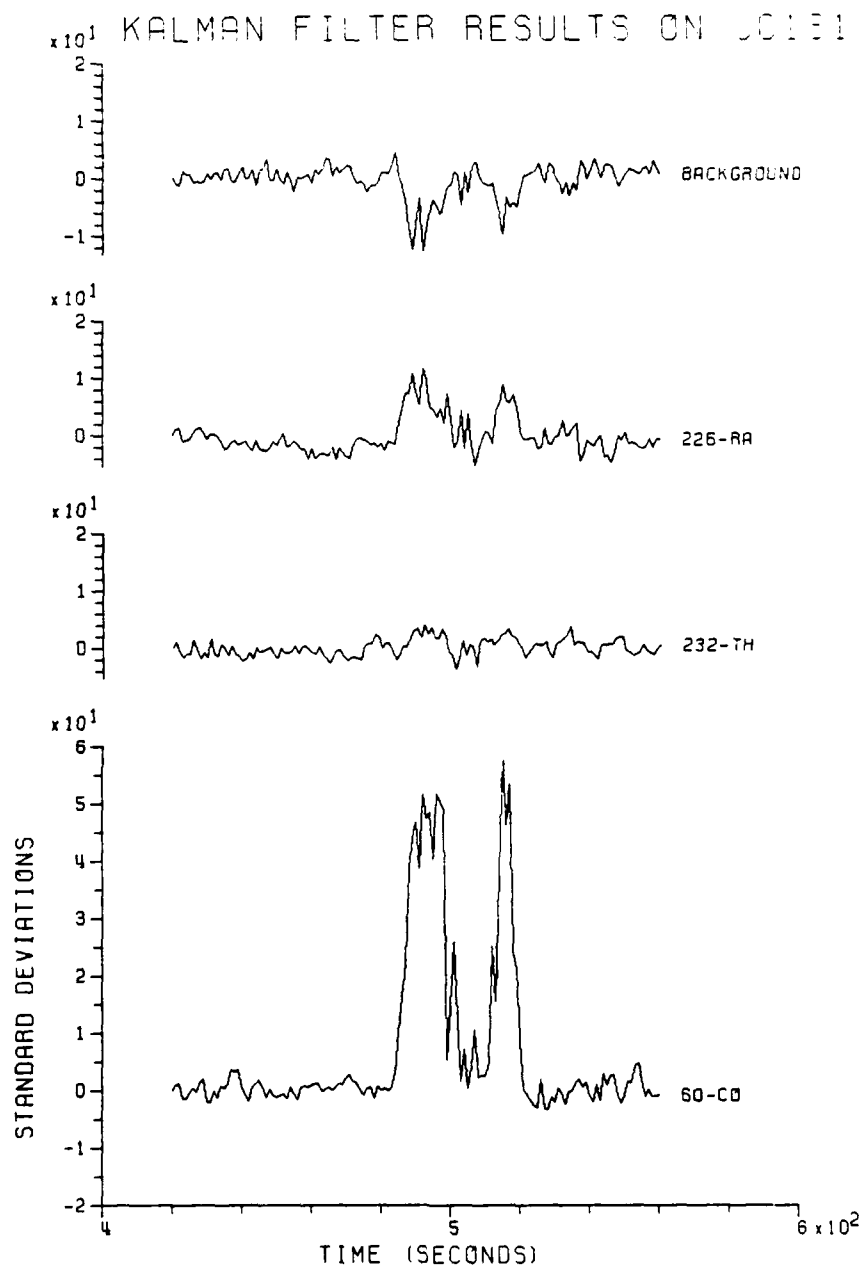


Figure A1 Plot of the normalized intensity estimates from the Kalman Filter for the data of J0161, containing a double peak due to ⁶⁰Co. The variation in the Background and ²²⁶Ra intensities are likely the effect of small corrections to the model due to differences between the shape of the ⁶⁰Co library standard and the source spectrum.

Appendix B: FORTRAN Listings, DEC RT-11

1. KLPREP
2. FREEFM
3. DATIN
4. DATOUT
5. KFILTR
6. KLIN
7. KINIT
8. LIBIN
9. KSTEP
10. KOUT

```

C      PROGRAM KLPREP
C      READS NIJAS DATA, CONDENSES, AND WRITES TO DATA
C      FILE FOR USE BY KALMAN FILTER PROGRAM
C      LAST MODIFIED BY G.W.PHILLIPS, APRIL 1982
C
0001      INTEGER DATA(1024),HEADER(256),NDREC(16)
0002      EQUIVALENCE (DATA,HEADER)
0003      INTEGER YES,PERIOD,BLANK
0004      REAL*8 ALPHA,FILDEF(5)
0005      COMMON/ARRAY/ DATA
0006      COMMON /DATA/IDAT(30)/FREE/INTEG(16),REALX(16),ALPHA(16)
0007      DATA FILDEF/0H0DEF 12 ,1H ,1H.,1H ,1H0/,ABLANK/0H /
0008      DATA YES/1HY/,IOUT/5/,IN/5/,LP/5/,NXREC/-1/,NFILE/12/
0009      DATA NCH/16/,INCH/16/,MS/3/,MF/255/,PERIOD/1H./,BLANK/1H /
0010      DATA NDEL/0/,NFSKIP/0/,NRSKIP/0/,IBLK/2H /

C
0011      10  CONTINUE
0012      WRITE(IOUT,20)
0013      20  FORMAT(' ENTER # OF FILES TO SKIP,# OF RECORDS TO SKIP,')
          1  4X,'FIRST RECORD, LAST RECORD')
0014      READ(IN,30) IDAT
0015      30  FORMAT(80A1)
0016      N=4
0017      M=1
0018      NA=1
0019      ITYPE=1
0020      CALL FREEFM(N,M,NA,ITYPE)      !PARSING SUBROUTINE
0021      IF(N.NE.4) GOTO 10
0023      IF(INTEG(1).NE.IBLK) NFSKIP=INTEG(1)
0025      IF(INTEG(2).NE.IBLK) NRSKIP=INTEG(2)
0027      NREC1=INTEG(3)
0028      NREC2=INTEG(4)

C
0029      NDEL=0
0030      WRITE(IOUT,32)
0031      32  FORMAT(' ENTER UP TO 16 BAD RECORDS TO DELETE')
0032      33  READ(IN,34) L,IDAT
0033      34  FORMAT(Q,80A1)
0034      IF(L.LE.0) GOTO 40
0036      N=16-NDEL
0037      IF(N.LE.0) GOTO 40
0039      M=1
0040      NA=1
0041      ITYPE=1
0042      CALL FREEFM(N,M,NA,ITYPE)
0043      IF(N.LE.0) GOTO 40
0045      DO 35 I=1,N
0046      35  NDREC(NDEL+I)=INTEG(1)
0047      NDEL=NDEL+N
0048      GOTO 33

```

```

C
0048 40 WRITE(IOUT,41)
0050 41 FORMAT(' ENTER STARTING, FINAL DATA CHANNELS'
1      ' (% OF OUTPUT CHANNELS FOR FILTER)')
0051 READ(IN,40) IDAT
0052 42 FORMAT(80A1)
0053      N=0
0054      N=1
0055      NA=1
0056      ITYPE=1
0057      CALL FREEFM(N,M,NA,ITYPE)
0058      IF(N.LT.2) GOTO 35
0059      NS=INTEG(1)
0061      NF=INTEG(2)
0062      IF(N.EQ.3) NCH=INTEG(3)
0064      IF(NCH.GT.INCH) NCH=INCH

C
C      INITILIZE OUTPUT FILE
C
0066 50 WRITE(IOUT,51)
0067 51 FORMAT(' FILENAME FOR DATA OUTPUT')
0068 READ(IN,52) LEN, IDAT
0069 52 FORMAT(Q,50A1)
0070 DO 54 I=1,LEN
0071 54 IF(IDAT(I).EQ.PERIOD) IDAT(I)=BLANK
0073      N=2
0074      NA=1
0075      M=1
0076      ITYPE=3
0077      CALL FREEFM(N,M,NA,ITYPE)
0078      FILDEF(2)=ALPHA(1)
0079      FILDEF(4)=ALPHA(2)
0080      IF(NX.LT.2) FILDEF(4)=BLANK
0082      CALL MCLI(FILDEF)
0083      NMFL=0
0084      CALL DANDC(NFILE,NERR,NDEV,NA8S,NVLS,NDTY,NRCZ,NBYL,NMFL)
0085      DEFINE FILE NFILE(4096,2,U,IV)

C
0086 WRITE(IOUT,56)
0087 56 FORMAT(' PRINT OUT CONDENSED SPECTRA, YES OR NO?',I)
0088 READ(IN,58) IANS
0089 58 FORMAT(A1)
0090      NPRT=0
0091      IF(IANS.EQ.YES) NPRT=1

C
C      INITIALIZE MAGTAPE
C
0093      MODE=1
0094      CALL DATIN(MODE,NR)
C

```

```

      C
      C      SKIP FILES ON TAPE
      C
0095  60      IF(NXREC.LT.0) GOTO 62
0097          IF(NRSKIP.EQ.0) GOTO 64
0099  52      MODE=5
0100          NR=NRSKIP
0101          CALL DATIN(MODE,NR)
0102          NXREC=0

      C
      C      READ HEADER RECORD
      C
0103  64      IF(NXREC.GT.0) GOTO 65
0105          MODE=4
0106          NR=NXREC
0107          CALL DATIN(MODE,NR)
0108  65      NTIME=10*(NREC2-NREC1+1)
0109          WRITE(LP,66) (HEADER(I), I=1,10),NREC1,NREC2,NTIME
0110  66      FORMAT(1H1,2X,7A2,3I6// RECORDS',15,' TO',15,
1          ',15,' SECONDS'//)
0111          WRITE(IOUT,67) (HEADER(I), I=1,10),NREC1,NREC2,NTIME
0112  67      FORMAT(1H1,2X,7A2,3I6// RECORDS',15,' TO',15,
1          ',15,' SECONDS'//)
0113          NXREC=1
0114          IV=1
0115          DO 68 I=1,10,2
0116  68      WRITE(NFILE'IV) HEADER(I),HEADER(I+1)
0117          WRITE(NFILE'IV) NREC1,NREC2
0118          WRITE(NFILE'IV) NTIME,NCH
0119          WRITE(NFILE'IV) MS,MF

      C
      C      SKIP RECORDS
      C
0120  69      IF(NRSKIP.EQ.0) GOTO 70
0122          MODE=6
0123          NR=NRSKIP
0124          CALL DATIN(MODE,NR)
0125          NXREC=NXREC+NRSKIP

```

```
CC
C      MAIN LOOP
C
0126 70  NRSTOP=9999
0127      KPRT=0
0128      DO 100 I=1,NRSTOP
0129      NDREC=I
0130      NR=NRREC
0131      CALL DATIN(MODE,NR)
0132      NRREC=NR+1
0133      IF(NR.LT.NREC1) GOTO 100
0135      IF(NDEL.LT.1) GOTO 70
0137      DO 74 J=1,NDEL
0138 74  IF(NR.EQ.NDREC(J)) DATA(15)=-999
0140 78  IF(KPRT.NE.0) GOTO 90
0142 80  CALL BTIME(IDAY,IYR,IHR,IMIN,ISEC)
0143      WRITE(IOUT,81) NR,IHR,IMIN,ISEC
0144 81  FORMAT(' BEGINNING AT RECORD',I4,
*      ' AT ',I2,':',I2,':',I2/)
0145      KPRT=1
0146 90  IF(NR.GE.NREC2) I=NRSTOP
0148      IF(NR.LT.9999) GOTO 92
0150      I=NRSTOP
0151      NR=NRREC-1
0152      GOTO 96
0153 92  CALL DATOUT(MS,MF,NCH,NFILE,NPRT,IY)
0154      IF(MOD(NR,100).NE.0) GOTO 96
0156      IF(NR.GE.9999) NR=NRREC-1
0158      CALL BTIME(IDAY,IYR,IHR,IMIN,ISEC)
0159      WRITE(IOUT,95) NR,IHR,IMIN,ISEC
0160 95  FORMAT(' PROCESSING COMPLETED THROUGH RECORD ',I54,
*      ' AT ',I2,':',I2,':',I2/)
0161 96  IF(I.LT.NRSTOP) GOTO 100
0163      CALL BTIME(IDAY,IYR,IHR,IMIN,ISEC)
0164      WRITE(IOUT,97) NR,IHR,IMIN,ISEC
0165 97  FORMAT(' THE END RECORD IS',I5,
*      ' AT ',I2,':',I2,':',I2/)
0166 100  CONTINUE
```

```

C
C   END OF MAIN LOOP
C
C   CLOSE OUTPUT FILE
0167   ENDFILE (HFILE)
0168   CALL CLOSEU(HFILE)
C
0169   IF (APT.EQ.1) WRITE(LP,120)
0171 120   FORMAT(1H1)
0172       WRITE(OUT,150)
0173 150   FORMAT(' DO YOU WISH TO CONTINUE?',/)
0174       READ(IN,200) NEWOP
0175 200   FORMAT(A1)
0176       IF(NEWOP.EQ.YES) GO TO 10
C
C   CLOSE MAGTAPE
C
0178 1000  MODE=10
0179       CALL DATIN(MODE,NR)
0180       WRITE(LP,1010)
0181 1010  FORMAT(1H1)
0182       END

```

NIJAS FORTRAN IV STORAGE MAP

NAME	ADDRESS	ATTRIBUTES
NOREC	000005	INTEGER*2 ARRAY (16)
FILDEF	000048	REAL*8 ARRAY (5)
YES	000103	INTEGER*2 VARIABLE
PERIOD	000145	INTEGER*2 VARIABLE
SLANK	000150	INTEGER*2 VARIABLE
COLANK	000155	REAL*4 VARIABLE
DOUF	000124	INTEGER*2 VARIABLE
IN	000126	INTEGER*2 VARIABLE
LP	000130	INTEGER*2 VARIABLE
NXREC	000132	INTEGER*2 VARIABLE
NFILE	000134	INTEGER*2 VARIABLE
NCH	000136	INTEGER*2 VARIABLE
INCH	000140	INTEGER*2 VARIABLE
MS	000142	INTEGER*2 VARIABLE
MF	000144	INTEGER*2 VARIABLE
NDEL	000152	INTEGER*2 VARIABLE
NSKIP	000154	INTEGER*2 VARIABLE
NSKIP	000156	INTEGER*2 VARIABLE
IBLK	000160	INTEGER*2 VARIABLE
N	000160	INTEGER*2 VARIABLE
M	000162	INTEGER*2 VARIABLE
NA	000164	INTEGER*2 VARIABLE
ITYPE	000166	INTEGER*2 VARIABLE
FPEEFM	000000	REAL*4 PROCEDURE
NREC1	000120	INTEGER*2 VARIABLE
NREC2	000122	INTEGER*2 VARIABLE
L	000124	INTEGER*2 VARIABLE
I	000126	INTEGER*2 VARIABLE
LEN	000130	INTEGER*2 VARIABLE
NX	000132	INTEGER*2 VARIABLE
MDLI	000000	INTEGER*2 PROCEDURE
MDLI	000134	INTEGER*2 VARIABLE
QANDC	000000	REAL*4 PROCEDURE
NERR	000136	INTEGER*2 VARIABLE
NDEV	000138	INTEGER*2 VARIABLE
NABS	000140	INTEGER*2 VARIABLE
NVLS	000142	INTEGER*2 VARIABLE
NDTY	000144	INTEGER*2 VARIABLE
NRCZ	000146	INTEGER*2 VARIABLE
NBYL	000148	INTEGER*2 VARIABLE
IV	000150	INTEGER*2 VARIABLE
IANS	000152	INTEGER*2 VARIABLE
NPRT	000154	INTEGER*2 VARIABLE
MODE	000156	INTEGER*2 VARIABLE
DATIN	000000	REAL*4 PROCEDURE
NR	000158	INTEGER*2 VARIABLE
NTIME	000160	INTEGER*2 VARIABLE
NRSTOP	000162	INTEGER*2 VARIABLE
NPRT	000164	INTEGER*2 VARIABLE
J	000166	INTEGER*2 VARIABLE
BTIME	000000	REAL*4 PROCEDURE
IDAY	000168	INTEGER*2 VARIABLE
YR	000170	INTEGER*2 VARIABLE
IHR	000172	INTEGER*2 VARIABLE
IMIN	000174	INTEGER*2 VARIABLE
ISEC	000176	INTEGER*2 VARIABLE
DATOUT	000000	REAL*4 PROCEDURE

MOD	000000	INTEGER*2 PROCEDURE
CLOSEU	000000	REAL*4 PROCEDURE
NEWOP	000160	INTEGER*2 VARIABLE

COMMON BLOCK /ARRAY/ LENGTH 004000

DATA	000000	INTEGER*2 ARRAY (1024)
HEADER	000000	INTEGER*2 ARRAY (256)

COMMON BLOCK /DATA/ LENGTH 000240

IDAT	000000	INTEGER*2 ARRAY (80)
------	--------	----------------------

COMMON BLOCK /FREE/ LENGTH 000340

INTEG	000000	INTEGER*2 ARRAY (16)
REALX	000040	REAL*4 ARRAY (16)
ALPHA	000140	REAL*8 ARRAY (16)

" 0000 LINKER

V02-A-1

LEAD MAP

5 JAN 1985 8:54:38 AM

[illegible]

	BLES	051474	BLT\$	051516	BNE\$	051514
	BPM\$	051510	NMI\$11	051495	NMI\$1M	051456
051522 000002	BAOTS	051522				
051524 000100	CBIS	051524	CBIS	051524	CFIS	051540
	CHC	051524	CHD	051524	CHZ	051540
051524 000100	CHAIT	051524				
051524 000100	CHCH	051524	CHISM	051540	CHICP	051536
051524 000100	CHCS	051524	CHVS	051524	CHPS	051532
	CHPS	051524				
051524 000100	CHIA	051524	CHLS	051540		
051524 000100	CHICPA	052002	CHICRM	051524	CHICSP	052100
	CHICPS	051524	CHICRS	051524		
052002 000100	CHISIA	052002	CHISIM	052022	CHISIS	052022
	CHISMA	052046	CHISMM	052042	CHISMS	052036
	CHISSA	052016	CHISSM	052012	CHISSS	052006
	CHISSA	052062	CHISSM	052056	CHISSS	052052
	CHISIA	052102	CHISIM	052074	CHISIS	052066
	CHISIS	052022	CHISSS	052006	CHLS	052022
052110 000046	CHISIP	052110	CHISMP	052120	CHISPI	052142
	CHISPM	052150	CHISPP	052130	CHISPS	052134
	CHISSP	052112				
052156 000044	CHISII	052176	CHISIM	052202	CHISIS	052172
	CHISMI	052212	CHISMM	052216	CHISMS	052206
	CHISSI	052162	CHISSM	052166	CHISSS	052156
052222 000034	DCISA	052252	DCISM	052244	DCISP	052250
	DCISS	052240	DCISA	052234	DCISM	052226
	DCISP	052232	DCISS	052222		
052256 000046	SUISIP	052256	SUISMP	052272	SUISPA	052316
	SUISPM	052310	SUISPP	052266	SUISPS	052302
	SUISSP	052260				
052324 000044	SUISIA	052344	SUISIM	052350	SUISIS	052340
	SUISMA	052360	SUISMM	052354	SUISMS	052354
	SUISSA	052330	SUISSM	052334	SUISSS	052324
052370 000040	ADISIP	052370	ADISMP	052404	ADISPA	052430
	ADISPM	052422	ADISPP	052400	ADISPS	052414
	ADISSP	052372				
052436 000044	ADISIA	052456	ADISIM	052462	ADISIS	052452
	ADISMA	052472	ADISMM	052476	ADISMS	052466
	ADISSA	052442	ADISSM	052446	ADISSS	052436
052502 000064	MOISIP	052502	MOISMP	052516	MOISPA	052542
	MOISPM	052534	MOISPP	052512	MOISPS	052526
	MOISSP	052504	MOISPP	052550	MOISIP	052556
052566 000042	MOISMA	052600	MOISMM	052566	MOISMP	052606
	MOISPA	052620	MOISPM	052614	MOISPP	052624
052630 000032	SAISIM	052630	SAISMM	052652	SAISSM	052632
	SVISIM	052640	SVISMM	052656	SVISSM	052642
052662 000032	TSDSI	052702	TSDSM	052676	TSDSP	052706
	TSDSS	052666	TSDSI	052702	TSDSM	052676
	TSDSP	052706	TSDSS	052672	TSDSI	052702
	TSDSM	052676	TSDSP	052706	TSDSS	052662
052714 000034	CPISIM	052736	CPISMM	052724	CPISPM	052720
	CPLISM	052714				
052750 000030	LGOS	052752	LGES	052762	LGTS	052750
	LLES	052750	LLTS	052774	LNES	052772

053000	000024	TSL\$1	053010	TSL\$M	053004	TSL\$P	053016
		TSL\$S	053000				
053024	000044	SAR\$SIM	053024	SAR\$MM	053040	SAR\$SM	053026
		SVF\$IN	053036	SVF\$IN	053064	SVF\$SN	053040
053070	000048	MOD\$M1	053110	MOD\$MM	053106	MOD\$MP	053074
		MOD\$P1	053114	MOD\$PM	053102	MOD\$PP	053070
053136	000020	MOD\$M1	053142	MOD\$MP	053136		
053156	000044	SAR\$SIM	053152	SAR\$MM	053212	SAR\$SM	053160
		SVF\$IN	053172	SVF\$IN	053216	SVF\$SN	053174
053216	000030	MOD\$M1	053222	MOD\$MP	053216	MOD\$PP	053222
		MOD\$P1	053214				
IOPKG	053252	001034	DOLC\$E	053352	DOPEN	053252	DREAD
			DRUN	054074	DWAIT	053660	DWRITE
			IAND	054026	IOR	054044	LSHIFT
							054056
MTAPEX	054306	002156	MTAPEF	054306			
MTULSU	056464	003762	MTULS	056464			
BTINEX	062446	000100	BTIME	052446			
MCLI\$	062546	000632	MCLI	062546			
CLOSU	063400	000032	CLOSEU	063400			
OANDCK	063432	000356	OCB	063746	ORC3	063720	OANDC
\$M.TVT	064010	000062	TVD\$	064024	TVF\$	064016	TVI\$
			TVL\$	064010	TVF\$	064040	TVQ\$
							064032
OT	064072	001510	\$ERR\$S	064602	\$PPERR	064444	\$OTI
STOP	065602	000112	EXIT	065626	FOO\$	065602	STP\$
RIO	065714	000600	DEF\$	066430	IRR\$	065714	IRW\$
			\$GETIN	066300			
GETREC	066514	000346	\$GETRE	066514	\$TTYIN	067014	
ENDFIL	067062	000042	EOF\$	067062			
CLOSS	067124	000550	\$CLOSE	067124			
OUTREC	067674	000414	\$PUTRE	067674			
\$M.FIO	070310	000216	\$FMTR	070310	\$FMTRW	070344	\$INITI
OPEN	070526	000610	\$OPEN	070526			
RWBLK	071336	000460	\$EOFIL	071746	\$GETBL	071560	\$PUTBL
ERRTB	072016	000100	\$ERRTB	072016			
ERRS	072116	000570	\$ERRS	072116			
\$M.LCV	074706	000106	LCI\$	074706	LCO\$	074754	

SEGMENT PARAMETER TABLE

SEG SIZE LIMIT
0 055014 075014

PROGRAM SIZE = 055014
DATA AREA SIZE = 000000
TRANSFER ADDRESS = 026054
STACK SIZE = 001000

```

001  SUBROUTINE FREEFM(N,M,NA,ITYPE)
      C  LAST REVISED AUGUST 1981 BY G.W.P.
      C
      C  GENERAL SUBROUTINE TO DECODE DATA READ IN FREE FIELD FORMAT
      C  DELIMITERS ARE EITHER A BLANK OR A COMMA
      C  THE ROUTINE ASSUMES THE DATA HAS BEEN READ INTO ARRAY IDATA WITH
      C  THE FORMAT (80A1)
      C  N IS THE NUMBER OF DATA ELEMENTS, MAXIMUM=16
      C  N IS RETURNED AS THE NUMBER OF DATA ELEMENTS FOUND
      C  M IS THE LOCATION IN THE ARRAY FOR STORING THE FIRST DATA ELEMENT
      C  M IS RETURNED AS THE LOCATION FOLLOWING THE NTH DATA ELEMENT
      C  NA IS THE BEGINNING COLUMN OF THE DATA
      C  NA IS RETURNED AS THE COLUMN FOLLOWING THE NTH DATA ELEMENT
      C  ITYPE IS THE TYPE OF DATA,
      C      1=INTEGER
      C      2=REAL
      C      3=ALPHANUMERIC
      C
0002  COMMON/DATA/IDATA(80)
0003  COMMON/FREE/INTEG(16),REALX(16),ALPHA(16)
0004  INTEGER SEMI,E,COMMA
0005  REAL*8 ALPHA,BLANK
0006  DIMENSION ITEMP(20),AFORM(2)
0007  DATA SEMI,E,IBLK1,COMMA,IBLK2,BLNK4,BLANK
      C  * / : , ' E ' , ' . ' , ' - ' , ' + ' , ' * ' , ' / ' /
      C
0008  L=M
0009  M=M+N-1
0010  DO 300 I=L,M
0011  IF(NA.GT.80) GO TO 400
      C
      C  LOOK FOR START OF CURRENT FIELD
      C
0013  DO 210 J=NA,80
0014  JQQ=J
0015  IF(IDATA(J).NE.IBLK1) GO TO 215
0017  210 CONTINUE
0018  NA=81
0019  GO TO 400
0020  215 IF(IDATA(JQQ).NE.COMMA) GO TO 220
0022  NA=JQQ+1
0023  GO TO 290

```

```
C
C      LOOK FOR END OF CURRENT FIELD
C
024  220  IL=J00
025        ILQ = IL
026  221  DO 230 J=ILQ,90
027        J00=J
028        IF(IDATA(J).EQ.IBLK1) GO TO 235
029        IF(IDATA(J).NE.COMMA) GO TO 230
032        IR=J-1
033        NA=J+1
034        GO TO 250
035  230  CONTINUE
036        IR=80
037        NA=81
038        GO TO 250

C
C      CHECK FOR EXPONENT
C
039  235  IF((ITYPE.NE.2).OR.(IDATA(J00-1).NE.E)) GO TO 236
041        ILQ = J00 + 1
042        GO TO 221
043  236  IR=J00 - 1
044        IJ=J00+1

C
C      SET NA TO START OF NEXT FIELD
C
045      DO 240 J=IJ,80
046        IF(IDATA(J).EQ.IBLK1) GO TO 240
048        NA=J
049        IF(IDATA(J).EQ.COMMA) NA=NA+1
051        GO TO 250
052  240  CONTINUE
053        NA=81

C
C      ENCODE DATA IN CURRENT FIELD
C
054  250  NI=IR-IL+1
055        IF(NI.LT.1) GO TO 290
057        ENCODE(NI,255,ITEMP) (IDATA(J),J=IL,IR)
058  255  FORMAT(83A1)
```

```

C
07 100 CALL KLIN(K,Y1,Y2,NR,NID,M,MS,MF)
08 IF(NR.LT.0) GOTO 300
09 IF(INIT.NE.0) GOTO 102
10 CALL KINIT(X1,X2,Y1,Y2,H,V1,V2,P1,P2,Q1,Q2,
11 S1,S2,N1,N2,M,MS,MF)
12 CALL BTIME(IDAY,IYR,IHR,IMIN,ISEC)
13 WRITE(ICRT,10)NR,IHR,IMIN,ISEC
14 101 FORMAT('BEGINNING AT RECORD ',I4,
15 ' AT ',I2,' ',I2,' ',I2,/)
16 INIT=1
17 102 IF(NID.EQ.1.OR.NID.EQ.2.OR.NID.EQ.3) GOTO 105
18 IF(NID.GT.10.AND.NID.LT.100) GOTO 105
21 WRITE(LP,104)NR,NID
22 104 FORMAT(I5,I3)
23 GOTO 100

C
24 105 L=N1
D WRITE(ICRT,110)K,NR
D110 FORMAT(' POD 1. CALLING KSTEP. STEP',I4,'. RECORD',I4,/)
25 CALL KSTEP(K,X1,Y1,S1,Q,Q1,R1,N1,M,NR)
D WRITE(ICRT,120)
D120 FORMAT(' CALLING DDKALM'/)
26 CALL DDKALM(K,X1,H,Y1,S1,Q,R1,V1,
1 IN,IS,IL,N1,M,L,T1,T2,IT,T3,IER)
27 DO 200 J=1,M
28 200 R1(J)=T3(J)
29 DO 210 I=1,N1
30 210 P1(I)=V1(I,I)
31 L=N2
D WRITE(ICRT,212)K,NR
D212 FORMAT(' POD 2. CALLING KSTEP. STEP',I4,'. RECORD',I4,/)
32 CALL KSTEP(K,X2,Y2,S2,Q,Q2,R2,N2,M,NR)
D WRITE(ICRT,214)
D214 FORMAT(' CALLING DDKALM'/)
33 CALL DDKALM(K,X2,H,Y2,S2,Q,R2,V2,
1 IN,IS,IL,N2,M,L,T1,T2,IT,T3,IER)
34 DO 220 J=1,M
35 220 R2(J)=T3(J)
36 DO 230 I=1,N2
37 230 P2(I)=V2(I,I)
38 K=K+1
D WRITE(ICRT,240)
D240 FORMAT(' CALLING KOUT')
39 CALL KOUT(K,NR,NID,X1,X2,P1,P2,Y1,Y2,S1,S2,
1 Q1,Q2,R1,R2,N1,N2,M,MS,MF)
40 IF(MOD(NR,50).NE.0) GOTO 250
42 CALL BTIME(IDAY,IYR,IHR,IMIN,ISEC)
43 WRITE(ICRT,250) NR,IHR,IMIN,ISEC
44 250 FORMAT(' PROCESSING COMPLETED THROUGH RECORD',I4,
45 ' AT ',I2,' ',I2,' ',I2,/)
45 250 IF(NP.GE.0) GOTO 100

```

```
C      PROGRAM KFILTR
C      RUNS KALMAN FILTER FOR NIAGARA DATA
C      LAST MODIFIED BY G.W.PHILLIPS, JANUARY 1985
C
001      REAL*4 N1(8),X2(6),H(8),Q(8),V1(8,8),V2(8,8)
002      REAL*4 Y1(16),Y2(16),R1(16),R2(16)
003      REAL*4 T1(16,16),T2(16,16),T3(16)
004      REAL*4 S1(16,8),S2(16,8),O1(8),O2(8),P1(8),P2(8)
005      DATA K/0/,IM/8/,IL/8/,IS/16/,IT/16/,NR/0/,ICRT/5/,LP/6/
006      DATA INIT/0/
C
```

MIDAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
Y1	000018	REAL*4 PARAMETER ARRAY (16)
Y2	000030	REAL*4 PARAMETER ARRAY (16)
Y	000034	INTEGER*2 ARRAY (16)
DATE	000074	INTEGER*2 ARRAY (12)
TIME	000124	INTEGER*2 ARRAY (16)
FILNEF	000164	REAL*8 ARRAY (5)
K	000014	INTEGER*2 PARAMETER VARIABLE
NR	000032	INTEGER*2 PARAMETER VARIABLE
NID	000024	INTEGER*2 PARAMETER VARIABLE
NCH	000026	INTEGER*2 PARAMETER VARIABLE
MS	000030	INTEGER*2 PARAMETER VARIABLE
MF	000032	INTEGER*2 PARAMETER VARIABLE
PERIOD	000240	INTEGER*2 VARIABLE
BLANK	000242	INTEGER*2 VARIABLE
IY	000234	INTEGER*2 VARIABLE
LP	000236	INTEGER*2 VARIABLE
IN	000244	INTEGER*2 VARIABLE
IOUT	000246	INTEGER*2 VARIABLE
NFILE	000250	INTEGER*2 VARIABLE
IBLK	000252	INTEGER*2 VARIABLE
INIT	000254	INTEGER*2 VARIABLE
LREC1	000256	INTEGER*2 VARIABLE
LREC2	000260	INTEGER*2 VARIABLE
NDEL	000262	INTEGER*2 VARIABLE
LEN	001174	INTEGER*2 VARIABLE
I	001176	INTEGER*2 VARIABLE
N	001200	INTEGER*2 VARIABLE
NA	001202	INTEGER*2 VARIABLE
N	001204	INTEGER*2 VARIABLE
ITYPE	001206	INTEGER*2 VARIABLE
FREEFM	000000	REAL*4 PROCEDURE
NCL1	000000	INTEGER*2 PROCEDURE
NPEC1	001210	INTEGER*2 VARIABLE
NREC2	001212	INTEGER*2 VARIABLE
NTIME	001214	INTEGER*2 VARIABLE
MAX0	000000	INTEGER*2 PROCEDURE
MIN0	000000	INTEGER*2 PROCEDURE
ATIME	000000	REAL*4 PROCEDURE
LR	001216	INTEGER*2 VARIABLE

COMMON BLOCK /DATA/ LENGTH 000240

IDATA 000000 INTEGER*2 ARRAY (80)

COMMON BLOCK /FREE/ LENGTH 000340

INTEG 000000 INTEGER*2 ARRAY (16)

REALX 000040 REAL*4 ARRAY (16)

ALPHA 000140 REAL*8 ARRAY (16)

COMMON BLOCK /HDR/ LENGTH 000024

HEADER 000000 INTEGER*2 ARRAY (10)

COMMON BLOCK /IO/ LENGTH 000004

IQ1 000000 INTEGER*2 VARIABLE

IQ2 000002 INTEGER*2 VARIABLE

```
      C
      C      READ OBSERVED SPECTRA FOR PODS 1 AND 2
      C
0084  200  IF(LR.LT.NREC2) GOTO 210
0085      NR=-NR
0087      RETURN
      C
0088  210  READ(NFILE*IV,END=300)NR,NID
0089      IF(NR.NE.LR+1) NID=0
0091      IF(NR.GT.0) GOTO 220
0093      NR=0
0094      IF(NID.GT.0) NID=-NID
      C
0096  220  LR=NR
0097      DO 230 I=1,NCH,2
0098  230  READ(NFILE*IV,END=300) IY(I),IY(I+1)
0099      DO 240 I=1,NCH
0100  240  Y1(I)=IY(I)
0101      DO 260 I=1,NCH,2
0102  260  READ(NFILE*IV,END=300) IY(I),IY(I+1)
0103      DO 270 I=1,NCH
0104  270  Y2(I)=IY(I)
0105      IF(NR.LT.NREC1) GOTO 210
0107      IF(NDEL.LE.0) GOTO 280
0109      DO 274 I=1,NDEL
0110  274  IF(NR.EQ.NDREC(I)) NID=-999
      C
0112  280  CONTINUE
      D      WRITE(LP,282)NR,NID,(Y1(I),I=1,NCH)
D282  FORMAT(1H0,'OBSERVED VECTORS. RECORD',I4,' MODE',I4/
      D      1X,'POD 1',8G13.3/(7X,8G13.3))
      D      WRITE(LP,284)(Y2(I),I=1,NCH)
D284  FORMAT(1X,'POD 2',8G13.3/(7X,8G13.3))
0113      RETURN
      C
0114  300  WRITE(1OUT,310) NR
0115  310  FORMAT(' PREMATURE END OF DATA AT RECORD',I4/)
0116      NR=-NR
0117      RETURN
0118      END
```

```
C
0043      NDEL=0
0044      WRITE(IOUT,132)
0045 132    FORMAT(' ENTER UP TO 16 BAD RECORDS TO DELETE'//)
0046 133    READ(IN,134) LEN, IDATA
0047 134    FORMAT(0.00A1)
0048      IF(LEN.LE.0) GOTO 140
0050      N=16-NDEL
0051      IF(N.LE.0) GOTO 140
0053      M=1
0054      NA=1
0055      ITYPE=1
0056      CALL FREEFM(N,M,NA,ITYPE)
0057      IF(N.LE.0) GOTO 140
0059      DO 135 I=1,N
0060 135    NDREC(NDEL+I)=INTEG(I)
0061      NDEL=NDEL+N
0062      GOTO 133

C
0063 140    CALL MCLI(FILDEF)
0064      IV=1
0065      DEFINE FILE NFILE(4096,2,U,IV)
0066      DO 142 I=1,10,2
0067 142    READ(NFILE'IV,END=300) HEADER(I),HEADER(I+1)
0068      READ(NFILE'IV,END=300) NREC1,NREC2
0069      READ(NFILE'IV,END=300) NTIME,NCH
0070      READ(NFILE'IV,END=300) MS,MF
0071      NREC1=MAX0(NREC1,LREC1)
0072      NREC2=MIN0(NREC2,LREC2)
0073      CALL ATIME(KDATE)
0074      WRITE(LP,144) (HEADER(I),I=1,7),KDATE
0075 144    FORMAT(1H1,2X,7A2,3X,'PROCESSED ON ',12A2)
0076      WRITE(LP,145) NREC1,NREC2,NTIME,MS,MF,NCH
0077 145    FORMAT(3X,'RECORDS',15,' TO',15,
1         2X,15,' SECS'//
2         ' STARTING CHANNEL=',15,
3         ' FINAL CHANNEL=',15,
4         ' CONDENSED TO',13,' CHANNELS FOR FILTER')
0078      WRITE(LP,148) IQ1,IQ2
0079 148    FORMAT(' FINAL LEARNING RECORDS FOR INPUT VARIANCE'//
1         4X,15,' FOR POD 1.',15,' FOR POD 2'//)
0080      WRITE(IOUT,150) (HEADER(I),I=1,7),NREC1,NREC2
0081 150    FORMAT(1X,7A2// ' RECORDS',15,' TO',15//)
0082      INIT=1
0083      LR=NREC1-1
```

```
0001      SUBROUTINE KLIN(K,Y1,Y2,NR,NID,NCH,MS,MF)
      C      READS IN DATA FOR KALMAN FILTER
      C      LAST MODIFIED BY G.W.PHILLIPS, JANUARY 1985
0002      DIMENSION Y1(16),Y2(16),IY(16)
0003      INTEGER PERIOD,BLANK,HEADER(10),KDATE(12),NDREC(16)
0004      REAL *8 ALPHA,FILDEF(5)
0005      COMMON /DATA/IDATA(80)/FREE/INTEG(16),PEALX(16),ALPHA(16)
0006      COMMON HDR-HEADER
0007      COMMON IQ/IQ1,IQ2
0008      DATA FILDEF/8H0DEF 12 ,1H ,1H.,1H ,1H0/,IY/0/,LP/6/
0009      DATA PERIOD/1H./,BLANK/1H /,IN/5/,IOUT/5/,NFILE/12/
0010      DATA IQ1/9999/,IQ2/9999/,IBLK/2H /,INIT/0/
0011      DATA LREC1/0/,LREC2/9999/,NDEL/0/
      C
      C      INITIALIZE FILE AND READ IN HEADER
      C
0012      IF(INIT.NE.0) GOTO 200
0014 100  WRITE(IOUT,110)
0015 110  FORMAT(' FILENAME FOR OBSERVED SPECTRA (,START,STOP RECORDS,')
      1 4X,'LAST LEARNING RECORDS FOR VARIANCE: POD 1,POD 2)')
0016      READ(IN,120)LEN,IDATA
0017 120  FORMAT(Q,80A1)
0018      IF(LEN.LT.1) GOTO 100
0020      DO 130 I=1,LEN
0021 130  IF(IDATA(I).EQ.PERIOD) IDATA(I)=BLANK
0023      N=2
0024      NA=1
0025      M=1
0026      ITYPE=3
0027      CALL FREEFM(N,M,NA,ITYPE)
0028      FILDEF(2)=ALPHA(1)
0029      FILDEF(4)=ALPHA(2)
0030      IF(N.LT.2) FILDEF(4)=BLANK
0032      N=4
0033      M=1
0034      CALL FREEFM(N,M,NA,1)
0035      IF(INTEG(1).NE.IBLK) LREC1=INTEG(1)
0037      IF(INTEG(2).NE.IBLK) LREC2=INTEG(2)
0039      IF(INTEG(3).NE.IBLK) IQ1=INTEG(3)
0041      IF(INTEG(4).NE.IBLK) IQ2=INTEG(4)
```

MIDAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
I1	000030	INTEGER*2 ARRAY (32)
I2	000130	INTEGER*2 ARRAY (32)
MS	000014	INTEGER*2 PARAMETER VARIABLE
MF	000016	INTEGER*2 PARAMETER VARIABLE
NCH	000020	INTEGER*2 PARAMETER VARIABLE
NFILE	000022	INTEGER*2 PARAMETER VARIABLE
NPRT	000024	INTEGER*2 PARAMETER VARIABLE
IV	000026	INTEGER*2 PARAMETER VARIABLE
NREC	000262	INTEGER*2 VARIABLE
NID	000264	INTEGER*2 VARIABLE
MR	000266	INTEGER*2 VARIABLE
ML	000270	INTEGER*2 VARIABLE
IJ	000272	INTEGER*2 VARIABLE
J	000274	INTEGER*2 VARIABLE
MJ	000276	INTEGER*2 VARIABLE
L1	000300	INTEGER*2 VARIABLE
L2	000302	INTEGER*2 VARIABLE
I	000304	INTEGER*2 VARIABLE
M	000306	INTEGER*2 VARIABLE

COMMON BLOCK /ARRAY/ LENGTH 004000

DATA 000000 INTEGER*2 ARRAY (1024)

```
0001      SUBROUTINE DATOUT(MS,MF,NCH,NFILE,NPRT,IV)
          C      CONDENSES NIAGARA DATA AND WRITES OUT FOR KALMAN FILTER
          C      WRITTEN BY G. PHILLIPS, JUNE 1981
          C
0002      INTEGER DATA(1024),I1(32),I2(32)
0003      COMMON/ARRAY/ DATA
          C
          C      EXTRACT OBSERVED SPECTRA FOR PODS 1 AND 2
          C
0004      NREC=DATA(3)
0005      NID=DATA(15)
0006      IF(MS.LT.3) MS=3
0008      IF(MF.GT.255) MF=255
0010      IF(NCH.GT.32) NCH=32
          C
          C      CONDENSE SPECTRA TO NCH CHANNELS
          C
0012      MR=MF-MS+1
0013      ML=(MR+NCH-1)/NCH
0014      IJ=0
0015      DO 160 J=MS,MF,ML
0016      MJ=J+ML-1
0017      IF(MJ.GT.MF) MJ=MF
0019      L1=0
0020      L2=0
0021      DO 140 I=J,MJ
0022      L1=L1+DATA(I+512)
0023      L2=L2+DATA(I+768)
0024 140    CONTINUE
0025      IJ=IJ+1
0026      I1(IJ)=L1
0027      I2(IJ)=L2
0028 160    CONTINUE
0029      M=IJ
          C
          C      WRITE TO OUTPUT FILE
          C
0030 200    WRITE(NFILE*IV)NREC,NID
0031      DO 220 I=1,NCH,2
0032 220    WRITE(NFILE*IV) I1(I),I1(I+1)
0033      DO 240 I=1,NCH,2
0034 240    WRITE(NFILE*IV) I2(I),I2(I+1)
0035      IF(NPRT.EQ.0) RETURN
          C
0037      WRITE(6,260) NREC,NID,(I1(I),I=1,NCH)
0038 260    FORMAT(1X,2I4,8I8/(9X,8I8))
0039      WRITE(6,280) (I2(I),I=1,NCH)
0040 280    FORMAT(9X,8I8)
0041      RETURN
0042      END
```

```

      C
      C
0007  600  WRITE(IOUT,610)MODE
0008  610  FORMAT(1H0,'ILLEGAL MODE',I4,' CALLED TO TAPE READ SUBROUTINE')
0009      GOTO 1000
      C
      C      END OF FILE
      C
0009  800  NR=NR-1
0091      WRITE(IOUT,810) NR
0092  810  FORMAT(1H0,'END OF FILE AFTER RECORD',I4)
0093      NR=9999
0094      RETURN
      C
      C      WRITE ERROR CODES
      C
0095  900  CONTINUE
0096  910  IF(IB.NE.2) GOTO 920
0098      WRITE(IOUT,911)
0099  911  FORMAT(1H0,'TRANSPORT NOT OPEN')
0100      GOTO 1000
0101  920  IF(IB.NE.20) GOTO 1000
0103      WRITE(IOUT,921)
0104  921  FORMAT(1H0,'TRANSPORT OFF LINE')
0105      GOTO 1000
      C
0106 1000  STOP
0107      END

```

MIDAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
------	--------	------------

MODE	000014	INTEGER*2 PARAMETER VARIABLE
NR	000016	INTEGER*2 PARAMETER VARIABLE
IOUT	000020	INTEGER*2 VARIABLE
INIT	000022	INTEGER*2 VARIABLE
MTAPEF	000000	INTEGER*2 PROCEDURE
IB	000634	INTEGER*2 VARIABLE
I	000636	INTEGER*2 VARIABLE
ILSHFT	000000	INTEGER*2 PROCEDURE
IBCD	000000	INTEGER*2 PROCEDURE

COMMON BLOCK	/ARRAY/	LENGTH
--------------	---------	--------

DATA	000000	INTEGER*2 ARRAY (1024)
HEADER	000000	INTEGER*2 ARRAY (256)

```
C
0060 400  IF(MODE.NE.7) GOTO 500
C
C    READ DATA RECORD
C
D    WRITE(IOUT,401) MODE,NR
D401  FORMAT(' DATIN,MODE=',I3,' RECORD=',I4)
0062  CALL MTAPEF(7,I3,0,2048,DATA)
0063  IF(IB.EQ.1) GOTO 440
0065  WRITE(IOUT,410) I3,NR
0066 410  FORMAT(1H0,'ERROR',I4,' IN DATA RECORD ',I4)
0067  IF(IB.EQ.2.OR.IB.EQ.20) GOTO 900
0069  IF(IB.EQ.9.OR.IB.EQ.22) GOTO 800
0071 440  CONTINUE
D    WRITE(IOUT,441) NR
D441  FORMAT(' RECORD',I4,' READ SUCCESSFULLY')
0072  DO 460 I=1,4
0073 460  DATA(I)=ILSHFT(DATA(I),8)
0074  DO 470 I=11,16
0075 470  DATA(I)=IBCD(DATA(I))          ! BCD TO DECIMAL CONVERSION
0076  DO 480 I=17,1024
0077 480  DATA(I)=ILSHFT(DATA(I),8)
0078  NR=DATA(3)
0079  RETURN
C
C    CLOSE TRANSPORT
C
0080 500  IF(MODE.NE.10) GOTO 600
0082  CALL MTAPEF(16,IB)
0083  IF(IB.NE.1) WRITE(IOUT,510) IB
0085 510  FORMAT(' ERROR',I3,' IN MAGTAPE CLOSE')
0086  RETURN
```

```
C
C
0025 100  IF(MODE.NE.4) GOTO 200
C
C      READ HEADER RECORD
C
0027      IF(NR.GT.0) RETURN
0029      CALL MTAPEF(7,IB,0,512,HEADER)
0030      NR=1
0031      IF(IB.EQ.1) GOTO 120
0033      WRITE(IOUT,110)IB
0034 110   FORMAT(1H0,'ERROR',I4,' IN HEADER RECORD READ')
0035      GOTO 900
0036 120   DO 140 I=1,3
0037 140   HEADER(I+7)=ILSHFT(HEADER(I+7),8)
0038      RETURN
C
0039 200   IF(MODE.NE.5) GOTO 300
C
C      SKIP FILES ON TAPE
0041      IF(NR.EQ.0) RETURN
0043      CALL MTAPEF(5,IB,NR)
0044      NR=0
0045      IF(IB.EQ.1) RETURN
0047      WRITE(IOUT,210)IB
0048 210   FORMAT(1H0,'ERROR',I4,' ON FILE SKIP')
0049      GOTO 900
C
0050 300   IF(MODE.NE.6) GOTO 400
C
C      SKIP RECORDS ON TAPE
C
0052      IF(NR.EQ.0) RETURN
0054      CALL MTAPEF(6,IB,NR)
0055      IF(IB.EQ.1) RETURN
0057      WRITE(IOUT,310)IB
0058 310   FORMAT(1H0,'ERROR',I4,' IN RECORD SKIP')
0059      GOTO 900
```

```
0001      SUBROUTINE DATIN(MODE,NR)
          C      NIAGARA TAPE READER
          C      LAST MODIFIED BY G.W.PHILLIPS, APRIL 1982
          C
0002      INTEGER DATA(1024),HEADER(256)
0003      EQUIVALENCE (DATA,HEADER)
0004      COMMON/ARRAY/ DATA
0005      DATA IOUT/5/,INIT/0/
          C
0006      IF(MODE.NE.1) GOTO 100
          C
          C      OPEN AND INITIALIZE MAGTAPE
          C
0008      IF(INIT.EQ.1) RETURN
0010      CALL MTAPEF(15,IB,1)
0011      IF(IB.NE.1) WRITE(IOUT,10)IB
0013 10     FORMAT(' ERROR',I3,' IN MAGTAPE OPEN'/)
0014      IF(IB.NE.20) GOTO 30
0016      WRITE(IOUT,20)
0017 20     FORMAT (' TRANSPORT OFF LINE'/)
0018      STOP
          C
0019 30     CALL MTAPEF(1,IB)
0020      IF(IB.NE.1) WRITE(IOUT,35)IB
0022 35     FORMAT(' ERROR',I3,' IN MAGTAPE INITIALIZE'/)
0023      INIT=1
0024      RETURN
```

MIDAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
ITEMP	000024	INTEGER*2 ARRAY (20)
AFORM	000074	REAL*4 ARRAY (2)
N	000014	INTEGER*2 PARAMETER VARIABLE
M	000016	INTEGER*2 PARAMETER VARIABLE
NA	000020	INTEGER*2 PARAMETER VARIABLE
ITYPE	000022	INTEGER*2 PARAMETER VARIABLE
SEMI	000104	INTEGER*2 VARIABLE
E	000106	INTEGER*2 VARIABLE
COMMA	000112	INTEGER*2 VARIABLE
BLANK	000122	REAL*8 VARIABLE
IBLK1	000110	INTEGER*2 VARIABLE
IBLK2	000114	INTEGER*2 VARIABLE
BLNK4	000116	REAL*4 VARIABLE
L	000210	INTEGER*2 VARIABLE
I	000212	INTEGER*2 VARIABLE
J	000214	INTEGER*2 VARIABLE
JQQ	000216	INTEGER*2 VARIABLE
IL	000220	INTEGER*2 VARIABLE
ILQ	000222	INTEGER*2 VARIABLE
IR	000224	INTEGER*2 VARIABLE
IJ	000226	INTEGER*2 VARIABLE
NI	000230	INTEGER*2 VARIABLE

COMMON BLOCK /DATA/ LENGTH 000240

IDATA 000000 INTEGER*2 ARRAY (80)

COMMON BLOCK /FREE/ LENGTH 000340

INTEG 000000 INTEGER*2 ARRAY (16)

REALX 000040 REAL*4 ARRAY (16)

ALPHA 000140 REAL*8 ARRAY (16)

```
C
C
0059      GO TO (250,270,280), ITYPE
C
C      DECODE INTEGER DATA
C
0060      260 ENCODE(8,255,AFORM) NI
0061      265 FORMAT('I12,')
0062      DECODE(NI,AFORM,ITEMP) INTEG(I)
0063      GO TO 300
C
C      DECODE REAL DATA
C
0064      270 ENCODE(8,275,AFORM) NI
0065      275 FORMAT('E12,')
0066      DECODE(NI,AFORM,ITEMP) REALX(I)
0067      GO TO 300
C
C      DECODE ALPHANUMERIC DATA
C
0068      280 IF(NI.GT.8) NI=8
0070      DO 287 J=1,NI
0071      IF(ITEMP(J).EQ.SEMI) ITEMP(J)=COMMA
0073      287 CONTINUE
0074      ENCODE(8,288,AFORM) NI
0075      288 FORMAT('A',I1,')
0076      DECODE(8,AFORM,ITEMP) ALPHA(I)
0077      GO TO 300
C
C      BLANK OUT REMAINING DATA
C
0078      290 INTEG(I)=IBLK2
0079      REALX(I)=BLNK4
0080      ALPHA(I)=BLANK
0081      300 CONTINUE
0082      M=M+1
0083      RETURN
C
C      BLANK INPUT, BLANK OUT ALL DATA
C
0084      400 DO 410 J=1,M
0085      INTEG(J)=IBLK2
0086      REALX(J)=BLNK4
0087      ALPHA(J)=BLANK
0088      410 CONTINUE
0089      M=1
0090      N=M-L
0091      RETURN
0092      END
```

```
      C  
      C  
0047 300  NR=-NR  
0048      CALL BTIME(IDAY,IYR,IHR,IMIN,ISEC)  
0049      WRITE(ICRT,302) NR,IHR,IMIN,ISEC  
0050 302  FORMAT(' ENDING AT RECORD',I4,  
      *      ' AT ',I2,' ',I2,' ',I2/)  
0051      WRITE(LP,310)  
0052 310  FORMAT(1H1)  
0053      STOP  
0054      END
```

MIDAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
X1	000006	REAL*4 ARRAY (8)
X2	000046	REAL*4 ARRAY (8)
H	000106	REAL*4 ARRAY (8)
Q	000146	REAL*4 ARRAY (8)
V1	000206	REAL*4 ARRAY (8,3) VECTORED
V2	000606	REAL*4 ARRAY (8,8) VECTORED
Y1	001206	REAL*4 ARRAY (16)
Y2	001306	REAL*4 ARRAY (16)
R1	001406	REAL*4 ARRAY (16)
R2	001506	REAL*4 ARRAY (16)
T1	001606	REAL*4 ARRAY (16,16) VECTORED
T2	003606	REAL*4 ARRAY (16,16) VECTORED
T3	005606	REAL*4 ARRAY (16)
S1	005706	REAL*4 ARRAY (16,8) VECTORED
S2	006706	REAL*4 ARRAY (16,8) VECTORED
Q1	007706	REAL*4 ARRAY (8)
Q2	007746	REAL*4 ARRAY (8)
P1	010006	REAL*4 ARRAY (8)
P2	010046	REAL*4 ARRAY (8)
K	010106	INTEGER*2 VARIABLE
IN	010110	INTEGER*2 VARIABLE
IL	010112	INTEGER*2 VARIABLE
IS	010114	INTEGER*2 VARIABLE
IT	010116	INTEGER*2 VARIABLE
NR	010120	INTEGER*2 VARIABLE
ICRT	010122	INTEGER*2 VARIABLE
LP	010124	INTEGER*2 VARIABLE
INIT	010126	INTEGER*2 VARIABLE
KLIN	000000	INTEGER*2 PROCEDURE
NID	010442	INTEGER*2 VARIABLE
M	010444	INTEGER*2 VARIABLE
MS	010446	INTEGER*2 VARIABLE
MF	010450	INTEGER*2 VARIABLE
KINIT	000000	INTEGER*2 PROCEDURE
N1	010452	INTEGER*2 VARIABLE
N2	010454	INTEGER*2 VARIABLE
BTIME	000000	REAL*4 PROCEDURE
IDAY	010456	INTEGER*2 VARIABLE
IYR	010460	INTEGER*2 VARIABLE
IHR	010462	INTEGER*2 VARIABLE
IMIN	010464	INTEGER*2 VARIABLE
ISEC	010466	INTEGER*2 VARIABLE
L	010470	INTEGER*2 VARIABLE
KSTEP	000000	INTEGER*2 PROCEDURE
DDKALM	000000	REAL*4 PROCEDURE
IER	010472	INTEGER*2 VARIABLE
J	010474	INTEGER*2 VARIABLE
I	010476	INTEGER*2 VARIABLE
KOUT	000000	INTEGER*2 PROCEDURE
MOD	000000	INTEGER*2 PROCEDURE

SECTION	ADDR	SIZE	ENTRY	ADDR	ENTRY	ADDR	ENTRY	ADDR
. ABS.	000000	000000	\$LRECL	000210	\$NLCHN	000006	\$USRSW	000000
			\$RF1B3	000000				
. ABS.	000000	000000	\$TRACE	004737				
. ABS.	000000	000000	\$V004A	000001				
. \$\$\$\$.&	020000	000000	.\$\$\$\$.	020000				
	030000	012522						
	032522	005754	KINIT	032522				
	040476	003762	KLIN	040476				
	044460	002364	DDKALM	044460				
	047044	007542	KOUT	047044				
	056606	001326	KSTEP	056606				
	060134	000372	LIBIN	060134				
	060526	000452	VMULFF	060526				
	061200	000436	VMULFP	061200				
	061636	000362	LEQTF	061636				
	062220	003554	LUDATF	062220				
	065774	000760	LUELME	065774				
	066754	002026	UERTST	066754				
	071002	002106	FREEFM	071002				
	073110	000146	XFF\$	073110	\$PWRR	073110		
	073256	000060	NMI\$II	073256	NMI\$MI	073304	NMI\$PI	073312
			NPI\$II	073322	NPI\$MI	073326	NPI\$PI	073332
	073336	000064	NMI\$IP	073336	NMI\$MP	073370	NMI\$PP	073376
			NPI\$IP	073406	NPI\$MP	073412	NPI\$PP	073416
	073422	000044	NMI\$IP	073422	NPI\$II	073442	NPI\$IM	073454
			NPI\$IP	073436				
	073466	000142	SQRT	073466				
	073630	000350	EXP	073630				
	074200	000360	ALOG	074204	ALOG10	074200		
	074560	000262	XFI\$	074560	\$PWRI	074560		
	075042	000040	MOD	075042				
	075102	000110	TAD\$	075132	TAF\$	075140	TAI\$	075102
			TAL\$	075110	TAP\$	075124	TAQ\$	075116
	075212	000020	STK\$F	075222	STK\$I	075216	STK\$L	075212
	075232	000056	\$OTIS	075232				
	075310	000210	ISN\$	075310	LSN\$	075330	\$ISNTR	075314
			\$LSNTR	075334				
	075520	000034	END\$	075520	ERR\$	075536		
	075554	000046	EOL\$	075554				
	075622	000066	DEO\$	075622				
	075710	000044	RET\$	075724	RET\$F	075714	RET\$I	075722
			RET\$L	075710				
	075754	001106	IBR\$	075762	IBW\$	075754	\$IBW	075766
	077062	000072	ENC\$	077062				
	077154	000020	IFR\$	077154	IFW\$	077166		
	077174	001562	\$FIO	077662				
	100756	002344	DCO\$	102402	ECO\$	102374	FCO\$	102370
			GCO\$	102362	ICI\$	100764	ICO\$	102136
			OCI\$	100756	OCO\$	102130	RCI\$	101160
			\$GET	101144				
	103322	000036	CFD\$	103322	\$DR	103322		

103360	000110	\$DUMPL	103360				
103470	000036	\$GETFI	103470				
103526	000042	DII\$IS	103540	DII\$MS	103534	DII\$PS	103526
		DII\$SS	103542	\$DVI	103542		
103570	000040	MUI\$IS	103602	MUI\$MS	103576	MUI\$PS	103570
		MUI\$SS	103604	\$MLI	103604		
103630	000130	CIC\$	103630	CID\$	103630	CIF\$	103640
		CIL\$	103752	CLC\$	103630	CLD\$	103630
		CLF\$	103640	CLIG	103756	\$DI	103630
		ERI	103640				
103760	000120	\$FCHNL	103760				
104100	000044	EEQ\$	104120	BGE\$	104130	BGT\$	104126
		BLE\$	104116	BLT\$	104140	BNE\$	104136
		BRA\$	104132	NMI\$II	104110	NMI\$IM	104100
104144	000002	\$AOTS	104144				
104146	000100	CCI\$	104146	CDI\$	104146	CFI\$	104162
		\$IC	104146	\$ID	104146	\$IR	104162
104246	000016	\$FCALL	104246				
104264	000012	\$WAIT	104264				
104276	000044	JMC\$	104304	JMI\$M	104300	JMI\$P	104276
104342	000030	AND\$	104346	EQV\$	104354	IOR\$	104342
		XOR\$	104356				
104372	000036	CAI\$	104372	CAL\$	104400		
104430	000024	MAX0	104430				
104454	000024	MIN0	104454				
104500	000016	ABS	104500				
104516	000016	MOI\$RA	104530	MOI\$RM	104522	MOI\$RP	104526
		MOI\$RS	104516	MOL\$RS	104516		
104534	000102	MOI\$IA	104560	MOI\$IM	104554	MOI\$IS	104550
		MOI\$MA	104574	MOI\$MM	104570	MOI\$MS	104564
		MOI\$SA	104544	MOI\$SM	104540	MOI\$SS	104534
		MOI\$OA	104610	MOI\$OM	104604	MOI\$OS	104600
		MOI\$IA	104630	MOI\$IM	104622	MOI\$IS	104614
		MOL\$IS	104550	MOL\$SS	104534	REL\$	104550
104636	000046	CMI\$IP	104636	CMI\$MP	104646	CMI\$PI	104670
		CMI\$PM	104676	CMI\$PP	104656	CMI\$PS	104662
		CMI\$SP	104640				
104704	000044	CMI\$II	104724	CMI\$IM	104730	CMI\$IS	104720
		CMI\$MI	104740	CMI\$MM	104744	CMI\$MS	104734
		CMI\$SI	104710	CMI\$SM	104714	CMI\$SS	104704
104750	000016	NGI\$A	104762	NGI\$M	104754	NGI\$P	104760
		NGI\$S	104750				
104766	000034	DCI\$A	105016	DCI\$M	105010	DCI\$P	105014
		DCI\$S	105004	ICI\$A	105000	ICI\$M	104772
		ICI\$P	104776	ICI\$S	104766		
105022	000046	SUI\$IP	105022	SUI\$MP	105036	SUI\$PA	105062
		SUI\$PM	105054	SUI\$PP	105032	SUI\$PS	105046
		SUI\$SP	105024				
105070	000044	SUI\$IA	105110	SUI\$IM	105114	SUI\$IS	105104
		SUI\$MA	105124	SUI\$MM	105130	SUI\$MS	105120
		SUI\$SA	105074	SUI\$SM	105100	SUI\$SS	105070
105134	000046	ADI\$IP	105134	ADI\$MP	105150	ADI\$PA	105174
		ADI\$PM	105166	ADI\$PP	105144	ADI\$PS	105160
		ADI\$SP	105136				

105202	000044	ADI\$IA	105222	ADI\$IM	105226	ADI\$IS	105216
		ADI\$MA	105236	ADI\$MM	105242	ADI\$MS	105232
		ADI\$SA	105206	ADI\$SM	105212	ADI\$SS	105202
105246	000064	MOI\$IP	105246	MOI\$MP	105262	MOI\$PA	105306
		MOI\$PM	105300	MOI\$PP	105256	MOI\$PS	105272
		MOI\$SP	105250	MOI\$0P	105314	MOI\$1P	105322
105332	000174	CMF\$II	105406	CMF\$IM	105470	CMF\$IP	105440
		CMF\$IS	105352	CMF\$MI	105376	CMF\$MM	105460
		CMF\$MP	105430	CMF\$MS	105336	CMF\$PI	105372
		CMF\$PM	105454	CMF\$PP	105424	CMF\$PS	105332
		CMF\$SI	105412	CMF\$SM	105474	CMF\$SP	105444
		CMF\$SS	105360	\$CMR	105360		
105336	000026	MOF\$RA	105544	MOF\$RM	105534	MOF\$RP	105550
		MOF\$RS	105526				
105554	000012	MOF\$SS	105554				
105566	000014	MOF\$IS	105566	MOF\$0S	105574		
105602	000016	MOF\$MS	105602	MOF\$PS	105614		
105620	000014	MOF\$SA	105620				
105634	000010	MOF\$IA	105634				
105644	000014	MOF\$SM	105644	MOF\$SP	105654		
105660	000020	MOF\$IM	105660	MOF\$IP	105672		
105700	000020	MOF\$0A	105710	MOF\$0M	105700	MOF\$0P	105714
105720	000042	MOF\$MA	105732	MOF\$MM	105720	MOF\$MP	105740
		MOF\$PA	105752	MOF\$PM	105746	MOF\$PP	105756
105762	000054	SAI\$PM	105762	SAI\$PP	106010	SVI\$PM	105774
		SVI\$PP	106022				
106036	000032	SAI\$IP	106036	SAI\$MP	106060	SAI\$SP	106040
		SVI\$IP	106046	SVI\$MP	106064	SVI\$SP	106050
106070	000032	SAI\$IM	106070	SAI\$MM	106112	SAI\$SM	106072
		SVI\$IM	106100	SVI\$MM	106116	SVI\$SM	106102
106122	000032	TSD\$I	106142	TSD\$M	106136	TSD\$P	106146
		TSD\$S	106126	TSF\$I	106142	TSF\$M	106136
		TSF\$P	106146	TSF\$S	106132	TSI\$I	106142
		TSI\$M	106136	TSI\$P	106146	TSI\$S	106122
106154	000034	CPD\$SM	106176	CPF\$SM	106164	CPI\$SM	106160
		CPL\$SM	106154				
106210	000030	LEQ\$	106212	LGE\$	106222	LGT\$	106220
		LLE\$	106210	LLT\$	106234	LNE\$	106232
106240	000024	TSL\$I	106250	TSL\$M	106244	TSL\$P	106256
		TSL\$S	106240				
106264	000044	SAF\$IP	106264	SAF\$MP	106320	SAF\$SP	106266
		SVF\$IP	106276	SVF\$MP	106324	SVF\$SP	106300
106330	000044	SAF\$IM	106330	SAF\$MM	106364	SAF\$SM	106332
		SVF\$IM	106342	SVF\$MM	106370	SVF\$SM	106344
106374	000036	NGD\$A	106426	NGD\$M	106406	NGD\$P	106422
		NGD\$S	106374	NGF\$A	106426	NGF\$M	106406
		NGF\$P	106422	NGF\$S	106374		
106432	000030	MOD\$MS	106442	MOD\$PS	106436	MOD\$SS	106432
		MOD\$VS	106444				
106462	000046	MOD\$MA	106512	MOD\$MM	106500	MOD\$MP	106466
		MOD\$PA	106506	MOD\$PM	106474	MOD\$PP	106462
106530	000020	MOD\$SM	106534	MOD\$SP	106530		
106550	000044	SAD\$IM	106550	SAD\$MM	106604	SAD\$SM	106552
		SVD\$IM	106564	SVD\$MM	106610	SVD\$SM	106566

106614	000016	CCF\$	106614	CDF\$	106614	\$RC	106614
		\$RD	106614				
106632	001166	ADD\$IS	106702	ADD\$MS	106662	ADD\$PS	106656
		ADD\$SS	106730	SUD\$IS	106714	SUD\$MS	106636
		SUD\$PS	106632	SUD\$SS	106724	\$ADD	106730
		\$SBD	106724				
110020	000074	SAD\$PM	110020	SAD\$PP	110056	SVD\$PM	110036
		SVD\$PP	110074				
ARRAY &	110114	004000	ARRAY	110114			
ARRAY &	110114	004000					
HEAD &	114114	000200	HEAD	114114			
HEAD &	114114	000200					
DATA &	114314	000240	DATA	114314			
DATA &	114314	000240					
DATA &	114314	000240					
FREE &	114554	000340	FREE	114554			
FREE &	114554	000340					
FREE &	114554	000340					
HDR &	115114	000024	HDR	115114			
HDR &	115114	000024					
IQ &	115140	000004	IQ	115140			
IQ &	115140	000004					
IQ &	115140	000004					
ATIME\$	115144	000100	ATIME	115144			
BTIME\$	115244	000100	BTIME	115244			
MCLI\$	115344	000632	MCLI	115344			
\$M.TVT	116176	000062	TV\$	116212	TVF\$	116204	TVI\$
			TVL\$	116176	TVP\$	116226	TVQ\$
OT	116260	001510	\$ERR\$	116770	\$FPERR	116632	\$OTI
STOP	117770	000112	EXIT	120014	FOO\$	117770	STP\$
RIO	120102	000600	DEF\$	120616	IRR\$	120102	IRW\$
			\$GETIN	120466			
GETREC	120702	000346	\$GETRE	120702	\$TTYIN	121202	
ENDFIL	121250	000042	EOF\$	121250			
CLOSS	121312	000550	\$CLOSE	121312			
\$M.RMM	122062	000170	AMAX1	122104	AMIN1	122062	MAX1
			MIN1	122066			
OUTREC	122252	000414	\$PUTRE	122252			
\$M.FIO	122666	000216	\$FMTDR	122666	\$FMTDW	122722	\$INITI
OPEN	123104	000610	\$OPEN	123104			
FADD	123714	000062	ADF\$IS	123714	ADF\$MS	123726	ADF\$PS
			ADF\$SS	123720	SUF\$IS	123742	SUF\$MS
			SUF\$PS	123750	SUF\$SS	123746	\$ADR
			\$SBR	123746			
FDIV	123776	000034	DIF\$IS	124014	DIF\$MS	124002	DIF\$PS
			DIF\$SS	124020	\$DVR	124020	
FMUL	124032	000034	MUF\$IS	124050	MUF\$MS	124036	MUF\$PS
			MUF\$SS	124054	\$MLR	124054	
RWBLK	124066	000460	\$EOFIL	124476	\$GETBL	124310	\$PUTBL
ERRTB	124546	000100	\$ERRTB	124546			
ERRS	124646	002570	\$ERRS	124646			
\$M.LCV	127436	000106	LCI\$	127436	LCO\$	127504	
ADDM	127544	000116	ADF\$IM	127544	ADF\$MM	127556	ADF\$PM
			ADF\$SM	127566	SUF\$IM	127602	SUF\$MM

		SUF\$PM 127632	SUF\$SM 127610		
ADDP	127662 000114	ADF\$IP 127662	ADF\$MP 127674	ADF\$PP 127570	
		ADF\$SP 127704	SUF\$IP 127720	SUF\$MP 127754	
		SUF\$PP 127750	SUF\$SP 127726		
ADDA	127776 000150	ADF\$IA 127776	ADF\$MA 130030	ADF\$PA 130024	
		ADF\$SA 130006	SUF\$IA 130054	SUF\$MA 130110	
		SUF\$PA 130104	SUF\$SA 130062	\$FPA 130050	
		\$FPA 130130			

SEGMENT PARAMETER TABLE

SEG	SIZE	LIMIT
0	110146	130146

PROGRAM SIZE = 110146
DATA AREA SIZE = 000000
TRANSFER ADDRESS = 020000
STACK SIZE = 001000

```
0001      SUBROUTINE KINIT(X1,X2,Y1,Y2,H,V1,V2,P1,P2,Q1,Q2,  
1        S1,S2,N1,N2,M,MS,MF)  
      C      INITIALIZES KALMAN FILTER FOR NIAGARA DATA  
      C      LAST MODIFIED BY G.W.PHILLIPS. MARCH 1982  
      C  
0002      COMMON/ARRAY/ARRAY(512)  
0003      COMMON/HEAD/HDR1(8),HDR2(8)  
0004      REAL*8 HDR1,HDR2  
0005      REAL*4 X1(8),X2(8),Y1(16),Y2(16),S1(16,8),S2(16,8)  
0006      REAL*4 P1(8),P2(8),Q1(8),Q2(8),H(8),V1(8,8),V2(8,8)  
0007      INTEGER CRT,DATA,PERIOD,BLANK,COMMA  
0008      COMMON/DATA/DATA(80)/FREE/INTEG(16),REALX(16),ALPHA(16)  
0009      REAL*8 FILDEF(5),ALPHA,ABLANK  
0010      DATA FILDEF/8HDEF 8 ,1H ,1H.,1H ,1H0/,ABLANK/8H /  
0011      DATA CRT/5/,LP/6/,IN/5/,IOUT/6/,PERIOD/1H./,BLANK/1H /  
0012      DATA COMMA/1H./,LUF/8/
```

```
C
0013 100 N1=0
0014      WRITE(CRT,110)
0015 110  FORMAT(1H0,'INPUT POD 1 LIBRARY SPECTRA'//
      1      1X,'FILENAME,COUNT TIME,REL. INTENS.(,FRACT. ERROR)'//
      2      1X,'TERMINATE WITH (CR)')
0016      WRITE(IOUT,112)
0017 112  FORMAT(1H0,'POD 1 LIBRARY')
0018 120  N1=N1+1
0019      WRITE(CRT,130)N1
0020 130  FORMAT(1X,I3,5)
0021      READ(CRT,140)LEN,DATA
0022 140  FORMAT(Q,80A1)
0023      IF(LEN.LT.1) GOTO 190
0025      WRITE(IOUT,142) N1,(DATA(I),I=1,LEN)
0026 142  FORMAT(1H0,I5,2X,80A1)
0027      DO 150 I=1,LEN
0028      IF(DATA(I).EQ.COMMA) GOTO 152
0030      IF(DATA(I).EQ.PERIOD) DATA(I)=BLANK
0032 150  CONTINUE
0033 152  NX=2
0034      MX=1
0035      NA=1
0036      CALL FREEFM(NX,MX,NA,3)
0037      FILDEF(2)=ALPHA(1)
0038      FILDEF(4)=ALPHA(2)
0039      HDR1(N1)=ALPHA(2)
0040      IF(NX.EQ.2) GOTO 153
0042      HDR1(N1)=ALPHA(1)
0043      FILDEF(4)=ABLANK
0044 153  CALL MCLI(FILDEF)
0045      NX=3
0046      MX=1
0047      CALL FREEFM(NX,MX,NA,2)
0048      FX=10./REALX(1)
0049      X1(N1)=REALX(2)
0050      P1(N1)=1
0051      IF(NX.EQ.3) P1(N1)=REALX(3)**2
```

C

```
C
0053      MI=256
0054      CALL LISIN(LUF,MI)
D
0055      WRITE(LP,155)N1,(ARRAY(I),I=1,MI)
D155     FORMAT(' POD 1, LIBRARY MEMBER',I3/('(10G13.3))
0056      IF(MI.GE.MF) GOTO 150
0057      WRITE(CRT,156) N1,MI
0058      156     FORMAT(1X,'LIBRARY MEMBER',I4,' SHORT, LENGTH=',I4/)
0059      160     ML=(MF-MS+M)/M
D
0060      WRITE(CRT,162) M,MS,MF,ML
D162     FORMAT(1H0,'M,MS,MF,ML=',4(I4,' '))
0061      Q1(N1)=0.
0062      IJ=0
0063      DO 180 J=MS,MF,ML
0064      MJ=J+ML-1
0065      IF(MJ.GT.MF) MJ=MF
0066      SUM=0.
0067      DO 170 I=J,MJ
0068      170     SUM=SUM+ARRAY(I)
0069      SUM=FX*SUM
0070      IJ=IJ+1
0071      S1(IJ,N1)=SUM
0072      Q1(N1)=Q1(N1)+SUM
0073      180     CONTINUE
0074      WRITE(IOUT,182) (S1(I,N1),I=1,M)
0075      182     FORMAT(1H0,'CONDENSED SPECTRUM',8G13.3/((19X,8G13.3))
0076      GOTO 120
C
0077      190     N1=N1-1
0078      WRITE(IOUT,192) (Q1(I),I=1,N1)
0079      192     FORMAT(1H0,'GRANDSUMS:',8G13.3/((10X,8G13.3))
C
```

```
C
0080 200 N2=0
0081      WRITE(CRT,210)
0082 210  FORMAT(1H0,'INPUT POD 2 LIBRARY SPECTRA'//
1      1X,'FILENAME,COUNT TIME,REL. INTENS.(,FRACT. ERROR)'//
2      1X,'TERMINATE WITH (CR)')
0083      WRITE(IOUT,212)
0084 212  FORMAT(1H0,'POD 2 LIBRARY')
0085 220  N3=N2+1
0086      WRITE(CRT,230)N2
0087 230  FORMAT(1X,I3,$)
0088      READ(CRT,240)LEN,DATA
0089 240  FORMAT(0,80A1)
0090      IF(LEN.LT.1) GOTO 290
0092      WRITE(IOUT,242) N2,(DATA(I),I=1,LEN)
0093 242  FORMAT(1H0,15,2X,80A1)
0094      DO 250 I=1,LEN
0095          IF(DATA(I).EQ.COMMA) GOTO 252
0097          IF(DATA(I).EQ.PERIOD) DATA(I)=BLANK
0099 250  CONTINUE
0100 252  NX=2
0101      MX=1
0102      NA=1
0103      CALL FREEFM(NX,MX,NA,3)
0104      FILDEF(2)=ALPHA(1)
0105      FILDEF(4)=ALPHA(2)
0106      HDR2(N2)=ALPHA(2)
0107      IF(NX.EQ.2) GOTO 253
0109      HDR2(N2)=ALPHA(1)
0110      FILDEF(4)=ABLANK
0111 253  CALL MCL1(FILDEF)
0112      NX=3
0113      MX=1
0114      CALL FREEFM(NX,MX,NA,2)
0115      FX=10./REALX(1)
0116      X2(N2)=REALX(2)
0117      P2(N2)=1.
0118      IF(NX.EQ.3) P2(N2)=REALX(3)**2
```

```

C
C
0120      MI=256
0121      CALL LISIN(LUF,MI)
D        WRITE(LP,255)N2,(ARRAY(I),I=1,MI)
0255      FORMAT('POD 2. LIBRARY MEMBER',I3/(10G13.3))
0122      IF(MI.GE.MF) GOTO 260
0124      WRITE(CRT,256) N2,MI
0125      256  FORMAT(1X,'LIBRARY MEMBER',I4,' SHORT. LENGTH=',I4/)
0126      260  NL=(MF-MS+M)/M
D        WRITE(CRT,262) M,MS,MF,ML
0252      FORMAT(1H0,'M,MS,MF,ML=',4(I4,' '))
0127      Q2(N2)=0.
0128      IJ=0
0129      DO 280 J=MS,MF,ML
0130      MJ=J+ML-1
0131      IF(MJ.GT.MF) MJ=MF
0133      SUM=0.
0134      DO 270 I=J,MJ
0135      270  SUM=SUM+ARRAY(I)
0136      SUM=FX*SUM
0137      IJ=IJ+1
0138      S2(IJ,N2)=SUM
0139      Q2(N2)=Q2(N2)+SUM
0140      280  CONTINUE
0141      WRITE(IOUT,282) (S2(I,N2),I=1,M)
0142      282  FORMAT(1H0,'CONDENSED SPECTRUM',8G13.3/(19X,8G13.3))
0143      GOTO 220
C
0144      290  N2=N2-1
0145      WRITE(IOUT,292) (Q2(I),I=1,N2)
0146      292  FORMAT(1H0,'GRANDSUMS',8G13.3/(10X,8G13.3))
C

```

```
C
47 000 SUMY1=0.
48 SUMY2=0.
49 DO 320 J=1,M
50 SUMY1=SUMY1+Y1(J)
51 SUMY2=SUMY2+Y2(J)
52 320 CONTINUE
53 SUMX1=0.
54 DO 340 I=1,N1
55 340 SUMX1=SUMX1+X1(I)*Q1(I)
D WRITE(CRT,341)SUMY1,SUMX1,(I,X1(I),Q1(I),I=1,N1)
D341 FORMAT(' POD1: SUMY=',G13.3,' GRANDSUMX=',G13.3/
D 1 (I10,2G13.3))
56 XNORM1=SUMY1/SUMX1
57 SUMX2=0.
58 DO 360 I=1,N2
59 360 SUMX2=SUMX2+X2(I)*Q2(I)
D WRITE(CRT,361)SUMY2,SUMX2,(I,X2(I),Q2(I),I=1,N2)
D361 FORMAT(' POD2* SUMY=',G13.3,' GRANDSUMX=',G13.3/
D 1 (I10,2G13.3))
60 XNORM2=SUMY2/SUMX2
```

```
      C
      C
1   400  DO 420 I=1,N1
2         X1(I)=X1(I)*XNORM1
3         Q1(I)=P1(I)
4         P1(I)=P1(I)*X1(I)**2
5         V1(I,I)=P1(I)
6   420  CONTINUE
7         DO 440 I=1,N2
8         X2(I)=X2(I)*XNORM2
9         Q2(I)=P2(I)
10        P2(I)=P2(I)*X2(I)**2
11        V2(I,I)=P2(I)
12  440  CONTINUE
13        WRITE(IOUT,442)(X1(I),I=1,N1)
14  442  FORMAT(1H0,'INITIAL INPUT VECTORS'/
15        1X,'POD 1',8G13.3/(7X,8G13.3))
16  444  WRITE(IOUT,444)(X2(I),I=1,N2)
17  444  FORMAT(1X,'POD 2',8G13.3/(7X,8G13.3))
18      C
19      N=MAX0(N1,N2)
20      DO 480 I=1,N
21        H(I)=1.
22  480  CONTINUE
23      C
24  500  RETURN
25      END
```

IE (Binary Time) Subroutine

BTIME subroutine is used to return to the user the binary time. This routine is written in Assembler.

n

.L BTIME (a,b,c,d,e)

re:

Integer to receive the Julian date.

Integer to receive the year.

Integer to receive the hour.

Integer to receive the minutes.

Integer to receive the seconds.

e

the parameter count is <5 or >5, then an exit is made to MIDAS.

mp le

EGER A,B,C,D,E

.L BTIME (A,B,C,D,E)

Appendix C: MIDAS System Utilities

1. STIME
2. MCLI
3. MTAPEF
4. GANDC

LE	001225	INTEGER*2	VARIABLE
IBLANK	001230	INTEGER*2	VARIABLE
ISTAR	001232	INTEGER*2	VARIABLE
DEPT	001235	REAL*4	PROCEDURE
J	001252	REAL*4	VARIABLE
TUB	001255	REAL*4	PROCEDURE
FOUR	001240	REAL*4	PROCEDURE
CHU	001275	REAL*4	VARIABLE
DIRT	002000	REAL*4	PROCEDURE
AL1	002370	REAL*4	VARIABLE
AL3	002375	REAL*4	VARIABLE
AL5	002400	REAL*4	VARIABLE
SOR3	002405	REAL*4	VARIABLE
SOP5	002412	REAL*4	VARIABLE
NM1	002416	INTEGER*2	VARIABLE
I	002420	INTEGER*2	VARIABLE
SIG	002422	REAL*4	VARIABLE
SIG3	002426	REAL*4	VARIABLE
SIG5	002432	REAL*4	VARIABLE
NM2	002436	INTEGER*2	VARIABLE
U1	002440	REAL*4	VARIABLE
U2	002444	REAL*4	VARIABLE
CHU1	002450	REAL*4	VARIABLE
J	002454	INTEGER*2	VARIABLE
AMAX1	002000	REAL*4	PROCEDURE
ICHFL1	002456	INTEGER*2	VARIABLE
XCH1	002460	REAL*4	VARIABLE
CHU2	002464	REAL*4	VARIABLE
ICHFL2	002470	INTEGER*2	VARIABLE
XCH2	002472	REAL*4	VARIABLE
ATIME	000000	REAL*4	PROCEDURE
NU1	002475	INTEGER*2	VARIABLE
NU2	002500	INTEGER*2	VARIABLE

COMMON BLOCK /HEAD/ LENGTH 000200

HDR1	000000	REAL*8	ARRAY (8)
HDR2	000100	REAL*8	ARRAY (8)

COMMON BLOCK /HDR/ LENGTH 000024

HEADER 000000 INTEGER*2 ARRAY (10)

COMMON BLOCK /IQ/ LENGTH 000004

IQ1	000000	INTEGER*2	VARIABLE
IQ2	000002	INTEGER*2	VARIABLE

FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
KDATE	000064	INTEGER*2 ARRAY (12)
X1	000022	REAL*4 PARAMETER ARRAY (8)
X2	000024	REAL*4 PARAMETER ARRAY (8)
P1	000026	REAL*4 PARAMETER ARRAY (8)
P2	000030	REAL*4 PARAMETER ARRAY (8)
Q1	000042	REAL*4 PARAMETER ARRAY (8)
Q2	000044	REAL*4 PARAMETER ARRAY (8)
Y1	000032	REAL*4 PARAMETER ARRAY (16)
Y2	000034	REAL*4 PARAMETER ARRAY (16)
R1	000046	REAL*4 PARAMETER ARRAY (16)
R2	000050	REAL*4 PARAMETER ARRAY (16)
S1	000036	REAL*4 PARAMETER ARRAY (16,8) VECTORED
S2	000040	REAL*4 PARAMETER ARRAY (16,8) VECTORED
A1X	000114	REAL*4 ARRAY (8)
A1SQ	000154	REAL*4 ARRAY (8)
A13	000214	REAL*4 ARRAY (8)
A15	000254	REAL*4 ARRAY (8)
A2X	000314	REAL*4 ARRAY (8)
A2SQ	000354	REAL*4 ARRAY (8)
A23	000414	REAL*4 ARRAY (8)
A25	000454	REAL*4 ARRAY (8)
XNM1	000514	REAL*4 ARRAY (8)
XNM2	000554	REAL*4 ARRAY (8)
VAR1	000614	REAL*4 ARRAY (8)
VAR2	000654	REAL*4 ARRAY (8)
XN1	000714	REAL*4 ARRAY (8)
XN2	000754	REAL*4 ARRAY (8)
SIG1	001014	REAL*4 ARRAY (8)
SIG2	001054	REAL*4 ARRAY (8)
J1FLAG	001114	INTEGER*2 ARRAY (8)
J2FLAG	001134	INTEGER*2 ARRAY (8)
IHEAD	001154	INTEGER*2 ARRAY (6)
AHEAD	001154	REAL*8 ARRAY (2)
K	000014	INTEGER*2 PARAMETER VARIABLE
NR	000016	INTEGER*2 PARAMETER VARIABLE
NID	000020	INTEGER*2 PARAMETER VARIABLE
N1	000052	INTEGER*2 PARAMETER VARIABLE
N2	000054	INTEGER*2 PARAMETER VARIABLE
M	000056	INTEGER*2 PARAMETER VARIABLE
MS	000060	INTEGER*2 PARAMETER VARIABLE
MF	000062	INTEGER*2 PARAMETER VARIABLE
AMX	001174	REAL*4 VARIABLE
AM3	001200	REAL*4 VARIABLE
AM5	001204	REAL*4 VARIABLE
MIN1	001210	INTEGER*2 VARIABLE
THRESH	001212	REAL*4 VARIABLE
THSIG	001216	REAL*4 VARIABLE
I1	001222	INTEGER*2 VARIABLE
I2	001224	INTEGER*2 VARIABLE

```
C
0136 400 CONTINUE
D WRITE(LP,360) IHEAD, (HDR1(I), I=1,N1), (HDR2(I), I=1,N2)
0137 IF(N1+N2.GT.10) GOTO 420
0139 WRITE(LP,410) NR,NID,XCH1,ICHFL1,XCH2,ICHFL2,
1 (X1(I),J1FLAG(I),I=1,N1),(X2(I),J2FLAG(I),I=1,N2)
0140 410 FORMAT(15,14,1X,2(F5.2,A1),10(1PG10.3,A1))
0141 WRITE(3,412) NR,NID,(XN1(I),I=1,N1),(XN2(I),I=1,N2)
0142 412 FORMAT(1X,2I5,1X,10(1PG11.3))
D WRITE(3,414) (A1X(I),I=1,N1),(A2X(I),I=1,N2)
D WRITE(3,414) (VAR1(I),I=1,N1),(VAR2(I),I=1,N2)
D414 FORMAT(12X,10(1PG11.3))
0143 GOTO 500
C
0144 420 WRITE(LP,440) NR,NID,XCH1,ICHFL1,XCH2,ICHFL2,
1 (X1(I),J1FLAG(I),I=1,N1),(X2(I),J2FLAG(I),I=1,N2)
0145 440 FORMAT(15,14,1X,2(F5.2,A1),10(1PG10.3,A1)/
1 (16X,10(G10.3,A1)))
0146 WRITE(3,442) NR,NID,(XN1(I),I=1,N1),(XN2(I),I=1,N2)
0147 442 FORMAT(1X,2I5,1X,10(1PG11.3)/(6X,10(G11.3)))
D WRITE(3,444) (A1X(I),I=1,N1),(A2X(I),I=1,N2)
D WRITE(3,444) (VAR1(I),I=1,N1),(VAR2(I),I=1,N2)
D444 FORMAT(12X,10(1PG11.3)/(6X,10(1PG11.3)))
C
0148 500 RETURN
0149 END
```

```

C
0100 200 U1=M-N1
0101      U2=M-N2
0102      CHU1=0.
0103      DO 210 J=1,M
0104 210 CHU1=CHU1+R1(J)**2/AMAX1(Y1(J),1.)
0105      CHU1=CHU1/U1
0106      ICHFL1=IBLANK
0107      XCH1=FXCH(CHU1,U1)
0108      IF(XCH1.GT.THRESH) ICHFL1=ISTAR
0110      CHU2=0.
0111      DO 220 J=1,M
0112 220 CHU2=CHU2+R2(J)**2/AMAX1(Y2(J),1.)
0113      CHU2=CHU2/U2
0114      ICHFL2=IBLANK
0115      XCH2=FXCH(CHU2,U2)
0116      IF(XCH2.GT.THRESH) ICHFL2=ISTAR
C
0118 300 IF(K.GT.1) GOTO 400
0120      CALL ATIME(KDATE)
0121      WRITE(LP,305) (HEADER(I),I=1,7),KDATE
0122 305 FORMAT(1H1,2X,7A2,3X,'PROCESSED ON ',12A2)
0123      WRITE(LP,310) MS,MF,M,THRESH,IQ1,IQ2
0124 310 FORMAT(1H0,'OUTPUT FOR KALMAN FILTER'/
1          1X,'USING DATA FROM CHANNEL',I4,' TO',I4/
2          1X,'CONDENSED TO',I3,' OUTPUT VECTOR CHANNELS'/
3          1X,'THRESHOLD IS',G13.3/
4          1X,'FINAL LEARNING RECORDS,',I5,' FOR POD 1,',
5          15,' FOR POD 2')
0125      NU1=M-N1
0126      NU2=M-N2
0127      WRITE(LP,320) I1,N1,NU1
0128      WRITE(LP,320) I2,N2,NU2
0129 320 FORMAT(1X,'INPUT VECTOR FOR POD',I1,' HAS',I3,
1          ' VARIABLE INTENSITIES, LEAVING',I3,' DEGREES OF FREEDOM')
0130      WRITE(LP,360) IHEAD,(HDR1(I),I=1,N1),
1          (HDR2(I),I=1,N2)
0131 360 FORMAT('0REC. MODE',1X,6A2,9(A8,3X),A8/
1          (17X,10(A8,3X)))
0132      WRITE(3,370) (HEADER(I),I=1,7),KDATE
0133 370 FORMAT(1X,7A2,2X,12A2)
0134      WRITE(3,380) (HDR1(I),I=1,N1),(HDR2(I),I=1,N2)
0135 380 FORMAT('0 REC. MODE',3X,9(A8,3X),A8/
1          (7X,10(A8,3X)))

```

```
      C  
      C  
0069      IF(K.GT.1) NM2=MNM  
0071      IF(NR.GT.102) NM2=1  
0073      DO 140 I=1,N2  
0074      A2X(I)=(A2X(I)*ALX+X2(I))/AMX  
0075      A2SQ(I)=(A2SQ(I)*ALX+X2(I)**2)/AMX  
0076      A23(I)=(A23(I)*AL3+X2(I))/AM3  
0077      A25(I)=(A25(I)*AL5+X2(I))/AM5  
0078      VAR2(I)=A2SQ(I)-A2X(I)**2  
0079      IF(K.EQ.1) VAR2(I)=Q2(I)*X2(I)**2  
0081      SIG=SQRT(VAR2(I))  
0082      IF(NM2.EQ.0) SIG2(I)=SIG  
0084      IF(NM2.EQ.0) XNM2(I)=A2X(I)  
0086      IF(SIG2(I).GT.1.E-20) XN2(I)=(X2(I)-XNM2(I))/SIG2(I)  
0088      SIG3=SIG/SQR3  
0089      SIG5=SIG/SQR5  
0090      J2FLAG(I)=IBLANK  
0091      IF(XN2(I).GT.THSIG) J2FLAG(I)=ISTAR  
0093      IF(X2(I)-A2X(I).GT.THSIG*SIG) J2FLAG(I)=ISTAR  
0095      IF(A23(I)-A2X(I).GT.THSIG*SIG3) J2FLAG(I)=ISTAR  
0097      IF(A25(I)-A2X(I).GT.THSIG*SIG5) J2FLAG(I)=ISTAR  
0099 140  CONTINUE
```

```
      C
0026 100 ALX=.95*AMX
0027      AMX=ALX+1.
0028      AL3=.666667*AM3
0029      AM3=AL3+1.
0030      AL5=.8*AM5
0031      AM5=AL5+1.
0032      SQR3=SQRT(3.)
0033      SQR5=SQRT(5.)
0034      IF(NID.EQ.2.OR.NID.EQ.3) MNM=1
0036      IF(NID.GT.10.AND.NID.LT.100) MNM=1
0038      IF(K.GT.1) NM1=MNM
0040      IF(NR.GT.1Q1) NM1=1
0042      DO 120 I=1,N1
0043      A1X(I)=(A1X(I)*ALX+X1(I))/AMX
0044      A1SQ(I)=(A1SQ(I)*ALX+X1(I)**2)/AMX
0045      A13(I)=(A13(I)*AL3+X1(I))/AM3
0046      A15(I)=(A15(I)*AL5+X1(I))/AM5
0047      VAR1(I)=A1SQ(I)-A1X(I)**2
0048      IF(K.EQ.1) VAR1(I)=Q1(I)*X1(I)**2
0050      SIG=SQRT(VAR1(I))
0051      IF(NM1.EQ.0) SIG1(I)=SIG
0053      IF(NM1.EQ.0) XNM1(I)=A1X(I)
0055      IF(SIG1(I).GT.1.E-20) XN1(I)=(X1(I)-XNM1(I))/SIG1(I)
0057      SIG3=SIG/SQR3
0058      SIG5=SIG/SQR5
0059      J1FLAG(I)=IBLANK
0060      IF(XN1(I).GT.THSIG) J1FLAG(I)=ISTAR
0062      IF(X1(I)-A1X(I).GT.THSIG*SIG) J1FLAG(I)=ISTAR
0064      IF(A13(I)-A1X(I).GT.THSIG*SIG3) J1FLAG(I)=ISTAR
0066      IF(A15(I)-A1X(I).GT.THSIG*SIG5) J1FLAG(I)=ISTAR
0068 120 CONTINUE
```

```

0001      SUBROUTINE KOUT(K,NR,NID,X1,X2,P1,P2,Y1,Y2,S1,S2,
1         Q1,Q2,R1,R2,N1,N2,M,MS,MF)
C         OUTPUT PROGRAM FOR THE KALMAN FILTER
C         LAST MODIFIED BY G.W.PHILLIPS, JANUARY 1985
C
0002      COMMON/HEAD/HDR1(8),HDR2(8)
0003      COMMON/HDR/HEADER(10)
0004      COMMON/IO/IQ1,IQ2
0005      INTEGER HEADER,KDATE(12)
0006      REAL*8 HDR1,HDR2
0007      REAL*4 X1(8),X2(8),P1(8),P2(8),Q1(8),Q2(8)
0008      REAL*4 Y1(16),Y2(16),R1(16),R2(16),S1(16,8),S2(16,8)
0009      REAL*4 A1X(8),A1SQ(8),A13(8),A15(8)
0010      REAL*4 A2X(8),A2SQ(8),A23(8),A25(8),XNM1(8),XNM2(8)
0011      REAL*4 VAR1(8),VAR2(8),XN1(8),XN2(8),SIG1(8),SIG2(8)
0012      INTEGER J1FLAG(8),J2FLAG(8),IHEAD(6)
0013      DATA A1X/8*0.,A1SQ/8*0.,A13/8*0.,A15/8*0./
0014      DATA A2X/8*0.,A2SQ/8*0.,A23/8*0.,A25/8*0./
0015      DATA XN1/8*0.,XN2/8*0.,SIG1/8*0.,SIG2/8*0./
0016      DATA XNM1/8*0.,XNM2/8*0./
0017      DATA AMX/0.,AM3/0.,AM5/0.,MNM/0/
0018      DATA THRESH/3.0/,THSIG/2.0/,I1/1/,I2/2/
0019      DATA LP/6/,IBLANK/1H/,ISTAR/1H*/
0020      REAL*8 AHEAD(2)
0021      DATA AHEAD/8HXSQ1 XS.8HQ2 /
0022      EQUIVALENCE (IHEAD,AHEAD)
C
0023      CBRT(U)=U**(.3333333)
0024      TW9(U)=2./(9.*U)
0025      FXCH(CHU,U)=(CBRT(CHU)-(1.-TW9(U)))/SQRT(TW9(U))
C
D      WRITE(LP,20) I1,(Y1(I),I=1,M)
D20    FORMAT(1H0,'POD',I2,' : OBSERVED VECTOR=',8G13.3/(6X8G13.3))
D      WRITE(LP,40) (R1(I),I=1,M)
D40    FORMAT(1X,'RESIDUALS=',8G13.3/(6X,8G13.3))
D      WRITE(LP,60) (X1(I),P1(I),I=1,N1)
D60    FORMAT(1X,'X,P=',4(G13.3,',',G9.3)/(6X,4(G13.3,',',G9.3)))
D      WRITE(LP,20) I2,(Y2(I),I=1,M)
D      WRITE(LP,40) (R2(I),I=1,M)
D      WRITE(LP,60) (X2(I),P2(I),I=1,N2)

```

MIDAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
X	000016	REAL*4 PARAMETER ARRAY (8)
Y	000020	REAL*4 PARAMETER ARRAY (16)
S	000022	REAL*4 PARAMETER ARRAY (16,3) VECTORED
O	000024	REAL*4 PARAMETER ARRAY (8)
R	000030	REAL*4 PARAMETER ARRAY (16)
QO	000026	REAL*4 PARAMETER ARRAY (8)
A1	000040	REAL*4 ARRAY (8)
A2	000100	REAL*4 ARRAY (8)
K	000014	INTEGER*2 PARAMETER VARIABLE
N	000032	INTEGER*2 PARAMETER VARIABLE
M	000034	INTEGER*2 PARAMETER VARIABLE
NR	000036	INTEGER*2 PARAMETER VARIABLE
IA	000140	INTEGER*2 VARIABLE
ICRT	000142	INTEGER*2 VARIABLE
A	000144	REAL*4 VARIABLE
B	000150	REAL*4 VARIABLE
I	000232	INTEGER*2 VARIABLE
AMAX1	000000	REAL*4 PROCEDURE
EPS	000234	REAL*4 VARIABLE
J	000240	INTEGER*2 VARIABLE
MOD	000000	INTEGER*2 PROCEDURE

COMMON BLOCK /IQ/ LENGTH 000004

IQ1	000000	INTEGER*2 VARIABLE
IQ2	000002	INTEGER*2 VARIABLE

```
0001      SUBROUTINE KSTEP(K,X,Y,S,Q,Q0,R,N,M,NR)
          C      SETS UP NOISE VARIANCES FOR NEXT STEP IN KALMAN FILTER
          C      LAST MODIFIED BY G.W.PHILLIPS, APRIL 1982
          C
0002      REAL*4 X(8),Y(16),S(16,8),Q(3),R(16),Q0(8)
0003      REAL*4 A1(8),A2(8)
0004      DATA A1/8*0.,A2/8*0./
0005      DATA IQ/9.,ICRT/5/
0006      DATA A/0.2/,B/0.8/
0007      COMMON/IQ/IQ1,IQ2
          C
0008      100  DO 120 I=1,N
0009           Q(I)=X(I)**2*Q0(I)
0010           Q(I)=AMAX1(Q(I),EPS)
0011      120  CONTINUE
0012           DO 140 J=1,M
0013           R(J)=AMAX1(Y(J),1.)
          C
          C      CALCULATE RUNNING AVERAGES OF Q(I)
          C      FOR EACH OF TWO PODS
          C
0014      200  IA=IA+1
0015           IF(MOD(IA,2).EQ.0) GOTO 250
0017           IF(NR.GT.IQ1) GOTO 230
0019           DO 220 I=1,N
0020           A1(I)=A*Q(I)+B*A1(I)
0021           DO 240 I=1,N
0022           Q(I)=AMAX1(Q(I),A1(I))
          D      GOTO 280
0023      RETURN
          C
0024      250  IF(NR.GT.IQ2) GOTO 270
0026           DO 260 I=1,N
0027           A2(I)=A*Q(I)+B*A2(I)
0028           DO 275 I=1,N
0029           Q(I)=AMAX1(Q(I),A2(I))
          D      GOTO 282
0030      RETURN
          C
0031      280  WRITE(ICRT,284) IA,(A1(I),I=1,N)
0032           GOTO 290
0033      282  WRITE(ICRT,284) IA,(A2(I),I=1,N)
0034      284  FORMAT(15,8(1PG10.3))
          C
0035      290  WRITE(ICRT,292) (Q(I),I=1,N)
0036      292  FORMAT(5X,8(1PG10.3))
0037      RETURN
0038      END
```

```
0001      SUBROUTINE LIBIN(LUF,M)
      C      READS DATA FROM ND SPECTRAL DATA FILES
      C      WRITTEN BY G.PHILLIPS, JULY 1981
      C
0002      COMMON/ARRAY/ARRAY(512)
      C
0003      DEFINE FILE LUF(0,2,0,IVAR)
0004      IVAR=193
0005      DO 100 K=1,M
0006      READ(LUF,IVAR,END=200)MSB,LSB
0007      RE1=MSB
0008      RE2=LSB
0009      IF(RE2.LT.0.)RE2=65536.+RE2
0010      IF(RE1.LT.0.)RE1=32768.+RE1
0011      IF(RE1.GE.16384.)RE1=RE1-16384.
0012      ARRAY(K)=RE1*65536.+RE2
0013
0015      100 CONTINUE
0016      200 M=K-1
0017      END FILE LUF
0018      RETURN
0019      END
0020
```

MIDAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
LUF	000014	INTEGER*2 PARAMETER VARIABLE
M	000016	INTEGER*2 PARAMETER VARIABLE
IVAR	000024	INTEGER*2 VARIABLE
K	000026	INTEGER*2 VARIABLE
MSB	000030	INTEGER*2 VARIABLE
LSB	000032	INTEGER*2 VARIABLE
RE1	000034	REAL*4 VARIABLE
RE2	000040	REAL*4 VARIABLE

COMMON BLOCK /ARRAY/ LENGTH 004000

ARRAY 000000 REAL*4 ARRAY (512)

```

N      001406  INTEGER*2 VARIABLE
MAX0   000000  INTEGER*2 PROCEDURE

COMMON BLOCK /ARRAY/      LENGTH 004000

ARRAY  000000  REAL*4      ARRAY (512)

COMMON BLOCK /HEAD/       LENGTH 000200

HDR1   000000  REAL*8      ARRAY (8)
HDR2   000100  REAL*8      ARRAY (8)

COMMON BLOCK /DATA/       LENGTH 000240

DATA   000000  INTEGER*2  ARRAY (80)

COMMON BLOCK /FREE/       LENGTH 000340

INTEG  000000  INTEGER*2  ARRAY (16)
REALX  000040  REAL*4     ARRAY (16)
ALPHA  000140  REAL*8     ARRAY (16)

```

MIDAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
X1	000014	REAL*4 PARAMETER ARRAY (8)
X2	000016	REAL*4 PARAMETER ARRAY (8)
Y1	000020	REAL*4 PARAMETER ARRAY (16)
Y2	000032	REAL*4 PARAMETER ARRAY (16)
S1	000042	REAL*4 PARAMETER ARRAY (16.8) VECTORED
S2	000044	REAL*4 PARAMETER ARRAY (16.8) VECTORED
P1	000032	REAL*4 PARAMETER ARRAY (8)
P2	000034	REAL*4 PARAMETER ARRAY (8)
Q1	000036	REAL*4 PARAMETER ARRAY (8)
Q2	000040	REAL*4 PARAMETER ARRAY (8)
H	000024	REAL*4 PARAMETER ARRAY (8)
V1	000026	REAL*4 PARAMETER ARRAY (8.8) VECTORED
V2	000030	REAL*4 PARAMETER ARRAY (8.8) VECTORED
FILDEF	000060	REAL*8 ARRAY (5)
N1	000046	INTEGER*2 PARAMETER VARIABLE
N2	000050	INTEGER*2 PARAMETER VARIABLE
M	000052	INTEGER*2 PARAMETER VARIABLE
MS	000054	INTEGER*2 PARAMETER VARIABLE
MF	000056	INTEGER*2 PARAMETER VARIABLE
CRT	000140	INTEGER*2 VARIABLE
PERIOD	000150	INTEGER*2 VARIABLE
BLANK	000152	INTEGER*2 VARIABLE
COMMA	000154	INTEGER*2 VARIABLE
ABLANK	000130	REAL*8 VARIABLE
LP	000142	INTEGER*2 VARIABLE
IN	000144	INTEGER*2 VARIABLE
IOU	000146	INTEGER*2 VARIABLE
LUF	000156	INTEGER*2 VARIABLE
LEN	001322	INTEGER*2 VARIABLE
I	001324	INTEGER*2 VARIABLE
NX	001326	INTEGER*2 VARIABLE
NY	001330	INTEGER*2 VARIABLE
NA	001332	INTEGER*2 VARIABLE
FREEFM	000000	REAL*4 PROCEDURE
MCLI	000000	INTEGER*2 PROCEDURE
FX	001334	REAL*4 VARIABLE
MI	001340	INTEGER*2 VARIABLE
LIBIN	000000	INTEGER*2 PROCEDURE
ML	001342	INTEGER*2 VARIABLE
IJ	001344	INTEGER*2 VARIABLE
J	001346	INTEGER*2 VARIABLE
MJ	001350	INTEGER*2 VARIABLE
SUM	001352	REAL*4 VARIABLE
SUMY1	001356	REAL*4 VARIABLE
SUMY2	001362	REAL*4 VARIABLE
SUMX1	001366	REAL*4 VARIABLE
XNORM1	001372	REAL*4 VARIABLE
SUMX2	001376	REAL*4 VARIABLE
XNORM2	001402	REAL*4 VARIABLE

MCLI (MIDAS Command Line Interpreter) Subroutine

The MCLI subroutine linked to FORTRAN allows the user to perform MIDAS control console statements from a FORTRAN program. This routine is written in assembler.

Form

CALL MCLI (<string>[<var>])

where:

<string> ASCII string to be executed; the string must have the same leading and terminating character; e.g., CALL MCLI ('@LUP@').

<var> Optional integer variable to receive the error code. If no <var> is specified and an error occurs, then a direct return to MIDAS is taken.

Codes returned in <var> are as follows:

- 1 if no errors normal condition.
- 2 if CLI\$ has returned an error.
- 3 if MIDAS command name is invalid.

Notes

The user should note that if MCLI is called with <1 or >2 arguments, then an immediate return is made to MIDAS.

The following MIDAS commands are forbidden in the MCLI subroutine: ABORT, BYE, CHAIN, DUPLICATE, ENDJOB, GOTO, HELLO, INIT, MESSAGE, PATCH, PAUSE, PROMPT, and REORDER.

The RUN command will cause the terminal to cease operating if either the calling program or the invoking program does an input from the terminal. The terminal will cease operating after both programs exit. This may require using CONTROL P to restore the terminal.

MTAPEF (FORMATTED MAGNETIC TAPE) SUBROUTINE

The MTAPEF subroutine controls the magnetic tape and its related functions. The subroutine is compatible with either 7 or 9 track magnetic tape. This subroutine is written in Assembler.

Form

CALL MTAPEF (a,b,c,d,e)

where:

- a = Command: INTEGER*2 variable (required argument).
- = 1 - Initialize control formatter (a,b).
 - = 2 - Transport off-line (a,b).
 - = 3 - Rewind (a,b).
 - = 4 - Search for logical EOT (a,b).
 - = 5 - Search for file (a,b,c).
 - = 6 - Search for record (a,b,c).
 - = 7 - Read one record (a,b,c,d,e).
 - = 8 - Verify one record (a,b,c,d,e).
 - = 9 - Write one record (a,b,c,d).
 - = 10 - Not used.
 - = 11 - Over write one record (a,b,c,d).
 - = 12 - Dump one record (a,b,c,d).
 - = 13 - Write one filemark (a,b).
 - = 14 - Write a logical EOT (a,b).
 - = 15 - Open transport (a,b,c).
 - = 16 - Close transport (a,b).
 - = 17 - Tagword (a,b,c,d)
- b = Error Number: INTEGER*2 variable (required).
- = 1 - No error.
 - = 2 - Transport assigned to other/or no user.
 - = 3 - Magnetic tape transport number error.
 - = 4 - Segment is read only.
 - = 5 - Segment is not accessible for I/O.
 - = 6 - Memory is not contiguous.
 - = 7 - Cross segments have different status.
 - = 8 - No filemark detected for last operation.
 - = 9 - Filemark detected during last operation.
 - = 10 - Located on or past physical EOT.
 - = 11 - Record read less than list word #14.
 - = 12 - Record read greater than list word #14.
 - = 13 - Invalid or undefined OP code.
 - = 14 - Data late.
 - = 15 - Invalid password.
 - = 16 - Motion error.
 - = 17 - Verification error.
 - = 18 - Write protect error.
 - = 19 - Parity CRC or LRC error during read.
 - = 20 - Operation attempt on off-line transport.
 - = 21 - No logical EOT detected during operation.

- = 22 - Logical EOT detected during operation.
- = 23 - Magnetic tape not off-line.
- = 24 - Undefined error bit in status.
- = 25 - Executive error during operation.
- = 26 - Illegal number of arguments.
- = 27 - Illegal command number.
- = 28 - Record length greater than 513 bytes for 7 track transport.

c = Command Parameter 1: INTEGER*2 variable.

<u>Command Number</u>	<u>Parameter</u>
5	# files to skip
6	# records to skip
7	# bytes to skip
8	# bytes to skip
9	# bytes to write
11	# bytes to write
12	# bytes to write
15	# transport to open
17	Subcommand code (c)
	c = 1 - Get tagword
	c = 2 - Increment tagword
	c = 3 - Put tagword

d = Command Parameter 2: INTEGER*2 variable or array name.

<u>Command Number</u>	<u>Parameter</u>
7	# bytes to read
8	# bytes to verify
9	Array name to write
11	Array name to write
12	Array name to write
17	Get, Increment, or Put tagword

e = Command parameter 3: Integer variable.

<u>Command Number</u>	<u>Parameter</u>
7	Array name to read
8	Array name to verify

OANDC (Open and Close) Subroutine

The OANDC subroutine allows the user to open a file and get the parameters in the device control block (DCB) table. If the input parameter EMFLAG is set to zero, the file is opened. The end sector is moved to its maximum extent. If the file is empty, a flag is returned to the user as EMFLAG=1 to indicate file was empty. OANDC then closes the file so that the FORTRAN program can open it. If a FORTRAN program uses OANDC, the program must call OANDC before the first read/write to the logical unit in the FORTRAN program. This subroutine is written in assembler.

Form

CALL OANDC (lunit,error,dvcode,absect,vlsect,dataty,rcsize,bylast,emflag)

where:

All the parameters are integers and have the following values:

lunit	Logical unit number (1-12) of the file (input parameter).
error	Error flag. Contains system error number if error occurred; otherwise set to zero. A negative error number means open error; a positive error number means close error.
dvcode	Device code (output parameter). = 0 - Teletype. = 2 - Dummy device. = 3 - Line printer. = 4 - High speed paper tape punch. = 5 - High speed paper tape reader. = 6 - Disk file.
absect	Absolute number of sectors in the file (output parameter).
vlsect	Valid number of sectors in the file (output parameter).
dataty	Data type (0-255) (output parameter).
rcsize	Record size (0-255) (output parameter).
bylast	Number of bytes in the last sector (1-256) (output parameter).
emflag	Empty flag (input and output parameter). Input - If zero, a file will have its end sector moved out to the absolute end sector minus one and bytes in the last sector set to 256. Output - Set to one if file was empty.

Appendix D: IMSL* Subroutines

1. DDKALM (FTKALM)
2. LEQTIF
3. LUDATF
4. LUELMF
5. UERTST
6. VMULFF
7. VMULFP

*Copyright, International Mathematical and Statistical Library, Inc.

```
C      SUBROUTINE DDKALM
C      BASED ON IMSL ROUTINE NAME   - FTKALM
C
C-----
C
C      COMPUTER           - ND 6620
C
C      LATEST REVISION    - JULY 1981 BY G.W.P.
C
C      PURPOSE            - KALMAN FILTERING
C
C      USAGE              - CALL DDKALM(K,X,H,Y,S,Q,R,P,IN,IS,IL,N,M1,L,
C                           T1,T2,IT,T3,IER)
C
C      ARGUMENTS          K   - INPUT STEP COUNTER. K=0,1,2,...
C                           WHEN K IS EQUAL TO ZERO, VECTOR X SHOULD
C                           CONTAIN THE PRIOR ESTIMATE OF THE MEAN OF X,
C                           AND THE PROGRAM CALCULATES THE ESTIMATED
C                           VARIANCE OF X AS P=GQG-TRANPOSE AT STEP 0.
C                           X   - INPUT/OUTPUT VECTOR OF LENGTH N. ON INPUT,
C                           X IS THE STATE VECTOR AT STEP K, AND ON
C                           OUTPUT, X CONTAINS THE ESTIMATED STATE
C                           VECTOR AT STEP K+1.
C                           H   - INPUT VECTOR OF DIMENSION N. H IS THE
C                           TRANSITION VECTOR AT STEP K.
C                           Y   - INPUT OBSERVATION VECTOR OF LENGTH M1 AT
C                           STEP K+1.
C                           S   - INPUT MATRIX OF DIMENSION M1 BY N AT STEP K+1.
C                           Q   - INPUT VARIANCE VECTOR OF DIMENSION L
C                           AT STEP K.
C                           R   - INPUT VARIANCE VECTOR OF DIMENSION
C                           M1 AT STEP K+1.
C                           P   - INPUT/OUTPUT MATRIX OF DIMENSION N BY N.
C                           ON INPUT, P IS THE VARIANCE MATRIX OF X
C                           AT STEP K. ON OUTPUT, P IS THE ESTIMATED
C                           VARIANCE MATRIX OF X AT STEP K+1.
C                           IN  - INPUT ROW DIMENSION OF THE MATRICES H,G, AND P
C                           EXACTLY AS SPECIFIED IN THE DIMENSION STATE-
C                           MENT IN THE CALLING PROGRAM.
C                           IS  - INPUT ROW DIMENSION OF THE MATRICES S AND R
C                           EXACTLY AS SPECIFIED IN THE DIMENSION STATE-
C                           MENT IN THE CALLING PROGRAM.
C                           IL  - INPUT ROW DIMENSION OF THE MATRIX Q EXACTLY
C                           AS SPECIFIED IN THE DIMENSION STATEMENT IN
C                           THE CALLING PROGRAM.
```

```
C      N      - INPUT SCALAR. SEE DESCRIPTIONS OF X,H,G,M,P.  
C      N MUST BE GREATER THAN 0.  
C      M1     - INPUT SCALAR. SEE DESCRIPTIONS OF Y,S,R,T3.  
C      M1 MUST BE GREATER THAN 0.  
C      L      - INPUT SCALAR. SEE DESCRIPTIONS OF G,Q.  
C      L MUST BE GREATER THAN 0.  
C      L MUST EQUAL N  
C      T1     - WORK MATRIX OF DIMENSION NM BY NM1, WHERE  
C      NM IS THE MAXIMUM OF N AND M1.  
C      T2     - WORK MATRIX OF DIMENSION NM BY NML, WHERE  
C      NML IS THE MAXIMUM OF NM AND L.  
C      IT     - ROW DIMENSION OF THE MATRICES T1 AND T2  
C      EXACTLY AS SPECIFIED IN THE DIMENSION  
C      STATEMENT IN THE CALLING PROGRAM.  
C      T3     - WORK VECTOR OF LENGTH M1.  
C      IER    - ERROR PARAMETER. (OUTPUT)  
C      TERMINAL ERROR  
C      IER=129 INDICATES ONE OF IN, IS, IL, OR IT  
C      IS TOO SMALL, OR THAT ONE OF N, M1,  
C      OR L IS NOT A POSITIVE INTEGER.  
C      IER=130 INDICATES AN ERROR OCCURRED IN  
C      IMSL ROUTINE LEQT1F.  
C  
C      REQD. IMSL ROUTINES - LEQT1F,LUDATF,LUELMF,UERTST,  
C      VMULFF,VMULFP,VXADD,VXMUL,VXSTO  
C-----
```

```

C
0001 SUBROUTINE DDKALM (K,X,H,Y,S,Q,R,P,IN,IS,IL,N,M1,L,T1,T2,IT,
1      T3,IER)
C      SPECIFICATIONS FOR ARGUMENTS
0002 INTEGER      K,IN,IS,IL,N,M1,L,IT,IER
0003 REAL*4      X(1),H(IN),Y(1),S(IS,1),Q(IL),
1      R(IS),P(IN,1),T1(IT,1),T2(IT,1),T3(1)
C      SPECIFICATIONS FOR LOCAL VARIABLES
0004 INTEGER      I,I0,I1,J
0005 DATA      I0/0/,I1/1/
C      FIRST EXECUTABLE STATEMENT
0006 IF (IN.GE.N .AND. IS.GE.M1 .AND. IL.GE.L
*      .AND. (IT.GE.N .OR. IT.GE.M1) .AND. N.GT.0
*      .AND. M1.GT.0 .AND. L.GT.0) GO TO 5
0008 IER = 129
0009 GO TO 9000
0010 S IER = 0
D      CALL KPRINT(K,X,H,G,Y,S,Q,R,P,IN,IS,IL,N,M1,L)
C      CALCULATE P IF K = ZERO
0011 IF (K .NE. 0) GO TO 10
0013 DO 6 I=1,N
0014 DO 6 J=1,N
0015 6 P(I,J)=0.
0016 DO 8 I=1,N
0017 8 P(I,I)=Q(I)
C      CALCULATE X-PRIME AT STEP K+1
0018 10 DO 15 I=1,N
0019 15 X(I)=X(I)*H(I)
C      CALCULATE P-PRIME AT STEP K+1
0020 DO 20 I=1,N
0021 DO 20 J=1,N
0022 20 P(I,J)=P(I,J)*H(I)*H(J)
0023 DO 25 I=1,N
0024 25 P(I,I)=P(I,I)+Q(I)

```

```

C                                     CALCULATE MATRIX K AT STEP K+1
C
0025 CALL VMULFF ( S, P, M1, N, N, IS, IN, T2, IT, IER)
0026 CALL VMULFF (T2, S, M1, N, M1, IT, IS, T1, IT, IER)
0027 DO 35 I = 1, M1
0028   T1(I, I) = T1(I, I) + R(I)
0029 23 CONTINUE
0030 CALL LSORTIF (T1, N, M1, IT, T2, IO, T3, IER)
0031 IF (IER .EQ. 0) GO TO 40
0032 IER = 130
0033 GO TO 9000
0034 40 DO 50 I = 1, M1
0035   DO 45 J = 1, N
0036     T1(J, I) = T2(I, J)
0037 45 CONTINUE
0038 50 CONTINUE
0039                                     CALCULATE X-HAT AT STEP K+1
C
0040 55 CALL VMULFF ( S, X, M1, N, I1, IS, IN, T3, IS, IER)
0041 DO 60 I = 1, M1
0042   T3(I) = T3(I) - Y(I)
0043 60 CONTINUE
0044 CALL VMULFF (T1, T3, N, M1, I1, IT, IS, T2, IT, IER)
0045 DO 65 I = 1, N
0046   X(I) = X(I) - T2(I, I)
0047 65 CONTINUE
C                                     CALCULATE P AT STEP K+1
0048 CALL VMULFF (T1, S, N, M1, N, IT, IS, T2, IT, IER)
0049 CALL VMULFF (T2, P, N, N, N, IT, IN, T1, IT, IER)
0050 DO 75 I = 1, N
0051   DO 70 J = 1, N
0052     P(I, J) = P(I, J) - T1(I, J)
0053 70 CONTINUE
0054 75 CONTINUE
0055 GO TO 9005
0056 9000 CONTINUE
0057 CALL UERTST (IER, 6HDDKALM)
0058 9005 RETURN
0059 END

```

MIDAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
X	000016	REAL*4 PARAMETER ARRAY (1)
Y	000020	REAL*4 PARAMETER ARRAY (IN)
IN	000034	INTEGER*2 PARAMETER VARIABLE
Y	000032	REAL*4 PARAMETER ARRAY (1)
S	000024	REAL*4 PARAMETER ARRAY (IS,1)
IS	000035	INTEGER*2 PARAMETER VARIABLE
Q	000016	REAL*4 PARAMETER ARRAY (IL)
IL	000040	INTEGER*2 PARAMETER VARIABLE
R	000030	REAL*4 PARAMETER ARRAY (IS)
P	000032	REAL*4 PARAMETER ARRAY (IN,1)
T1	000050	REAL*4 PARAMETER ARRAY (IT,1)
IT	000054	INTEGER*2 PARAMETER VARIABLE
T2	000052	REAL*4 PARAMETER ARRAY (IT,1)
T3	000055	REAL*4 PARAMETER ARRAY (1)
K	000014	INTEGER*2 PARAMETER VARIABLE
N	000042	INTEGER*2 PARAMETER VARIABLE
M1	000044	INTEGER*2 PARAMETER VARIABLE
L	000046	INTEGER*2 PARAMETER VARIABLE
IER	000060	INTEGER*2 PARAMETER VARIABLE
I	000076	INTEGER*2 VARIABLE
I0	000062	INTEGER*2 VARIABLE
I1	000064	INTEGER*2 VARIABLE
J	000100	INTEGER*2 VARIABLE
VMULFF	000000	REAL*4 PROCEDURE
LEQTF	000000	INTEGER*2 PROCEDURE
VMULFF	000000	REAL*4 PROCEDURE
UERTST	000000	REAL*4 PROCEDURE

C IMSL ROUTINE NAME - LEQT1F
C
C-----
C
C COMPUTER - ND 6620
C
C LATEST REVISION - MAY 1981 BY G.W.P.
C
C PURPOSE - LINEAR EQUATION SOLUTION - FULL STORAGE
C MODE - SPACE ECONOMIZER SOLUTION.
C
C USAGE - CALL LEQT1F (A,M,N,IA,B,IDGT,WKAREA,IER)
C
C ARGUMENTS A - INPUT MATRIX OF DIMENSION N BY N CONTAINING
C THE COEFFICIENT MATRIX OF THE EQUATION
C $AX = B$.
C ON OUTPUT, A IS REPLACED BY THE LU
C DECOMPOSITION OF A ROWWISE PERMUTATION OF
C A.
C M - NUMBER OF RIGHT-HAND SIDES. (INPUT)
C N - ORDER OF A AND NUMBER OF ROWS IN B. (INPUT)
C IA - ROW DIMENSION OF A AND B EXACTLY AS SPECIFIED
C IN THE DIMENSION STATEMENT OF THE CALLING
C PROGRAM. (INPUT)
C B - INPUT MATRIX OF DIMENSION N BY M CONTAINING
C RIGHT-HAND SIDES OF THE EQUATION $AX = B$.
C ON OUTPUT, THE N BY M SOLUTION X REPLACES B.
C IDGT - INPUT OPTION.
C IF IDGT IS GREATER THAN 0, THE ELEMENTS OF
C A AND B ARE ASSUMED TO BE CORRECT TO IDGT
C DECIMAL DIGITS AND THE ROUTINE PERFORMS
C AN ACCURACY TEST.
C IF IDGT EQUALS ZERO, THE ACCURACY TEST IS
C BYPASSED.
C WKAREA - WORK AREA OF DIMENSION GREATER THAN OR EQUAL
C TO N.
C IER - ERROR PARAMETER. (OUTPUT)
C TERMINAL ERROR
C IER = 129 INDICATES THAT MATRIX A IS
C ALGORITHMICALLY SINGULAR. (SEE THE
C CHAPTER L PRELUDE).
C WARNING ERROR
C IER = 34 INDICATES THAT THE ACCURACY TEST
C FAILED. THE COMPUTED SOLUTION MAY BE IN
C ERROR BY MORE THAN CAN BE ACCOUNTED FOR
C BY THE UNCERTAINTY OF THE DATA. THIS
C WARNING CAN BE PRODUCED ONLY IF IDGT IS
C GREATER THAN 0 ON INPUT. (SEE CHAPTER L
C PRELUDE FOR FURTHER DISCUSSION).
C-----

```

001 C      SUBROUTINE LEQTF (A,M,N,IA,B,IDGT,WKAREA,IER)
002 C      DIMENSION      A(IA,1),B(IA,1),WKAREA(1)
003 C      INITIALIZE IER
004 C      FIRST EXECUTABLE STATEMENT
005 IER=0
006 C      DECOMPOSE A
007 CALL LUDATF (A,A,N,IA,IDGT,D1,D2,WKAREA,WKAREA,WA,IER)
008 IF (IER .GT. 128) GO TO 9000
009 C      CALL ROUTINE LUELMF (FORWARD AND
010 C      BACKWARD SUBSTITUTIONS)
011 DO 10 J=1,M
012 CALL LUELMF (A,B(1,J),WKAREA,N,IA,B(1,J))
013 10 CONTINUE
014 IF (IER .EQ. 0) GO TO 9005
015 9000 CONTINUE
016 CALL UERTST (IER,6HLEQTF)
017 9005 RETURN
018 END

```

DAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
I	000014	REAL*4 PARAMETER ARRAY (IA,1)
IA	000022	INTEGER*2 PARAMETER VARIABLE
B	000024	REAL*4 PARAMETER ARRAY (IA,1)
WKAREA	000030	REAL*4 PARAMETER ARRAY (1)
N	000016	INTEGER*2 PARAMETER VARIABLE
M	000020	INTEGER*2 PARAMETER VARIABLE
IDGT	000026	INTEGER*2 PARAMETER VARIABLE
IER	000032	INTEGER*2 PARAMETER VARIABLE
LUDATF	000000	INTEGER*2 PROCEDURE
D1	000044	REAL*4 VARIABLE
D2	000050	REAL*4 VARIABLE
WA	000054	REAL*4 VARIABLE
J	000060	INTEGER*2 VARIABLE
LUELMF	000000	INTEGER*2 PROCEDURE
UERTST	000000	REAL*4 PROCEDURE

```
C  IMSL ROUTINE NAME   - LUDATF
C
C -----
C
C  COMPUTER           - ND6620
C
C  LATEST REVISION    - MAY 1981 BY G.W.P
C
C  PURPOSE            - L-U DECOMPOSITION BY THE CROUT ALGORITHM
C                      WITH OPTIONAL ACCURACY TEST.
C
C  USAGE              - CALL LUDATF (A,LU,N,IA,IDGT,D1,D2,IPVT,
C                      EQUIL,WA,IER)
C
C  ARGUMENTS          A   - INPUT MATRIX OF DIMENSION N BY N CONTAINING
C                          THE MATRIX TO BE DECOMPOSED.
C                      LU  - REAL OUTPUT MATRIX OF DIMENSION N BY N
C                          CONTAINING THE L-U DECOMPOSITION OF A
C                          ROWWISE PERMUTATION OF THE INPUT MATRIX.
C                          FOR A DESCRIPTION OF THE FORMAT OF LU, SEE
C                          EXAMPLE.
C                      N   - INPUT SCALAR CONTAINING THE ORDER OF THE
C                          MATRIX A.
C                      IA  - INPUT SCALAR CONTAINING THE ROW DIMENSION OF
C                          MATRICES A AND LU EXACTLY AS SPECIFIED IN
C                          THE CALLING PROGRAM.
C                      IDGT - INPUT OPTION.
C                          IF IDGT IS GREATER THAN ZERO, THE NON-ZERO
C                          ELEMENTS OF A ARE ASSUMED TO BE CORRECT TO
C                          IDGT DECIMAL PLACES. LUDATF PERFORMS AN
C                          ACCURACY TEST TO DETERMINE IF THE COMPUTED
C                          DECOMPOSITION IS THE EXACT DECOMPOSITION
C                          OF A MATRIX WHICH DIFFERS FROM THE GIVEN
C                          ONE BY LESS THAN ITS UNCERTAINTY.
C                          IF IDGT IS EQUAL TO ZERO, THE ACCURACY TEST
C                          IS BYPASSED.
```

```

C      D1      - OUTPUT SCALAR CONTAINING ONE OF THE TWO
C                COMPONENTS OF THE DETERMINANT. SEE
C                DESCRIPTION OF PARAMETER D2, BELOW.
C      D2      - OUTPUT SCALAR CONTAINING ONE OF THE
C                TWO COMPONENTS OF THE DETERMINANT. THE
C                DETERMINANT MAY BE EVALUATED AS (D1)(2**D2).
C      IPVT     - OUTPUT VECTOR OF LENGTH N CONTAINING THE
C                PERMUTATION INDICES. SEE DOCUMENT
C                (ALGORITHM).
C      EQUIL    - OUTPUT VECTOR OF LENGTH N CONTAINING
C                RECIPROCAL OF THE ABSOLUTE VALUES OF
C                THE LARGEST (IN ABSOLUTE VALUE) ELEMENT
C                IN EACH ROW.
C      WA      - ACCURACY TEST PARAMETER. OUTPUT ONLY IF
C                IDGT IS GREATER THAN ZERO.
C                SEE ELEMENT DOCUMENTATION FOR DETAILS.
C      IER      - ERROR PARAMETER. (OUTPUT)
C                TERMINAL ERROR
C                IER = 129 INDICATES THAT MATRIX A IS
C                ALGORITHMICALLY SINGULAR. (SEE THE
C                CHAPTER L PRELUDE).
C                WARNING ERROR
C                IER = 34 INDICATES THAT THE ACCURACY TEST
C                FAILED. THE COMPUTED SOLUTION MAY BE IN
C                ERROR BY MORE THAN CAN BE ACCOUNTED FOR
C                BY THE UNCERTAINTY OF THE DATA. THIS
C                WARNING CAN BE PRODUCED ONLY IF IDGT IS
C                GREATER THAN 0 ON INPUT. SEE CHAPTER L
C                PRELUDE FOR FURTHER DISCUSSION.
C
C      REQD. IMSL ROUTINES - UERTST,UGETIO
C
C      REMARKS   A TEST FOR SINGULARITY IS MADE AT TWO LEVELS:
C                1. A ROW OF THE ORIGINAL MATRIX A IS NULL.
C                2. A COLUMN BECOMES NULL IN THE FACTORIZATION PROCESS.
C
C      -----

```

AD-A154 995

KALMAN FILTER TIME SERIES ANALYSIS OF GAMMA-RAY DATA
FROM NAI(T1) DETECTORS FOR THE ND6620 COMPUTER(U) NAVAL
RESEARCH LAB WASHINGTON DC G W PHILLIPS 08 MAY 85

2/2

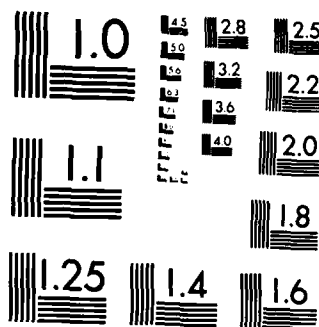
UNCLASSIFIED

NRL-MR-5541

F/G 20/6

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
C
0001 SUBROUTINE LUDATF (A,ALU,N,IA,IDGT,D1,D2,IPVT,EQUIL,WA,IER)
C
0002 DIMENSION A(IA,1),ALU(IA,1),IPVT(1),EQUIL(1)
0003 DATA ZERO,ONE,FOUR,SIXTN,SIXTH/0.,1.,4.,
* 16...0525/
C FIRST EXECUTABLE STATEMENT
C INITIALIZATION
D WRITE(6,1000)
D1000 FORMAT(1H0,'ENTERING LUDATF, INPUT MATRIX')
D DO 1020 I=1,N
D WRITE(6,1010)(A(I,J),J=1,N)
D1010 FORMAT(1X,8G13.3/(5X,8G13.3))
D1020 CONTINUE
0004 IER = 0
0005 RN = N
0006 WREL = ZERO
0007 D1 = ONE
0008 D2 = ZERO
0009 BIGA = ZERO
0010 DO 10 I=1,N
0011 BIG = ZERO
0012 DO 5 J=1,N
0013 P = A(I,J)
0014 ALU(I,J) = P
0015 P = ABS(P)
0016 IF (P .GT. BIG) BIG = P
0018 5 CONTINUE
0019 IF (BIG .GT. BIGA) BIGA = BIG
0021 IF (BIG .EQ. ZERO) GO TO 110
0023 EQUIL(I) = ONE/BIG
0024 10 CONTINUE
0025 DO 105 J=1,N
0026 JM1 = J-1
0027 IF (JM1 .LT. 1) GO TO 40
```

```

C                                     COMPUTE U(I,J), I=1,...,J-1
0029      DO 35 I=1, JM1
0030          SUM = ALU(I,J)
0031          IM1 = I-1
0032          IF (IDGT .EQ. 0) GO TO 25
C                                     WITH ACCURACY TEST
0034          AI = ABS(SUM)
0035          WI = ZERO
0036          IF (IM1 .LT. 1) GO TO 20
0038          DO 15 K=1, IM1
0039              T = ALU(I,K)*ALU(K,J)
0040              SUM = SUM-T
0041              WI = WI+ABS(T)
0042      15      CONTINUE
0043              ALU(I,J) = SUM
0044      20      WI = WI+ABS(SUM)
0045              IF (AI .EQ. ZERO) AI = BIGA
0047              TEST = WI/AI
0048              IF (TEST .GT. WREL) WREL = TEST
0050              GO TO 35
C                                     WITHOUT ACCURACY
0051      25      IF (IM1 .LT. 1) GO TO 35
0053              DO 30 K=1, IM1
0054                  SUM = SUM-ALU(I,K)*ALU(K,J)
0055      30      CONTINUE
0056              ALU(I,J) = SUM
0057      35      CONTINUE
0058      40      P = ZERO

```

```
      C                                COMPUTE U(J,J) AND L(I,J), I=J+1,...,
0059      DO 70 I=J,N
0060          SUM = ALU(I,J)
0061          IF (IDGT .EQ. 0) GO TO 55
      C                                WITH ACCURACY TEST
0063          AI = ABS(SUM)
0064          WI = ZERO
0065          IF (JMI .LT. 1) GO TO 50
0067          DO 45 K=1,JMI
0068              T = ALU(I,K)*ALU(K,J)
0069              SUM = SUM-T
0070              WI = WI+ABS(T)
0071      45      CONTINUE
0072          ALU(I,J) = SUM
0073      50      WI = WI+ABS(SUM)
0074          IF (AI .EQ. ZERO) AI = BIGA
0076          TEST = WI/AI
0077          IF (TEST .GT. WREL) WREL = TEST
0079          GO TO 65
      C                                WITHOUT ACCURACY TEST
0080      55      IF (JMI .LT. 1) GO TO 65
0082          DO 60 K=1,JMI
0083              SUM = SUM-ALU(I,K)*ALU(K,J)
0084      60      CONTINUE
0085          ALU(I,J) = SUM
0086      65      Q = EQUIL(I)*ABS(SUM)
0087          IF (P .GE. Q) GO TO 70
0089          P = Q
0090          IMAX = I
0091      70      CONTINUE
```

```

C                                TEST FOR ALGORITHMIC SINGULARITY
0092      IF (RN+P .EQ. RN) GO TO 110
0094      IF (J .EQ. IMAX) GO TO 80
C                                INTERCHANGE ROWS J AND IMAX
0096      D1 = -D1
0097      DO 75 K=1,N
0098          P = ALU(IMAX,K)
0099          ALU(IMAX,K) = ALU(J,K)
0100          ALU(J,K) = P
0101      75  CONTINUE
0102      EQUIL(IMAX) = EQUIL(J)
0103      80  IPVT(J) = IMAX
0104      D1 = D1*ALU(J,J)
0105      85  IF (ABS(D1) .LE. ONE) GO TO 90
0107      D1 = D1*SIXTH
0108      D2 = D2+FOUR
0109      GO TO 85
0110      90  IF (ABS(D1) .GE. SIXTH) GO TO 95
0112      D1 = D1*SIXTH
0113      D2 = D2-FOUR
0114      GO TO 90
0115      95  CONTINUE
0116      JP1 = J+1
0117      IF (JP1 .GT. N) GO TO 105
C                                DIVIDE BY PIVOT ELEMENT U(J,J)
0119      P = ALU(J,J)
0120      DO 100 I=JP1,N
0121          ALU(I,J) = ALU(I,J)/P
0122      100  CONTINUE
0123      105  CONTINUE
```

```
      C                                PERFORM ACCURACY TEST
0124      IF (IDGT .EQ. 0) GO TO 9005
0126      P = 3.14159
0127      WA = P*WREL
0128      IF (WA+10.*K*(-IDGT) .NE. WA) GO TO 9005
0130      IER = 34
0131      GO TO 9000

      C                                ALGORITHMIC SINGULARITY
0132      110 IER = 129
0133      D1 = ZERO
0134      D2 = ZERO
0135      9000 CONTINUE

      C                                PRINT ERROR
0136      CALL UERTST(IER,6HLDATF)
      D      STOP
      C
0137      9005 RETURN
0138      END
```

MIDAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
A	000014	REAL*4 PARAMETER ARRAY (IA,1)
IA	000022	INTEGER*2 PARAMETER VARIABLE
ALU	000016	REAL*4 PARAMETER ARRAY (IA,1)
IPVT	000032	INTEGER*2 PARAMETER ARRAY (1)
EQUIL	000034	REAL*4 PARAMETER ARRAY (1)
N	000020	INTEGER*2 PARAMETER VARIABLE
IDGT	000024	INTEGER*2 PARAMETER VARIABLE
D1	000026	REAL*4 PARAMETER VARIABLE
D2	000030	REAL*4 PARAMETER VARIABLE
WA	000036	REAL*4 PARAMETER VARIABLE
IER	000040	INTEGER*2 PARAMETER VARIABLE
ZERO	000042	REAL*4 VARIABLE
ONE	000046	REAL*4 VARIABLE
FOUR	000052	REAL*4 VARIABLE
SIXTH	000056	REAL*4 VARIABLE
SIXTH	000062	REAL*4 VARIABLE
RN	000076	REAL*4 VARIABLE
WREL	000102	REAL*4 VARIABLE
BIGA	000106	REAL*4 VARIABLE
I	000112	INTEGER*2 VARIABLE
BIG	000114	REAL*4 VARIABLE
J	000120	INTEGER*2 VARIABLE
P	000122	REAL*4 VARIABLE
ABS	000000	REAL*4 PROCEDURE
JM1	000126	INTEGER*2 VARIABLE
SUM	000130	REAL*4 VARIABLE
IM1	000134	INTEGER*2 VARIABLE
AI	000136	REAL*4 VARIABLE
WI	000142	REAL*4 VARIABLE
K	000146	INTEGER*2 VARIABLE
T	000150	REAL*4 VARIABLE
TEST	000154	REAL*4 VARIABLE
Q	000160	REAL*4 VARIABLE
IMAX	000164	INTEGER*2 VARIABLE
JP1	000166	INTEGER*2 VARIABLE
UERTST	000000	REAL*4 PROCEDURE

```
C  IMSL ROUTINE NAME   - LUELMF
C
C -----
C
C  COMPUTER           - ND 6620
C
C  LATEST REVISION    - MAY 1981 BY G.W.P.
C
C  PURPOSE            - ELIMINATION PART OF SOLUTION OF AX=B
C                      (FULL STORAGE MODE)
C
C  USAGE              - CALL LUELMF (A,B,IPVT,N,IA,X)
C
C  ARGUMENTS          A   - A = LU (THE RESULT COMPUTED IN THE IMSL
C                          ROUTINE LUDATF) WHERE L IS A LOWER
C                          TRIANGULAR MATRIX WITH ONES ON THE MAIN
C                          DIAGONAL. U IS UPPER TRIANGULAR. L AND U
C                          ARE STORED AS A SINGLE MATRIX A AND THE
C                          UNIT DIAGONAL OF L IS NOT STORED. (INPUT)
C                          B   - B IS A VECTOR OF LENGTH N ON THE RIGHT HAND
C                          SIDE OF THE EQUATION AX=B. (INPUT)
C                          IPVT - THE PERMUTATION MATRIX RETURNED FROM THE
C                          IMSL ROUTINE LUDATF, STORED AS AN N LENGTH
C                          VECTOR. (INPUT)
C                          N   - ORDER OF A AND NUMBER OF ROWS IN B. (INPUT)
C                          IA  - ROW DIMENSION OF A EXACTLY AS SPECIFIED IN
C                          THE DIMENSION STATEMENT IN THE CALLING
C                          PROGRAM. (INPUT)
C                          X   - THE RESULT X. (OUTPUT)
C -----
```

```

C
0001 SUBROUTINE LUELMF (A,B,IPVT,N,IA,X)
C
0002 DIMENSION          A(IA,1),B(1),IPVT(1),X(1)
C                      FIRST EXECUTABLE STATEMENT
C                      SOLVE LY = B FOR Y
0003 DO 5 I=1,N
0004 5 X(I) = B(I)
0005 IW = 0
0006 DO 20 I=1,N
0007 IP = IPVT(I)
0008 SUM = X(IP)
0009 X(IP) = X(I)
0010 IF (IW.EQ. 0) GO TO 15
0012 IM1 = I-1
0013 DO 10 J=IW,IM1
0014 SUM = SUM-A(I,J)*X(J)
0015 10 CONTINUE
0016 GO TO 20
0017 15 IF (SUM.NE. 0.) IW = I
0019 20 X(I) = SUM
C
0020 SOLVE UX = Y FOR X
0021 DO 30 IB=1,N
0022 I = N+1-IB
0023 IP1 = I+1
0024 SUM = X(I)
0025 IF (IP1.GT. N) GO TO 30
0026 DO 25 J=IP1,N
0027 SUM = SUM-A(I,J)*X(J)
0028 25 CONTINUE
0029 30 X(I) = SUM/A(I,I)
0030 RETURN
0031 END

```

MIDAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
A	000014	REAL*4 PARAMETER ARRAY (IA,1)
IA	000024	INTEGER*2 PARAMETER VARIABLE
B	000016	REAL*4 PARAMETER ARRAY (1)
IPVT	000020	INTEGER*2 PARAMETER ARRAY (1)
X	000026	REAL*4 PARAMETER ARRAY (1)
N	000022	INTEGER*2 PARAMETER VARIABLE
I	000030	INTEGER*2 VARIABLE
IW	000032	INTEGER*2 VARIABLE
IP	000034	INTEGER*2 VARIABLE
SUM	000036	REAL*4 VARIABLE
IM1	000042	INTEGER*2 VARIABLE
J	000044	INTEGER*2 VARIABLE
IB	000046	INTEGER*2 VARIABLE
IP1	000050	INTEGER*2 VARIABLE

```
C  IMSL ROUTINE NAME   - UERTST
C
C -----
C  COMPUTER           - ND 6620
C
C  LATEST REVISION    - MAY 1981 BY G.W.P.
C
C  PURPOSE            - PRINT A MESSAGE REFLECTING AN ERROR CONDITION
C
C  USAGE              - CALL UERTST (IER,NAME)
C
C  ARGUMENTS          IER  - ERROR PARAMETER. (INPUT)
C                        IER = I+J WHERE
C                        I = 128 IMPLIES TERMINAL ERROR.
C                        I = 64  IMPLIES WARNING WITH FIX, AND
C                        I = 32  IMPLIES WARNING.
C                        J = ERROR CODE RELEVANT TO CALLING
C                        ROUTINE.
C
C                        NAME - A SIX CHARACTER LITERAL STRING GIVING THE
C                        NAME OF THE CALLING ROUTINE. (INPUT)
C -----
```

```

C
01  SUBROUTINE UERTST (IER,NAME)
C      SPECIFICATIONS FOR ARGUMENTS
02  INTEGER          IER
03  INTEGER*2        NAME(3)
C      SPECIFICATIONS FOR LOCAL VARIABLES
04  INTEGER*2        NAMSET(3),NAMEQ(3)
05  DATA            IOUNIT/7/
06  DATA            NAMSET/2HUE,2HRS,2HET/
07  DATA            NAMEQ/2H ,2H ,2H /
C      FIRST EXECUTABLE STATEMENT
08  DATA            LEVEL/4/,IEQDF/0/,IEQ/1H=/
09  IF (IER.GT.999) GO TO 25
11  IF (IER.LT.-32) GO TO 55
113 IF (IER.LE.128) GO TO 5
115 IF (LEVEL.LT.1) GO TO 30
C      PRINT TERMINAL MESSAGE
117 IF (IEQDF.EQ.1) WRITE(IOUNIT,35) IER,NAMEQ,IEQ,NAME
119 IF (IEQDF.EQ.0) WRITE(IOUNIT,35) IER,NAME
121 GO TO 30
122 5 IF (IER.LE.64) GO TO 10
124 IF (LEVEL.LT.2) GO TO 30
C      PRINT WARNING WITH FIX MESSAGE
126 IF (IEQDF.EQ.1) WRITE(IOUNIT,40) IER,NAMEQ,IEQ,NAME
128 IF (IEQDF.EQ.0) WRITE(IOUNIT,40) IER,NAME
130 GO TO 30
131 10 IF (IER.LE.32) GO TO 15
C      PRINT WARNING MESSAGE
133 IF (LEVEL.LT.3) GO TO 30
135 IF (IEQDF.EQ.1) WRITE(IOUNIT,45) IER,NAMEQ,IEQ,NAME
137 IF (IEQDF.EQ.0) WRITE(IOUNIT,45) IER,NAME
139 GO TO 30
140 15 CONTINUE
C      CHECK FOR UERSET CALL
141 DO 20 I=1,3
142 IF (NAME(I).NE.NAMSET(I)) GO TO 25
144 20 CONTINUE
145 LEVOLD = LEVEL
146 LEVEL = IER
147 IER = LEVOLD
148 IF (LEVEL.LT.0) LEVEL = 4
150 IF (LEVEL.GT.4) LEVEL = 4
152 GO TO 30
153 25 CONTINUE
154 IF (LEVEL.LT.4) GO TO 30

```

```

C          PRINT NON-DEFINED MESSAGE
56      IF (IEQDF.EQ.1) WRITE(IUNIT,50) IER,NAMEQ,IEQ,NAME
58      IF (IEQDF.EQ.0) WRITE(IUNIT,50) IER,NAME
50      IEQDF = 0
51      RETURN
52      35 FORMAT(12H *** TERMINAL ERROR,10X,7H( IER = ,I3,
1          20H) FROM IMSL ROUTINE ,3A2,A1,3A2)
53      40 FORMAT(36H *** WARNING WITH FIX ERROR ( IER = ,I3,
1          20H) FROM IMSL ROUTINE ,3A2,A1,3A2)
54      45 FORMAT(18H *** WARNING ERROR,11X,7H( IER = ,I3,
1          20H) FROM IMSL ROUTINE ,3A2,A1,3A2)
55      50 FORMAT(20H *** UNDEFINED ERROR,9X,7H( IER = ,I5,
1          20H) FROM IMSL ROUTINE ,3A2,A1,3A2)
C          SAVE P FOR P = R CASE
C          P IS THE PAGE NAME
C          R IS THE ROUTINE NAME
56      55 IEQDF = 1
57      DO 60 I=1,3
58      60 NAMEQ(I) = NAME(I)
59      65 RETURN
70      END

```

MIDAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
NAME	000016	INTEGER*2 PARAMETER ARRAY (3)
NAMSET	000020	INTEGER*2 ARRAY (3)
NAMEQ	000026	INTEGER*2 ARRAY (3)
IER	000014	INTEGER*2 PARAMETER VARIABLE
IOUNIT	000034	INTEGER*2 VARIABLE
LEVEL	000036	INTEGER*2 VARIABLE
IEQDF	000040	INTEGER*2 VARIABLE
IEQ	000042	INTEGER*2 VARIABLE
I	000460	INTEGER*2 VARIABLE
LEVOLD	000462	INTEGER*2 VARIABLE

```
C  IMSL ROUTINE NAME   - VMULFF
C
C -----
C  COMPUTER           - ND 6620
C
C  LATEST REVISION    - MAY 1981 BY G.W.P.
C
C  PURPOSE            - MATRIX MULTIPLICATION (FULL STORAGE MODE)
C
C  USAGE              - CALL VMULFF (A,B,L,M,N,IA,IB,C,IC,IER)
C
C  ARGUMENTS          A   - L BY M MATRIX STORED IN FULL STORAGE MODE.
C                        (INPUT)
C                        B   - M BY N MATRIX STORED IN FULL STORAGE MODE.
C                        (INPUT)
C                        L   - NUMBER OF ROWS IN A. (INPUT)
C                        M   - NUMBER OF COLUMNS IN A (SAME AS NUMBER OF
C                        ROWS IN B). (INPUT)
C                        N   - NUMBER OF COLUMNS IN B. (INPUT)
C                        IA  - ROW DIMENSION OF MATRIX A EXACTLY AS
C                        SPECIFIED IN THE DIMENSION STATEMENT IN THE
C                        CALLING PROGRAM. (INPUT)
C                        IB  - ROW DIMENSION OF MATRIX B EXACTLY AS
C                        SPECIFIED IN THE DIMENSION STATEMENT IN THE
C                        CALLING PROGRAM. (INPUT)
C                        C   - L BY N MATRIX CONTAINING THE PRODUCT
C                        C = A*B. (OUTPUT)
C                        IC  - ROW DIMENSION OF MATRIX C EXACTLY AS
C                        SPECIFIED IN THE DIMENSION STATEMENT IN THE
C                        CALLING PROGRAM. (INPUT)
C                        IER - ERROR PARAMETER. (OUTPUT)
C                        TERMINAL ERROR
C                        IER=129 INDICATES A,B,OR C WAS DIMENSIONED
C                        INCORRECTLY.
C
C  REQD. IMSL ROUTINES - UERTST
C -----
```

```

C
0001 SUBROUTINE VMULFF (A,B,L,M,N,IA,IB,C,IC,IER)
C
C SPECIFICATIONS FOR ARGUMENTS
0002 INTEGER L,M,N,IA,IB,IC,IER
0003 REAL*4 A(IA,M),B(IB,N),C(IC,N)
C SPECIFICATIONS FOR LOCAL VARIABLES
0004 DOUBLE PRECISION TEMP
C FIRST EXECUTABLE STATEMENT
0005 IF (IA .GE. L .AND. IB .GE. M .AND. IC .GE. N) GO TO 5
C TERMINAL ERROR
0007 IER=129
0008 GO TO 9000
C ROW INDICATOR
0009 5 IER = 0
0010 DO 15 I=1,L
C COLUMN INDICATOR
0011 DO 15 J=1,N
0012 TEMP=0.0
C VECTOR DOT PRODUCT
0013 DO 10 K=1,M
0014 TEMP=A(I,K)*B(K,J)+TEMP
0015 10 CONTINUE
0016 C(I,J)=TEMP
0017 15 CONTINUE
0018 GO TO 9005
0019 9000 CONTINUE
0020 CALL UERTST (IER,6HVMULFF)
0021 9005 RETURN
0022 END
```

MIDAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
A	000014	REAL*4 PARAMETER ARRAY (IA,M)
IA	000026	INTEGER*2 PARAMETER VARIABLE
M	000022	INTEGER*2 PARAMETER VARIABLE
B	000016	REAL*4 PARAMETER ARRAY (IB,N)
IB	000030	INTEGER*2 PARAMETER VARIABLE
N	000024	INTEGER*2 PARAMETER VARIABLE
C	000032	REAL*4 PARAMETER ARRAY (IC,N)
IC	000034	INTEGER*2 PARAMETER VARIABLE
L	000020	INTEGER*2 PARAMETER VARIABLE
IER	000036	INTEGER*2 PARAMETER VARIABLE
TEMP	000050	REAL*8 VARIABLE
I	000060	INTEGER*2 VARIABLE
J	000062	INTEGER*2 VARIABLE
K	000064	INTEGER*2 VARIABLE
UERTST	000000	REAL*4 PROCEDURE

```
C   IMSL ROUTINE NAME   - VMULFP
C
C-----
C
C   COMPUTER            - ND6620
C
C   LATEST REVISION     - MAY 1981 BY G.W.P.
C
C   PURPOSE             - MATRIX MULTIPLICATION OF MATRIX A BY THE
C                       TRANSPOSE OF MATRIX B (FULL STORAGE MODE)
C
C   USAGE               - CALL VMULFP (A,B,L,M,N,IA,IB,C,IC,IER)
C
C   ARGUMENTS           A   - L BY M MATRIX STORED IN FULL STORAGE MODE.
C                       (INPUT)
C                       B   - N BY M MATRIX STORED IN FULL STORAGE MODE.
C                       (INPUT)
C                       L   - NUMBER OF ROWS IN A AND C. (INPUT)
C                       M   - NUMBER OF COLUMNS IN A AND B. (INPUT)
C                       N   - NUMBER OF ROWS IN MATRIX B AND NUMBER OF
C                           COLUMNS IN MATRIX C. (INPUT)
C                       IA  - ROW DIMENSION OF MATRIX A EXACTLY AS
C                           SPECIFIED IN THE DIMENSION STATEMENT IN THE
C                           CALLING PROGRAM. (INPUT)
C                       IB  - ROW DIMENSION OF MATRIX B EXACTLY AS
C                           SPECIFIED IN THE DIMENSION STATEMENT IN THE
C                           CALLING PROGRAM. (INPUT)
C                       C   - L BY N MATRIX CONTAINING THE PRODUCT
C                           C = A*B-TRANSPPOSE. (OUTPUT)
C                       IC  - ROW DIMENSION OF MATRIX C EXACTLY AS
C                           SPECIFIED IN THE DIMENSION STATEMENT IN THE
C                           CALLING PROGRAM. (INPUT)
C                       IER  - ERROR PARAMETER.
C                           TERMINAL ERROR
C                           IER=129 INDICATES A,B,OR C WAS DIMENSIONED
C                           INCORRECTLY.
C   REQD. IMSL ROUTINES - UERTST
C-----
```

```
      C
0001      SUBROUTINE VMULFP (A,B,L,M,N,IA,IB,C,IC,IER)
      C
0002      REAL*4  A(IA,M),B(IB,M),C(IC,N)
      C                      FIRST EXECUTABLE STATEMENT
0003      IF (IA.GE.L .AND. IB.GE.N .AND. IC.GE.L) GO TO 5
      C                      TERMINAL ERROR
0005      IER = 129
0006      GO TO 9000
      C                      ROW INDICATOR
0007      5 IER = 0
0008      DO 20 I = 1,L
      C                      COLUMN INDICATOR
0009          DO 15 J = 1,N
0010              TEMP = 0.0
      C                      VECTOR DOT PRODUCT
0011              DO 10 K = 1,M
0012                  TEMP = TEMP + A(I,K)*B(J,K)
0013          10      CONTINUE
0014              C(I,J) = TEMP
0015          15      CONTINUE
0016          20      CONTINUE
0017          GO TO 9005
0018      9000      CONTINUE
0019          CALL UERTST (IER,6HVMULFP)
0020      9005      RETURN
0021          END
```

MIDAS FORTRAN IV STORAGE MAP

NAME	OFFSET	ATTRIBUTES
A	000014	REAL*4 PARAMETER ARRAY (IA,M)
IA	000026	INTEGER*2 PARAMETER VARIABLE
M	000022	INTEGER*2 PARAMETER VARIABLE
B	000016	REAL*4 PARAMETER ARRAY (IB,M)
IB	000030	INTEGER*2 PARAMETER VARIABLE
C	000032	REAL*4 PARAMETER ARRAY (IC,N)
IC	000034	INTEGER*2 PARAMETER VARIABLE
N	000024	INTEGER*2 PARAMETER VARIABLE
L	000020	INTEGER*2 PARAMETER VARIABLE
IER	000036	INTEGER*2 PARAMETER VARIABLE
I	000050	INTEGER*2 VARIABLE
J	000052	INTEGER*2 VARIABLE
TEMP	000054	REAL*4 VARIABLE
K	000060	INTEGER*2 VARIABLE
UERTST	000000	REAL*4 PROCEDURE

END

FILMED

7-85

DTIC