

UNCLASSIFIED

AD NUMBER

ADB015216

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; NOV 1975. Other requests shall be referred to Air Force Flight Dynamics Lab., Wright-Patterson AFB, OH 45433.

AUTHORITY

AFFDL ltr 2 May 1979

THIS PAGE IS UNCLASSIFIED

THIS REPORT HAS BEEN DELIMITED
AND CLEARED FOR PUBLIC RELEASE
UNDER DOD DIRECTIVE 5200.20 AND
NO RESTRICTIONS ARE IMPOSED UPON
ITS USE AND DISCLOSURE.

DISTRIBUTION STATEMENT A

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

✓
AFFDL-TR-75-146

ADBOI5216

ACTIVE CONTROL SYNTHESIS FOR FLEXIBLE VEHICLES
Volume II KONPACT Program Listing

HONEYWELL
SYSTEMS & RESEARCH CENTER
2600 RIDGWAY PARKWAY
MINNEAPOLIS, MINNESOTA 55413

JULY 1976

TECHNICAL REPORT AFFDL-TR-75-146 FOR PERIOD APRIL 1975 - APRIL 1976

Distribution limited to U. S. Government agencies only; test and evaluation statement applied November 1975. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FCG), Wright-Patterson Air Force Base, Ohio 45433.

**COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION**

Prepared for
U.S. AIR FORCE FLIGHT DYNAMICS LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433



AD No. _____
DDC FILE COPY

NOTICE

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

Distribution limited to U. S. Government agencies only; test and evaluation statements applied November 1975. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FGC), Wright-Patterson Air Force Base, Ohio 45433.

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

Charles R. Stockdale

Charles R. Stockdale
Project Engineer

FOR THE COMMANDER

Edward H. Flinn

Edward H. Flinn, Chief
Control Criteria Branch
Air Force Flight Dynamics Laboratory

ACCESSION BY	
NTIS	White Envelope <input type="checkbox"/>
GRE	Self Storage <input checked="" type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
DISTRIBUTION AVAILABILITY STATEMENT	
DISC.	AVAIL. STATEMENT
B	

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER AFFDL-TR-75-146 Vol. 2	2. GOV'T ACCESSION NUMBER	3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (AND SUBTITLE) ACTIVE CONTROL SYNTHESIS FOR FLEXIBLE VEHICLES, Volume II. KONPACT Program Listing.		5. TYPE OF REPORT/PERIOD COVERED Final Report, April 1975 to April 1976	
7. AUTHOR(S) A. F. Konar J. K. Mahesh C. R. Stone M. Hank		6. PERFORMING ORG. REPORT NUMBER 76SRC21-Vol-2	
9. PERFORMING ORGANIZATIONS NAME/ADDRESS Honeywell Inc., Systems and Research Center 2600 Ridgway Parkway N.E. Minneapolis, Minnesota 55413		8. CONTRACT OR GRANT NUMBER(S) F33615-75-C-3046	
11. CONTROLLING OFFICE NAME/ADDRESS U. S. Air Force Flight Dynamics Laboratory Wright Patterson Air Force Base, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project 8219, Task 82190221	
14. MONITORING AGENCY NAME/ADDRESS (IF DIFFERENT FROM CONT. OFF.)		12. REPORT DATE January 1976	
		13. NUMBER OF PAGES 308	
		15. SECURITY CLASSIFICATION (OF THIS REPORT) Unclassified	
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (OF THIS REPORT) Distribution limited to U. S. Government agencies only; test and evaluation statement applied November 1975. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FGC), Wright-Patterson Air Force Base, Ohio 45433.			
17. DISTRIBUTION STATEMENT (OF THE ABSTRACT ENTERED IN BLOCK 20, IF DIFFERENT FROM REPORT)			
18. SUPPLEMENTARY NOTES			
19. KEY WORDS (CONTINUE ON REVERSE SIDE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER) Active Control C-5A Overlay CCV (Control Configured Vehicles) Optimal Control Variable Dimensioning Flight FLEXSTAB Dynamic Data Storage Flexible Vehicle Modeling PRECOMPILER			
20. ABSTRACT (CONTINUE ON REVERSE SIDE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER) KONPACT is a system of computer programs that will design optimal or suboptimal control systems especially for aircraft with lightly damped modes. This program represents advanced computational techniques to perform modern control synthesis, analysis and design of automatic control systems. These programs augment aircraft mathematical models produced from such advanced program as the FLEXSTAB Level 2.01.00 with control system dynamics and then design and analyze			

HD-168 REV 11/74

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 55 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

402 349

next page

mt

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

cont.

→ quadratic optimal or suboptimal control systems.

The KONPACT Program Listings is the second volume of report prepared under contract F33615-75-C-3046.

→ This report

X contains the program listings of KONPACT.



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

FOREWORD

The research described in this report was prepared by Honeywell Inc., Minneapolis, Minnesota 55413, under Air Force Contract F33615-75-C-3046. It was initiated under the AFFDL task number 82190221, "Optimal Control of Flexible Aircraft," project number 8219 "Stability and Control of Aerospace Vehicles." This work was directed by the Control Criteria Branch (FGC), Flight Control Division of the Air Force Flight Dynamics Laboratory and was administered by Mr. Charles R. Stockdale of the Control Criteria Branch. Special thanks to Mr. Robert C. Schwanz of FGC and Mr. Gary Grimes of ASD/ADDP for their continued support toward this contract.

The technical work reported in this volume was conducted by the Research Department at the Systems and Research Center of Honeywell Inc. Dr. A. F. Konar was the Honeywell Program Manager and the principal investigator on this contract. He was assisted by Mr. C. R. Stone, Dr. J. K. Mahesh, and Miss M. Hank. This report covers work from April 1975 to April 1976.

The work under this contract was reported in three volumes entitled, "Active Control Synthesis for Flexible Vehicles."

Volume I.	KONPACT Theoretical Description	<i>AD-B015 1984</i>
Volume II.	KONPACT Program Listing	
Volume III.	KONPACT Users Manual	<i>AD-B015 0254</i>

TABLE OF CONTENTS

Section		Page
I	INTRODUCTION	1
II	DESCRIPTION OF KONPACT PROGRAMS	3
	Overlay Organization	3
	Information Flow	4
	Variable Dimensioning	7
III	MODELING PROGRAM (KONPACT-1)	10
	Overlay Structure	10
	Description of Main Programs	10
	Program MAIN	10
	Program MAIN1	18
	Program MAIN2	18
	Program MAIN3	18
	Program MAIN4	22
	Program MAIN5	22
	Description of Basic Subroutines	22
	Subroutine KORG1	22
	Subroutine NAMEL	26
	Subroutine STAMK1	26
	Subroutine SIMK1	26
	Subroutine STAMK2	26
	Subroutine SIMKT	28
	Subroutine TRANSK	28
	Subroutine DFN	28

TABLE OF CONTENTS (continued)

Section	Page
Subroutine PHERR	28
Subroutine STAMK3	30
Subroutine SIMK	30
Subroutine QUADK	30
Subroutine STAMK4	30
Subroutine SIMK2	34
Subroutine CONDK	34
Subroutine MNAME	34
Subroutine IMRATE	34
Subroutine DIFFK	35
Subroutine REDUCE	35
Subroutine SCAL	35
Subroutine SHUFF	35
Subroutine SHUF1	35
Subroutine SHUF2	36
Description of Auxiliary Subroutines	36
Subroutine QDIO	36
Subroutine IDRO	36
Subroutine RSDRD	36
Subroutine SDRD	37
Subroutine FILE	37
Subroutine TPR	37
Subroutine HPR	37
Subroutine IDPR	37

TABLE OF CONTENTS (continued)

Section	Page
Subroutine MPRS	38
Subroutine MPRS1	38
Subroutine ZERO	38
Subroutine INPT	38
Subroutine INPT1	38
Subroutine DEBUG	39
Subroutine ERRM	39
Subroutine DERRM	39
Subroutine DERRMS	39
Subroutine SHIFT	39
Subroutine TDINVR	39
 IV	
DESIGN PROGRAM (KONPACT-2)	40
Overlay Structure	40
Description of Main Programs	45
Program MAIN	45
Program DIAK	45
Program FFOC	45
Program DATAK	45
Description of Basic Subroutines	47
Subroutine KORG2	47
Subroutine TIMER	47
Subroutine SGUST	47
Subroutine CAL1	47
Subroutine STRIC	47
Subroutine SHUFL	47

TABLE OF CONTENTS (continued)

Section	Page
Subroutine SHUF	48
Subroutine RESP	48
Subroutine COVAR	48
Subroutine COSTAT	48
Subroutine TRANS	49
Subroutine UNSCR	49
Subroutine GCAL	49
Subroutine CAL	49
Subroutine DDIAK	49
Subroutine DFFOC	50
Subroutine DLSA	50
Subroutine FINK	50
Description of Auxiliary Subroutines	50
Subroutine MP	50
Subroutine OUTP	51
Subroutine POLES	51
Subroutine HESSEN	51
Subroutine QRCALL	51
Subroutine QR	51
Subroutine INPTM	51
Subroutine WTP	52
Function GRAN	52
V CONCLUSIONS AND RECOMMENDATIONS	53
Recommendations for Future Software Development Work	53
Conclusions	53

TABLE OF CONTENTS (concluded)

Section	Page
APPENDIX	303
REFERENCES	308

LIST OF FIGURES

Figure		Page
1	Interface Between LSA and KONPACT Programs	54
2	Overlay Structure of KONPACT-1	55
3	Overlay Structure of KONPACT-2	56
4	Overlay Structure and Subroutines in KONPACT-1	57
5	Program MAIN Program Listing	58
6	Program MAIN1 Program Listing	59
7	Program MAIN2 Program Listing	60 - 61
8	Program MAIN3 Program Listing	62 - 63
9	Program MAIN4 Program Listing	64
10	Program MAIN5 Program Listing	65 - 66
11	Subroutine KORG1 Flow Chart	67 - 69
12	Subroutine KORG1 Program Listing	70 - 74
13	Subroutine NAMEL Flow Chart	75
14	Subroutine NAMEL Program Listing	76 - 82
15	Subroutine STAMK1 Flow Chart	83
16	Subroutine STAMK1 Program Listing	84 - 86
17	Subroutine SIMK1 Flow Chart	87
18	Subroutine SIMK1 Program Listing	88 - 101

LIST OF FIGURES (continued)

Figure		Page
19	Subroutine STAMK2 Flow Chart	102
20	Subroutine STAMK2 Program Listing	103 - 107
21	Subroutine SIMKT Flow Chart	108
22	Subroutine SIMKT Program Listing	109 - 113
23	Subroutine TRANSK Flow Chart	114
24	Subroutine TRANSK Program Listing	115
25	Subroutine DFN Flow Chart	116
26	Subroutine DFN Program Listing	117 - 119
27	Subroutine PHERR Flow Chart	120
28	Subroutine PHERR Program Listing	121
29	Subroutine STAMK3 Flow Chart	122
30	Subroutine STAMK3 Program Listing	123 - 127
31	Subroutine SIMK Flow Chart	128
32	Subroutine SIMK Program Listing	129 - 136
33	Subroutine QUADK Flow Chart	137
34	Subroutine QUADK Program Listing	138 - 139
35	Subroutine STAMK4 Flow Chart	140
36	Subroutine STAMK4 Program Listing	141 - 143

LIST OF FIGURES (continued)

Figure		Page
37	Subroutine SIMK2 Flow Chart	144
38	Subroutine SIMK2 Program Listing	145 - 147
39	Subroutine CONDK Flow Chart	148 - 149
40	Subroutine CONDK Program Listing	150 - 155
41	Subroutine MNAME Flow Chart	156
42	Subroutine MNAME Program Listing	157 - 161
43	Subroutine IMRATE Flow Chart	162
44	Subroutine IMRATE Program Listing	163 - 164
45	Subroutine DIFFK Flow Chart	165
46	Subroutine DIFFK Program Listing	166 - 167
47	Subroutine REDUCE Flow Chart	168
48	Subroutine REDUCE Program Listing	169 - 171
49	Subroutine SCAL Flow Chart	172
50	Subroutine SCAL Program Listing	173
51	Subroutine SHUFF Flow Chart	174
52	Subroutine SHUFF Program Listing	175
53	Subroutine SHUFF1 Flow Chart	176
54	Subroutine SHUF1 Program Listing	177
55	Subroutine SHUF2 Flow Chart	178

LIST OF FIGURES (continued)

Figure		Page
56	Subroutine SHUF2 Program Listing	179
57	Subroutine QDIO Flow Chart	180
58	Subroutine QDIO Program Listing	181 - 183
59	Subroutine IDRO Flow Chart	184
60	Subroutine IDRO Program Listing	185 - 187
61	Subroutine RSDRD Flow Chart	188
62	Subroutine RSDRD Program Listing	189 - 192
63	Subroutine SDRD Flow Chart	193
64	Subroutine SDRD Program Listing	194 - 195
65	Subroutine FILE Flow Chart	196
66	Subroutine FILE Program Listing	197 - 198
67	Subroutine TPR Flow Chart	199
68	Subroutine TPR Program Listing	200 - 201
69	Subroutine HPR Program Listing	202
70	Subroutine IDPR Program Listing	203
71	Subroutine MPRS Program Listing	204 - 205
72	Subroutine MPRS1 Program Listing	206
73	Subroutine ZERO Program Listing	207
74	Subroutine INPT Program Listing	208

LIST OF FIGURES (continued)

Figure		Page
75	Subroutine INPT1 Program Listing	209
76	Subroutine DEBUG Program Listing	210
77	Subroutine ERRM Program Listing	211
78	Subroutine DERRM Program Listing	212
79	Subroutine DERRMS Program Listing	213
80	Subroutine SHIFT Program Listing	214
81	Subroutine TDINVR Program Listing	215 - 217
82	Overlay Structure and Subroutines in KONPACT-2	218
83	Program MAIN Program Listing	219
84	Program DIAK Program Listing	220 - 234
85	Program FFOC Program Listing	235 - 244
86	Program DATAK Program Listing	245 - 246
87	Subroutine KOR2 Flow Chart	247 - 248
88	Subroutine KOR2 Program Listing	249 - 251
89	Subroutine TIMER Program Listing	252 - 256
90	Subroutine SGUST Program Listing	257
91	Subroutine CAL1 Program Listing	258 - 259
92	Subroutine STRIC Program Listing	260
93	Subroutine SHUFL Program Listing	261

LIST OF FIGURES (continued)

Figure		Page
94	Subroutine SHUF Program Listing	262
95	Subroutine RESP Program Listing	263 - 265
96	Subroutine COVAR Program Listing	266 - 267
97	Subroutine COSTAT Program Listing	268
98	Subroutine TRANS Program Listing	269
99	Subroutine UNSCR Program Listing	270
100	Subroutine GCAL Program Listing	271 - 272
101	Subroutine CAL Program Listing	273 - 274
102	Subroutine DDIAK Flow Chart	275
103	Subroutine DDIAK Program Listing	276 - 278
104	Subroutine DFFOC Flow Chart	279
105	Subroutine DFFOC Program Listing	280 - 282
106	Subroutine DLSA Flow Chart	283
107	Subroutine DLSA Program Listing	284 - 286
108	Subroutine FINK Flow Chart	287
109	Subroutine FINK Program Listing	288 - 289
110	Subroutine MP Program Listing	290
111	Subroutine OUTP Program Listing	291

LIST OF FIGURES (concluded)

Figure		Page
112	Subroutine POLES Program Listing	292
113	Subroutine HESSEN Program Listing	293 - 294
114	Subroutine QRCALL Program Listing	295 - 296
115	Subroutine QR Program Listing	297 - 298
116	Subroutine INPTM Program Listing	299
117	Subroutine WTP Program Listing	300
118	Subroutine GRAN Program Listing	301
119	Program PRECOM Flow Chart	302
120	Program PRECOM Program Listing	304 - 307

LIST OF TABLES

Table		Page
1	KONPACT Program Descriptions	5
2	KONPACT Data Tapes	7
3	Typical Dynamic Storage Map	9
4	KONPACT-1 Subroutine Summary	11 - 17
5	Dynamic Storage Map for Program MAIN1	19
6	Dynamic Storage Map for Program MAIN2	20
7	Dynamic Storage Map for Program MAIN3	21
8	Dynamic Storage Map for Program MAIN4	23
9	Dynamic Storage Map for Program MAIN5	24 - 25
10	Dynamic Storage Map for Program STAMP1	27
11	Dynamic Storage Map for Program STAMP2	29
12	Dynamic Storage Map for Program STAMP3	31 - 32
13	Dynamic Storage Map for Program STAMP4	33
14	KONPACT-2 Subroutine Summary	41 - 44
15	Dynamic Storage Map for Program DATAK	46

SECTION I

INTRODUCTION

The general objective of this program is to develop techniques and tools necessary for rapid design of an active control system for aircraft with lightly damped structural modes. The synthesis techniques provided here are aimed at reducing the engineering man-hours presently required for flight control system design thus effecting a cost reduction. Improvements in the fatigue life, ride qualities, and/or handling qualities of military aircraft are sought by controlling the lightly damped modes thus improving mission performance.

The present scope of this program is to develop programs to interface the level 2.01.00 FLEXSTAB computer program system with existing Air Force-owned optimal control computer programs. These programs represent advanced computational techniques required to perform quantitative analysis of multi-surface control systems. The resulting interface program system is called "KONPACT - Computer Programs for Active Control Technology." KONPACT provides the capability to model, synthesize, analyze, and design automatic control systems by efficiently working together with FLEXSTAB. It can also be used as a stand-alone program.

The work performed under this contract is reported in three volumes:

- Volume I. KONPACT Theoretical Description and Demonstration
- Volume II. KONPACT Program Listing
- Volume III. KONPACT Users Manual

This document reports the program listings of KONPACT. Complete documentation of KONPACT is beyond the scope of this contract.

Section II presents a brief description of KONPACT programs. The variable dimensioning technique for efficient data storage and memory allocation is discussed here. This approach is used throughout KONPACT-1.

The Modeling Program (KONPACT-1) is described in Section III. The Design Program (KONPACT-2) is described in Section IV. The appendix contains a description of the precompiler program for KONPACT-1.

The analytical techniques and algorithms used in KONPACT are described in Volume I. Volume I also demonstrates how these techniques are applied to flexible aircraft control system design.

User's information on KONPACT is given in Volume III. The input cards are fully described for each program. Brief descriptions of programs and information flow in KONPACT are also presented for completeness. Demonstration examples are included to guide the user in data mechanics.

SECTION II

DESCRIPTION OF KONPACT PROGRAMS

KONPACT is a system of computer programs developed by Honeywell under Air Force Contract No. F33615-75-C-3046. KONPACT uses the state space approach for modeling flight control systems and designs the controllers using optimal control methodology. KONPACT interfaces with the Linear Systems Analysis (LSA) Program of the Level 2 FLEXSTAB Program system developed by Boeing under Air Force Contract No. F33615-72-C-1172 (Reference 1). KONPACT can also be used as a stand-alone program.

KONPACT operates on CDC6000 and CDC7000 series computers and can be easily modified to operate on other computers. KONPACT has been written in Extended Fortran IV language.

In this section, a description of KONPACT programs is presented in terms of overlay organization and information flow.

OVERLAY ORGANIZATION

KONPACT consists of two programs, namely, a modeling program (KONPACT-1) and a design program (KONPACT-2). KONPACT-1 interfaces with FLEXSTAB through the LSA program to obtain the vehicle model and augments the specified dynamics to obtain the state space description (quadruple data) of the flight control system. These data are utilized by KONPACT-2

which contains the subprograms DIAK and FFOC (documented in Reference 2) to the design of the optimal feedback gains. DIAK stands for Doubly Iterative Algorithm developed by Konar (Reference 5). The DIAK program designs full state feedback optimal controllers. FFOC stands for Fixed Form Optimal Controllers. FFOC stands for Fixed Form Optimal Control developed by Stein and Scharmack (Reference 6). The FFOC program designs reduced state (practical) feedback optimal controllers. KONPACT-2 also interfaces with FLEXSTAB through the LSA program to evaluate performances of the above designed optimal flight control system.

Table 1 provides a brief description of programs KONPACT-1 and KONPACT-2 and their subprograms. The interface between KONPACT and the LSA program is illustrated in Figure 1. The overlay structure of KONPACT-1 program is illustrated in Figure 2. It consists of a main overlay and five primary overlays (Reference 3). The overlay structure of KONPACT-2 program is illustrated in Figure 3. It consists of a main overlay and three primary overlays.

INFORMATION FLOW

The normal sequence for obtaining an overall state space model of a flight control system using the modeling program (KONPACT-1) is as follows:

- The vehicle model is obtained by using either subprogram STAMK1 for LSA data or subprogram STAMK4 for other types of vehicle data.

Table 1. KONPACT Program Descriptions

PROGRAM	SUBPROGRAM	DESCRIPTION
KONPACT-1		State space modeling program
	STAMK1	Obtains state space model from LSA simulator deck data
	STAMK2	Obtains state space model from transfer function data
	STAMK3	Obtains state space model from quadruple data and interconnection data
	STAMK4	Obtains state space model from simulation equations (user written)
KONPACT-2	CONDK	Modifies the state space model by scaling, shuffling, truncating and residualizing the system variables
		Optimal design program
	DATAK	Prepares data for DIAK, FFOC and LSA programs
	DIAK	Designs full state feedback optimal controllers
	FFOC	Designs reduced state (Practical) feedback optimal controllers

- The actuator, sensor, controller, implicit and explicit models are obtained by using either subprogram STAMK2 with transfer function input data or subprogram STAMK3 with quadruple input data.
- The subsystems defined above are combined to get an overall system by using subprogram STAMK3 with interconnection input data.
- The overall system model is conditioned (modified) by scaling and/or shuffling and/or truncating and/or residualizing the variables using the CONDK program. This program also develops the rate of change of response variables when required.

The normal sequence for designing optimal feedback controllers and evaluating the performance of the resulting system using the design program KONPACT-2 is as follows:

- Full state feedback control gains are obtained by varying the quadratic weights and using the DIAK subprogram.
- The resulting full state feedback control gains are reduced to gains only on specified measurements by using the FFOC subprogram.
- The performance of the resulting closed loop system is evaluated using the LSA program.
- The above steps are repeated until a satisfactory design is obtained.

Table 2 describes all the data tapes used in KONPACT-1 and KONPACT-2 programs. The state space model data (quadruple data) and the Name List data are written on tapes QDATA and NDATA, respectively. The

vehicle data (simulator deck data) are written on tape VDATA. The feedback gain data from DIAK and FFOC are written on tapes DDATA and FDATA, respectively. The overall system data in frequency representation form are written on tape SDSTP for use by the LSA program. The DATAK subprogram is used in preparing data tapes for DIAK, FFOC, and LSA.

Table 2. KONPACT Data Tapes

TAPE NAME	DESCRIPTION	GENERATING PROGRAM	BENEFITING PROGRAM(S)
VDATA	Simulator Interface data in the form of card images	LSA	KONPACT-1
QDATA	Quadruple (A, B, C, D) or state variable representation data	KONPACT-1	KONPACT-1 KONPACT-2
NDATA	Name list data of the state variable representation	KONPACT-1	KONPACT-1
DDATA	Full state feedback gain data in the form of card images	KONPACT-2	KONPACT-2
FDATA	Reduced feedback gain data in the form of card images	KONPACT-2	KONPACT-2
SDSTP	Frequency domain representation of quadruple data	KONPACT-2	LSA

VARIABLE DIMENSIONING

Variable dimensioning (dynamic data storage) techniques (Reference 4) are used for efficient data storage. This technique also facilitates changing the amount of allocated (required) storage space by a data card input.

In KONPACT the subprogram arrays, whose size depend on the maximum

system dimension inputs, are stored in scratch storage blocks using variable entry points. In the subprograms the arrays are dimensioned with integer variables. These "variable DIMENSION statements" remain unchanged although the amount of required data storage is altered. The maximum size of the scratch storage blocks is specified, in a "fixed DIMENSION statement," in the main program.

The size of storage actually needed by the arrays varies depending on the maximum system dimension inputs. Thus, if the maximum size a user allows for his program changes, there are only the "fixed DIMENSION statements," in the main program, to be changed. Changing the main program of KONPACT-1 is done by a precompiler, as discussed in Section V. The user provides the new maximum system dimensions by data cards. Updating and running with the updated main program are done with control cards in a single run.

In KONPACT programs, four scratch storage blocks, namely S1, S2, S3, and S4 are used. These are specified in the MAIN program of main overlay in labeled COMMON statements under SC1, SC2, SC3, and SC4, respectively. The maximum sizes of these scratch storage blocks are defined there.

The main programs in the primary overlays perform four specific tasks of variable dimensioning. A primary overlay main program first defines the scratch storage blocks under labeled COMMON statements as follows:

```
COMMON/SC1/S1(1)
```

```
COMMON/SC2/S2(1)
```

```
COMMON/SC3/S3(1)
```

```
COMMON/SC4/S4(1)
```

Second, it calculates the start indexes (N1, N2, ... etc.) of the scratch arrays for the stored data as shown in Table 3. Third, it checks the total length occupied by the arrays against the size of the allocated scratch storage blocks. Fourth, it passes the start indexes of the arrays to the subprograms.

Table 3. Typical Dynamic Storage Map

Storage Block	Arrays	Block Addresses
S1 (N1)	V(MAXN)	$N1 = 1$
S1 (N2)	W(MAXM)	$N2 = N1 + MAXN$
S1 (N3)	F(MAXN, MAXM)	$N3 = N2 + MAXM$
S1 (N4)	U(NUM)	$N4 = N3 + MAXN * MAXM$

SECTION III

MODELING PROGRAM (KONPACT-1)

KONPACT-1 interfaces with FLEXSTAB through the LSA program to obtain the unaugmented vehicle model. It augments this model with the specified dynamics (actuator, sensor, controller, gust, etc.) to obtain the state space description (quadruple data) of the overall flight control system for design.

In this section, a description of the KONPACT-1 program is presented in terms of overlay structure, flow charts, and program listings.

OVERLAY STRUCTURE

The KONPACT-1 program consists of a main overlay and five primary overlays. The overlay structure and the subroutines in each overlay are given in Figure 4. The subroutine summary consisting of name, description, reference, overlay position, and interrelationship is given in Table 4.

DESCRIPTION OF MAIN PROGRAMS

Program MAIN

This is the main program for overlay (0,0). This program assigns the various file numbers used in KONPACT-1. Maximum system dimensions

Table 4. KONPACT-1 Subroutine Summary

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship	
					Calls	Called by
MAIN	Sets up system dimensions and scratch array dimensions.		5	0,0	KORGI	
KORGI	Organizes input data and calls the primary overlays.	11	12	0,0	FILE IDPR IDRO	MAIN
NAMEL	Reads, prints and updates name list data for the systems.	13	14	0,0	FILE HPR	STAMK1 STAMK2 STAMK3 QUADK STAMK4
QDIO	Reads and writes quadruple data.	57	58	0,0	MPRS FILE	STAMK1 STAMK2 STAMK3 QUADK STAMK4 HESPK
IDRC	Reorganizes the input data.	59	60	0,0		KORGI
FILE	Locates and inserts system labels on disc files and writes end of data mark on the disc files.	65	66	0,0		KORGI NAMEL QDIO SIMK MNAME
TPR	Prints transfer function data.	67	68	0,0		SIMKT
HPR	Prints heading for the system name.		69	0,0		NAMEL STAMK1 STAMK2 STAMK3 STAMK4 MNAME

Table 4. KONPACT-1 Subroutine Summary (Continued)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship	
					Calls	Called by
IDPR	Prints input data.		70	0.0		KORG1
MPRS	Prints matrix data on line printer.		71	0.0		QDIO STAMK1 STAMK2 SIMKT STAMK3 SIMK STAMK4 IMRATE REDUCE
ZERO	Initializes (or zeros) the elements of matrices.		73	0.0		QUADK SIMK
INPT	Reads non zero elements of a matrix.		74	0.0		SIMKT QUADK SIMK
DEBUG	Prints a debugging message.		76	0.0		STAMK1 SIMK1 MAIN2 STAMK2 SIMKT DFN PHERR TRANSK MAIN5 RESPK MNAME RSDRD SDED SHIFT
ERRM	Prints error message.		77	0.0		RESPK MNAME RSDRD SDED

Table 4. KONPACT-1 Subroutine Summary (Continued)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship	
					Calls	Called by
DERRM	Prints an error message when the dimensions for scratch arrays are not sufficient.		78	0.0		MAIN1 STAMK1 SIMK1 MAIN2 STAMK2 MAIN3 STAMK3 MAIN4 STAMK4 MAIN5
DERRMS	Prints an error message when the system dimensions are not sufficient.		79	0.0		STAMK1 STAMK2 STAMK3 STAMK4
TDINVR	Inverts a non-singular matrix or solves a set of linear equations.		81	0.0		STAMK1 STAMK2 STAMK3 STAMK4 IMRATE REDUCT.
MAIN1	Sets up block addresses and checks if scratch array size is sufficient.		6	1.0	DERRM STAMK1	
STAMK1	Obtains state space model from LSA simulator deck data and load equation data (implemented in SIMK1 subroutine).	15	16	1.0		MAIN1 DERRM TDINVR MPRS NAMEL DEBUG QDIO HPR
SIMK1	Reads simulator deck data and load equation data and implements them into simulation equations.	17	18	1.0		STAMK1 DERRM INPTI MPRS1
MPRS1	Prints simulator deck data and load equation data.		72	1.0		SIMK1

Table 4. KONPACT-1 Subroutine Summary (Continued)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship	
					Calls	Called by
INPT1	Reads simulator deck data and Load equation data.		75	1,0		SIMK1
MAIN2	Sets up block addresses and checks if scratch array size is sufficient.		7	2,0	DERRM DEBEG STAMK2	
STAMK2	Obtains state space model from Transfer function data and connection data (implemented in SIMKT subroutine).	19	20	2,0	SIMKT DERRM TDINVR DEBEG MPRS HPR QDIO DERRMS NAMEL	MAIN2
SIMKT	Reads transfer function data and Connection data and implements them into simulation equations.	21	22	2,0	DEBEG TPR DFN PHERR TRANSK INPT MPRS	STAMK2
TRANSK	Computes state space model for rational transfer functions of up to 5th order.	23	24	2,0	DEBEG	SIMKT
DFN	Selects the specified pade approximation to transport (time) delay from a table of pade approximations.	25	26	2,0	DEBEG	SIMKT
PHERR	Computes the phase error of pade approximation to transport (time) delay.	27	28	2,0	DEBEG	SIMKT
MAIN3	Sets up block addresses and checks if scratch array size is sufficient.		8	3,0	DERRM STAMK3	

Table 4. KONPACT-1 Subroutine Summary (Continued)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship	
					Calls	Called by
STAMK3	Obtains state space model from state space data of subsystems and interconnection data (implemented in SIMK subroutine).	29	30	3.0	SIMK TDINVR DERRM NAMEL QDIO QUADK HPR DERRMS MPRS	MAIN3
SIMK	Reads state space data of subsystems and interconnection data and implements them into simulation equations.	31	32	3.0	ZERO INPUT MPRS FILE	STAMK3
QUADK	Reads directly the state space data for the system.	33	34	3.0	NAMEL QDIO	STAMK3
MAIN4	Sets up block addresses and checks if scratch array size is sufficient.		9	4.0	DERRM STAMK4	
STAMK4	Obtains state space model for the ALDCS controller (implemented in SIMK2 subroutine).	35	36	4.0	SIMK2 DERRM DERRMS TDINVR HPR MPRS NAMEL QDIO	MAIN4
SIMK2	Reads ALDCS controller gains and switch modes and implements ALDCS controller into simulation equations.	37	38	4.0		STAMK4
MAIN5	Sets up block addresses and checks if scratch array size is sufficient.		10	5.0	DERUG DERRM CCNDK	

Table 4. KONPACT-1 Subroutine Summary (Continued)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship	
					Calls	Called by
CONDK	Modifies state space data and name list data according to the design specifications.	39	40	5.0	MNAME QDIO DEBUG SDRD SCAL ERRM DIFFK REDUCE SHUFF RSDRD	MAINS
MNAME	Reads, modifies, and prints the name list data for a system.	41	42	5.0	ERRM DEBUG SHIFT HPR FILE	CONDK
IMRATE	Obtains the Implicit model error rates and truncates the Implicit model.	43	44	5.0	TDINVR MPRS	CONDK
DIFFK	Differentiates either a specified response or state of a system.	45	46	5.0		CONDK
REDUCE	Residualizes or truncates the state space data of a system.	47	48	5.0	TDINVR MPRS	CONDK
SCAL	Computes scaled state space data.	49	50	5.0		CONDK
SHUFF	Shuffles the states space data and name list data for a system.	51	52	5.0	SHUF1 SHUF2	CONDK
SHUF1	Shuffles the specified rows and columns of a matrix.	53	54	5.0		SHUFF
SHUF2	Shuffles the name list data arrays.	55	56	5.0		SHUFF

Table 4. KONPACT-1 Subroutine Summary (Concluded)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship	
					Calls	Called by
RSDRD	Reads residualization, truncation, and shuffling data.	61	62	5.0	DEBUG ERRM	CONDK
SDRD	Reads scaling data.	63	64	5.0	DEBUG ERRM	CONDE
SHIFT	Shifts the contents of old name list arrays into new name list arrays.		80	5.0	DEBUG	MINAME

and scratch array dimensions are set in this program. The program calls the organizing subroutine KORGI. The program listing is given in Figure 5.

Program MAIN1

This is the main program for overlay (1,0). This program computes the required scratch array dimensions as explained in Section II, and checks if the scratch array sizes are sufficient. The program calls the state modeling subroutine STAMK1. The program listing is given in Figure 6. The dynamic storage map is given in Table 5.

Program MAIN2

This is the main program for overlay (2,0). This program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the state modeling subroutine STAMK2. The program listing is given in Figure 7. The dynamic storage map is given in Table 6.

Program MAIN3

This is the main program for overlay (3,0). This program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the state modeling subroutine STAMK3. The program listing is given in Figure 8. The dynamic storage map is given in Table 7.

Table 5. Dynamic Storage Map for Program MAIN1

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S1(N1)	N1 = 1	V(MAXN)	MAXN = NXM + NYM + NRM	Calculated in KORGI
S1(N2)	N2 = N1 + MAXN	W(MAXM)	MAXM = NXM * 2 + NYM + NUM	Calculated in KORGI
S1(N3)	N3 = N2 + MAXM	F(MAXN, MAXM)		
S1(N4)	N4 = N3 + MAXN * MAXM	U(NUM)	NUM	Defined in MAIN
S2(M1)	M1 = 1	A(NXM, NXM)	NXM	Defined in MAIN
S2(M2)	M2 = M1 + NXM * NXM	B(NXM, NUM)		
S2(N3)	M3 = M2 + NXM * NUM	C(NRM, NXM)		
S2(M4)	M4 = M3 + NRM * NXM	D(NRM, NUM)		
S3(L1)	L1 = 1	NNS(NXM)		
S3(L2)	L2 = L1 + NXM	VNS(NXM, 2)		
S3(L3)	L3 = L2 + NXM * 2	DES(NXM, 10)		
S3(L4)	L4 = L3 + NXM * 10	UNITS(NXM, 4)		
S3(L5)	L5 = L4 + NXM * 4	NNO(NRM)		
S3(L6)	L6 = L5 + NRM	VNO(NRM, 2)		
S3(L7)	L7 = L6 + NRM * 2	DESONRM, 10)		
S3(L8)	L8 = L7 + NRM * 10	UNITO(NRM, 4)		
S3(L9)	L9 = L8 + NRM * 4	NNI(NUM)		
S3(L10)	L10 = L9 + NUM	VNI(NUM, 2)		
S3(L11)	L11 = L10 + NUM * 2	DESI(NUM, 10)		
S3(L12)	L12 = L11 + NUM * 10	UNITI(NUM, 4)		

Table 6. Dynamic Storage Map for Program MAIN2

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S1(N1)	N1 = 1	V(MAXN)	MAXN = NXM + NYM + NRM	Calculated in KORG1
S1(N2)	N2 = N1 + MAXN	W(MAXM)	MAXM = NXM 2 + NYM + NUM	Calculated in KORG1
S1(N3)	N3 = N2 + MAXM	F(MAXN, MAXM)		
S1(N4)	N4 = N3 + MAXN + MAXM	X(DOT(MST, MTFB))	MST	Defined in KORG1
S1(N5)	N5 = N4 + MST*MTFB	X(MST, MTFB)	MTFB	Defined in MAIN
S1(N6)	N6 = N5 + MST*MTFB	R(MTFB)		
S1(N7)	N7 = N6 + MTFB	U(MTFB)		
S1(N8)	N8 = N7 + MTFB	U(NUM)	NUM	Defined in MAIN
S1(N9)	N9 = N8 + NUM	NNX(MTFB)		
S1(N10)	N10 = N9 + MTFB	NNR(MTFB)		
S1(N11)	N11 = N10 + MTFB	NNU(MTFB)		
S2(M1)	M1 = 1	A(NXM, NXM)	NXM	Defined in MAIN
S2(M2)	M2 = M1 + NXM * NXM	B(NXM, NUM)		
S2(M3)	M3 = M2 + NXM * NUM	C(NRM, NXM)		
S2(M4)	M4 = M3 + NRM * NXM	D(NRM, NUM)	NRM	Defined in MAIN
S3(L1)	L1 = 1	NNS(NXM)		
S3(L2)	L2 = L1 + NXM	VNS(NXM, 2)		
S3(L3)	L3 = L2 + NXM * 2	DESS(NXM, 10)		
S3(L4)	L4 = L3 + NXM * 10	UNITS(NXM, 4)		
S3(L5)	L5 = L4 + NXM * 4	NNO(NRM)		
S3(L6)	L6 = L5 + NRM	VNO(NRM, 2)		
S3(L7)	L7 = L6 + NRM * 2	DESO(NRM, 10)		
S3(L8)	L8 = L7 + NRM * 10	UNTO(NRM, 4)		
S3(L9)	L9 = L8 + NRM * 4	NNI(NUM)		
S3(L10)	L10 = L9 + NUM	VNI(NUM, 2)		
S3(L11)	L11 = L10 + NUM * 2	DESI(NUM, 10)		
S3(L12)	L12 = L11 + NUM * 10	UNITI(NUM, 4)		

Table 7. Dynamic Storage Map for Program MAIN3

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S1(N1)	N1 = 1	V(MAXN)	MAXN = NXM + NYM + NRM	Calculated in KORG1
S1(N2)	N2 = N1 + MAXN	W(MAXM)	MAXM = NXM*2 + NYM + NUM	Calculated in KORG1
S1(N3)	N3 = N2 + MAXM	F(MAXN, MAXM)		
S1(N4)	N4 = N3 + MAXN*MAXM	XDOT(NXM, MB)		Defined in MAIN
S1(N5)	N5 = N4 + NXM*MB	X(NXM, MB)		Defined in MAIN
S1(N6)	N6 = N5 + NXM*MB	R(NRM, MB)		Defined in MAIN
S1(N7)	N7 = N6 + NRM*MB	U(NUM, MB)		Defined in MAIN
S1(N8)	N8 = N7 + NUM*MB	U(NUM)		
S1(N9)	N9 = N8 + NUM	RIN(NRMMB)		
S1(N10)	N10 = N9 + NRMMB	NNX(MB)	NRMMB = NRM*MB	Calculated in MAIN3
S1(N11)	N11 = N10 + MB	NNR(MB)		
S1(N12)	N12 = N11 + MB	NNU(MB)		
S2(M1)	M1 = 1	A(NXM, NXM)		
S2(M2)	M2 = M1 + NXM*NXM	B(NXM, NUM)		
S2(M3)	M3 = M2 + NXM*NUM	C(NRM, NXM)		
S2(M4)	M4 = M3 + NRM*NXM	D(NRM, NUM)		
S3(L1)	L1 = 1	NNS(NXM)		
S3(L2)	L2 = L1 + NXM	VNS(NXM, 2)		
S3(L3)	L3 = L2 + NXM*2	DESS(NXM, 10)		
S3(L4)	L4 = L3 + NXM*10	UNITS(NXM, 4)		
S3(L5)	L5 = L4 + NXM*4	NNO(NRM)		
S3(L6)	L6 = L5 + NRM	VNO(NRM, 2)		
S3(L7)	L7 = L6 + NRM*2	DESO(NRM, 10)		
S3(L8)	L8 = L7 + NRM*10	UNITO(NRM, 4)		
S3(L9)	L9 = L8 + NRM*4	NNI(NUM)		
S3(L10)	L10 = L9 + NUM	VNI(NUM, 2)		
S3(L11)	L11 = L10 + NUM*2	DESI(NUM, 10)		
S3(L12)	L12 = L11 + NUM*10	UNITI(NUM, 4)		

Program MAIN4

This is the main program for overlay (4,0). This program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the state modeling subroutine STAMK4. The program listing is given in Figure 9. The dynamic storage map is given in Table 8.

Program MAIN5

This is the main program for overlay (5,0). This program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the conditioning subroutine CONDK. The program listing is given in Figure 10. The dynamic storage map is given in Table 9.

DESCRIPTION OF BASIC SUBROUTINES

Subroutine KORG1

This subroutine organizes the execution of KONPACT-1 program. The input data cards for KONPACT-1 program are read and printed by this subroutine. The print specification cards are read in this subroutine and the print control parameter IPRINT is set for the printer output options of KONPACT-1 program. The flow chart is given in Figure 11 and the program listing is given in Figure 12.

Table 8. Dynamic Storage Map for Program MAIN4

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S1(N1)	N1 = 1	V(MAXN)	MAXN = NXM + NYM + NRM	Calculated in KORGI
S1(N2)	N2 = N1 + MAXN	W(MAXM)	MAXM = NXM*2 + NYM + NUM	Calculated in KORGI
S1(N3)	N3 = N2 + MAXM	F(MAXN, MAXM)		
S1(N4)	N4 = N3 + MAXN MAXM	U(NUM)	NUM	Defined in MAIN
S2(M1)	M1 = 1	A(NXM, NXM)	NXM	Defined in MAIN
S2(M2)	M2 = M1 + NXM NXM	B(NXM, NUM)		
S2(M3)	M3 = M2 + NXM*NUM	C(NRM, NXM)		
S2(M4)	M4 = M3 + NRM NXM	D(NRM, NUM)	NRM	Defined in MAIN
S3(L1)	L1 = 1	NNS(NXM)		
S3(L2)	L2 = L1 + NXM	VNS(NXM, 2)		
S3(L3)	L3 = L2 + NXM*2	DESS(NXM, 10)		
S3(L4)	L4 = L3 + NXM*10	UNITS(NXM, 4)		
S3(L5)	L5 = L4 + NXM*4	NNO(NRM)		
S3(L6)	L6 = N5 + NRM	VNO(NRM, 2)		
S3(L7)	L7, L6 + NRM*2	DESO(NRM, 10)		
S3(L8)	L8 = L7 + NRM*10	UNITO(NRM, 4)		
S3(L9)	L9 = L8 + NRM*4	NN(NUM)		
S3(L10)	L10 = L9 + NUM	VNI(NUM, 2)		
S3(L11)	L11 = L10 + NUM*2	DES(NUM, 10)		
S3(L12)	L12 = L11 + NUM*10	UNITI(NUM, 4)		

Table 9. Dynamic Storage Map for Program MAIN5

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S1(L1)	L1 = 1	DUMMY1(NDM11, NDM12)	NDM11 = MAX(17, NXM, NRM)	Calculated in MAIN5
S1(L2)	L2 = L1 + NDM11 * NDM12	DUMMY2(NDM21, NDM22)	NDM12 = MAX(NXM + NUM, NRM)	Calculated in MAIN5
S1(L3)	L3 = L2 + NDM21 * NDM22	DUMMY3(NUM)	NDM21 = MAX(NRM, NXM)	Calculated in MAIN5
S1(L4)	L4 = L3 + NUM	ES(NXM, NUM)	NDM22 = MAX(NXM, NRM, NUM)	Calculated in MAIN5
S1(L5)	L5 = L4 + NXM * NUM	ER(NRM, NUM)	NRM	Defined in MAIN
S1(L6)	L6 = L5 + NRM * NUM	NSHUF5(NXM)	NXM	Defined in MAIN
S1(L7)	L7 = L6 + NXM	NSHUF6(NRM)		
S1(L8)	L8 = L7 + NRM	NSHUF7(NUM)	NUM	Defined in MAIN
S1(L9)	L9 = L8 + NUM	CS(NRM, NXM)		
S1(L10)	L10 = L9 + NRM * NXM	DS(NRM, NUM)		
S1(L11)	L11 = L10 + NRM * NUM	CW(NRM, NXM)		
S1(L12)	L12 = L11 + NRM * NXM	DW(NRM, NUM)		
S1(L13)	L13 = L12 + NRM * NUM	IRS(NRM)		
S1(L14)	L14 = L13 + NRM	Q(NRM, NRM)		
S2(M1)	M1 = 1	A(NXM, NXM)		
S2(M2)	M2 = M1 + NXM * NXM	B(NXM, NUM)		
S2(M3)	M3 = M2 + NXM * NUM	C(NRM, NXM)		
S2(M4)	M4 = M3 + NRM * NXM	D(NRM, NUM)		
S2(M5)	M5 = M4 + NRM * NUM	CM(NRM, NXM)		
S2(M6)	M6 = M5 + NRM * NXM	DM(NRM, NUM)		
S3(N1)	N1 = 1	NNS(NXM)		
S3(N2)	N2 = N1 + NXM	VNS(NXM, 2)		
S3(N3)	N3 = N2 + NXM * 2	DESS(NXM, 10)		
S3(N4)	N4 = N3 + NXM * 10	UNITS(NXM, 4)		

Table 9. Dynamic Storage Map for Program MAIN5 (Concluded)

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S3(N5)	N5 = N4 + NXM*4	NNO(NRM)		
S3(N6)	N6 = N5 + NRM	VNO(NRM, 2)		
S3(N7)	N7 = N6 + NRM*2	DESO(NRM, 10)		
S3(N8)	N8 = N7 + NRM*10	UNITO(NRM, 4)		
S3(N9)	N9 = N8 + NRM*4	NNI(NUM)		
S3(N10)	N10 = N9 + NUM	VNI(NUM, 2)		
S3(N11)	N11 = N10 + NUM*2	DESINUM, 10)		
S3(N12)	N12 = N11 + NUM*10	UNITI(NUM, 4)		
S3(N13)	N13 = N12 + NUM*4	VNNS(NXM)		
S3(N14)	N14 = N13 + NXM	VNNS(NXM, 2)		
S3(N15)	N15 = N14 + NXM*2	DESNS(NXM, 10)		
S3(N16)	N16 = N15 + NXM*10	UNITNS(NXM, 4)		
S3(N17)	N17 = N16 + NXM*4	NNNO(NRM)		
S3(N18)	N18 = N17 + NRM	VVNO(NRM, 2)		
S3(N19)	N19 = N18 + NRM*2	DESNO(NRM, 10)		
S3(N20)	N20 = N19 + NRM*10	UNITNO(NRM, 4)		
S3(N21)	N21 = N20 + NRM*4	NNNI(NUM)		
S3(N22)	N22 = N21 + NUM	VNNI(NUM, 2)		
S3(N23)	N23 = N22 + NUM*2	DESINIUM, 10)		
S3(N24)	N24 = N23 + NUM*10	UNITNI(NUM, 4)		

Subroutine NAMEL

This subroutine obtains the name list data for the system variables. The subroutine either reads the name list data from input data cards or internally obtains a default name list data. In the case of combining various subsystems into an overall system, the subroutine uses the interconnection data to obtain the appropriate name list data. This subroutine also writes the name list data of each system on NDATA file for use by other programs. The flow chart is given in Figure 13 and the program listing is given in Figure 14.

Subroutine STAMK1

This subroutine obtains the state space model (quadruple data) of the system implemented in subroutine SIMK1. The flow chart is given in Figure 15 and the program listing is given in Figure 16. The dynamic storage map is given in Table 10.

Subroutine SIMK1

This subroutine reads simulator deck data and load equation data obtained by the Linear System Analysis (LSA) program and implements them into simulation equations. The flow chart is given in Figure 17 and the program listing is given in Figure 18.

Subroutine STAMK2

This subroutine obtains the state space model (quadruple data) of the system implemented in subroutine SIMKT. The flow chart is given in

Table 10. Dynamic Storage Map for Program STAMK1

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
W(N1)	N1 = 1	XDOT(NX)	NX	Calculated in SIMK1
W(N2)	N2 = N1 + NX	Y(NY)	NY	Calculated in SIMK1
W(N3)	N3 = N2 + NY	X(NX)		
W(N4)	N4 = N3 + NX	U(NU)	NU	Calculated in SIMK1
V(N1)	N1 = 1	XDOTL(NX)		
V(N2)	N2 = N1 + NX	YL(NY)		
V(N3)	N3 = N2 + NY	RL(NR)	NR	Calculated in SIMK1
S1(L1)	L1 = 1	DESS(NXM, 10, MB)	NXM, MB	Defined in MAIN
S1(L2)	L2 = L1 + NXM*10*MB	UNITSS(NXM, 4, MB)		
S1(L3)	L3 = L2 + NXM*4*MB	DESOO(NRM, 10, MB)	NRM	Defined in MAIN
S1(L4)	L4 = L3 + NRM*10*MB	UNITOO(NRM, 4, MB)		
S1(L5)	L5 = L4 + NRM*4*MB	DESI(NUM, 10, MB)	NUM	Defined in MAIN
S1(L6)	L6 = L5 + NUM*10*MB	UNITII(NUM, 4, MB)		
S1(L7)	L7 = L6 + NUM*4*MB	NXX(MB)		
S1(L8)	L8 = L7 + MB	NRR(MB)		
S1(L9)	L9 = L8 + MB	NUU(MB)		

Figure 19 and the program listing is given in Figure 20. The dynamic storage map is given in Table 11.

Subroutine SIMKT

This subroutine reads transfer function data and connection data and implements them into simulation equations. The flow chart is given in Figure 21 and the program listing is given in Figure 22.

Subroutine TRANSK

This subroutine computes the state space model for rational transfer functions using the input Frobenius form of realization. The flow chart is given in Figure 23 and the program listing is given in Figure 24.

Subroutine DFN

This subroutine selects the specified Pade approximation to transport (time) delay from a table of Pade approximations. The flow chart is given in Figure 25 and the program listing is given in Figure 26.

Subroutine PHERR

This subroutine computes the phase error of the Pade approximation to transport (time) delay. The flow chart is given in Figure 27 and the program listing is given in Figure 28.

Table 11. Dynamic Storage Map for Program STAMK2

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
V(N1)	N1 = 1	XDOL(NX)	NX	Calculated in SIMKT
V(N2)	N2 = N1 + NX	YL(NY)	NY	Calculated in SIMKT
V(N3)	N3 = N2 + NY	RL(NR)	NR	Calculated in SIMKT
S1(L1)	L1 = 1	DESS(NXM, 10, MB)	NXM, MB	Defined in MAIN
S1(L2)	L2 = L1 + NXM*10*MB	UNITSS(NXM, 4, MB)		
S1(L3)	L3 = L2 + NXM*4*MB	DESOO(NRM, 10, MB)	NRM	Defined in MAIN
S1(L4)	L4 = L3 + NRM*10*MB	UNITOO(NRM, 4, MB)		
S1(L5)	L5 = L4 + NRM*4*MB	DESII(NUM, 10, MB)	NUM	Defined in MAIN
S1(L6)	L6 = L5 + NUM*10*MB	UNITII(NUM, 4, MB)		
S1(L7)	L7 = L6 + NUM*4*MB	NXX(MB)		
S1(L8)	L8 = L7 + MB	NRR(MB)		
S1(L7)	L9 = L8 + MB	NUU(MB)		

Subroutine STAMK3

This subroutine obtains the state space model (quadruple data) of the system implemented in subroutine SIMK. The flow chart is given in Figure 29 and the program listing is given in Figure 30. The dynamic storage map is given in Table 12.

Subroutine SIMK

This subroutine reads interconnection data and state space data for subsystems and implements the simulation equations for the overall system. SIMK also writes the interconnection data on the scratch file for use by subroutine NAMEL. The flow chart is given in Figure 31 and the program listing is given in Figure 32.

Subroutine QUADK

This subroutine reads directly the state space data for the system. The flow chart is given in Figure 33 and the program listing is given in Figure 34.

Subroutine STAMK4

This subroutine obtains the state space model (quadruple data) of the system implemented in subroutine SIMK2. The flow chart is given in Figure 35 and the program listing is given in Figure 36. The dynamic storage map is given in Table 13.

Table 12. Dynamic Storage Map for Program STAMK3

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
V(N1)	N1 = 1	XDOTL(NX)	NX	Calculated in SIMK
V(N2)	N2 = N1 + NX	YL(NY)	NY	Calculated in SIMK
V(N3)	N3 = N2 + NY	RL(NR)	NR	Calculated in SIMK
S1(L1)	L1 = 1	DESS(NXM, 10, MB)	NXM, MB	Defined in MAIN
S1(L2)	L2 = L1 + NXM*10*MB	UNITSS(NXM, 4, MB)		
S1(L3)	L3 = L2 + NXM*4*MB	DESOO(NRM, 10, MB)	NRM	Defined in MAIN
S1(L4)	L4 = L3 + NRM*10*MB	UNITI(NRM, 4, MB)		
S1(L5)	L5 = L4 + NRM*4*MB	DESII(NUM, 10, MB)	NUM	Defined in MAIN
S1(L6)	L6 = L5 + NUM*10*MB	UNITII(NUM, 4, MB)		
S1(L7)	L7 = L6 + NUM*4*MB	NXX(MB)		
S1(L8)	L8 = L7 + MB	NRR(MB)		
S1(L9)	L9 = L8 + MB	NUU(MB)		
S2(M1)	M1 = 1	A T(NXM, NXM, MB)		
S2(M2)	M2 = M1 + NXM*NXM*MB	BT(NXM, NUM, MB)		
S2(M3)	M3 = M2 + NXM*NUM*MB	CT(NRM, NXM, MB)		
S2(M4)	M4 = M3 + NRM*NXM*MB	DT(NRM, NUM, MB)		
S2(M5)	M5 = M4 + NRM*NUM*MB	P(MN, MN)	MN=MM*MB	Calculated in KORGI

Table 12. Dynamic Storage Map for Program STAMK3 (Concluded)

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S2(M6)	M6 = M5 + MN*MM	Q(MN, NUM)		
S2(M7)	M7 = M6 + MN*NUM	R(NRM, MN)		
S2(M8)	M8 = M7 + NRM*MM	S(NRM, NUM)		
S3(K1)	K1 = 1	PP(MP, MM, MM)	MN = MAX (NRM, NUM)	Calculated in KORGI
S3(K2)	K2 = K1 + MP*MM*MM	QQ(MQ, MM, NUM)	MQ = MB	Calculated in KORGI
S3(K3)	K3 = K2 + MQ*MM*NUM	RR(MR, NRM, MM)	MR = MB	Calculated in KORGI
S3(K4)	K4 = K3 + MR*NRM*MM	NSP(MP)	MP = MB*2	
S3(K5)	K5 = K4 + MP	NSQ(MQ)		
S3(K6)	K6 = K5 + MQ	NSR(MR)		

Table 13. Dynamic Storage Map for Program STAMK4

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
W(N1)	N1 = 1	XDOT(NX)	NX	Calculated in SIMK2
W(N2)	N2 = N1 + NX	Y(NY)	NY	Calculated in SIMK2
W(N3)	N3 = N2 + NY	X(NX)		
W(N4)	N4 = N3 + NX	U(NU)	PU	Calculated in SIMK2
V(N1)	N1 = 1	XDOTL(NX)		
V(N2)	N2 = N1 + NX	YL(NY)		
V(N3)	N3 = N2 + NY	RL(NR)	NR	Calculated in SIMK2
S1(L1)	L1 = 1	DESS(NXM, 10, MB)	NXM, MB	Defined in MAIN
S1(L2)	L2 = L1 + NXM*10*MB	UNITSS(NXM, 4, MB)		
S1(L3)	L3 = L2 + NXM*4*MB	DES00(NRM, 10, MB)	NRM	Defined in MAIN
S1(L4)	L4 = L3 + NRM*10*MB	UNIT00(NRM, 4, MB)		
S1(L5)	L5 = L4 + NRM*4*MB	DESI(NUM, 10, MB)	NUM	Defined in MAIN
S1(L6)	L6 = L5 + NUM*10*MB	UNITII(NUM, 4, ME)		
S1(L7)	L7 = L6 + NUM*4*MB	NXX(MB)		
S1(L8)	L8 = L7 + MB	NRR(MB)		
S1(L9)	L9 = L8 + MB	NUU(MB)		

Subroutine SIMK2

This is a users written subroutine. Here it is written for the ALDCS controller. It reads ALDCS controller gains and switch modes (positions) and implements the controller into simulation equations. The flow chart is given in Figure 37 and the program listing is given in Figure 38.

Subroutine CONDK

This subroutine organizes the modification (conditioning) of the state space data and name list data according to specification. The flow chart is given in Figure 39 and the program listing is given in Figure 40.

Subroutine MNAME

This subroutine modifies the name list data of the system according to the conditioning data. The flow chart is given in Figure 41 and the program listing is given in Figure 42.

Subroutine IMRATE

This subroutine obtains the implicit model error rates and truncates the implicit model. The flow chart is given in Figure 43 and the program listing is given in Figure 44.

Subroutine DIFFK

This subroutine obtains the rate of change of either a specified response or state of the system by differentiation. If the differentiation requires external rate inputs in the model, a message is printed by the subroutine. The flow chart is given in Figure 45 and the program listing is given in Figure 46.

Subroutine REDUCE

This subroutine residualizes or truncates the state space data of the system. In addition it computes the error of residualization. The flow chart is given in Figure 47 and the program listing is given in Figure 48.

Subroutine SCAL

This subroutine computes the scaled state space data. The flow chart is given in Figure 49 and the program listing is given in Figure 50.

Subroutine SHUFF

This subroutine shuffles the state space data and the name list data by calling subroutines SHUF1 and SHUF2. The flow chart is given in Figure 51 and the program listing is given in Figure 52.

Subroutine SHUF1

This subroutine shuffles the rows and columns of a matrix. The flow chart is given in Figure 53 and the program listing is given in Figure 54.

Subroutine SHUF2

This subroutine shuffles the name list data arrays. The flow chart is given in Figure 55 and the program listing is given in Figure 56.

DESCRIPTION OF AUXILIARY SUBROUTINES

Subroutine QDIO

This subroutine reads the state space data from file QDATA and prints it. It also writes the state space data on file QDATA. The flow chart is given in Figure 57 and the program listing is given in Figure 58.

Subroutine IDRO

This subroutine reorganizes the input data. The reorganized input data is written on file BINPUT. The flow chart is given in Figure 59 and the program listing is given in Figure 60.

Subroutine RSDRD

This subroutine reads residualization, truncation, and shuffling data for the variables of the system. The flow chart is given in Figure 61 and the program listing is given in Figure 62.

Subroutine SDRD

This subroutine reads the scaling factor and the new units for the system variables. The flow chart is given in Figure 63 and the program listing is given in Figure 64.

Subroutine FILE

This subroutine positions the data file for reading or writing data. There are three modes of calling this subroutine. INSERT mode inserts the label name and positions the data file for writing. LOCATE mode locates the label name and positions the data file for reading. NULL mode removes the label name from the data file. The flow chart is given in Figure 65 and the program listing is given in Figure 66.

Subroutine TPR

This subroutine prints transfer function data. The flow chart is given in Figure 67 and the program listing is given in Figure 68.

Subroutine HPR

This subroutine prints the headings for the system label name. The program listing is given in Figure 69.

Subroutine IDPR

This subroutines prints the input data. The program listing is given in Figure 70.

Subroutine MPRS

This subroutine prints a matrix, identifying the rows and columns. The program listing is given in Figure 71.

Subroutine MPRS1

This subroutine prints the simulator interface matrix data from the Linear System Analysis (LSA) program. The program listing is given in Figure 72.

Subroutine ZERO

This subroutine initializes (or zeros) the elements of a matrix. The program listing is given in Figure 73.

Subroutine INPT

This subroutine reads the nonzero elements of a matrix. The program listing is given in Figure 74.

Subroutine INPT1

This subroutine reads the simulator interface matrix data from Linear System Analysis (LSA) program. The program listing is given in Figure 75.

Subroutine DEBUG

This subroutine prints a debugging message. The program listing is given in Figure 76.

Subroutine ERRM

This subroutine prints an error message indicating the program and overlay at which the error was detected. The program listing is given in Figure 77.

Subroutine DERRM

This subroutine prints a message when the maximum dimensions for scratch arrays are not sufficient. The program listing is given in Figure 78.

Subroutine DERRMS

This subroutine prints a message when the Maximum System dimensions are not sufficient. The program listing is given in Figure 79.

Subroutine SHIFT

This subroutine shifts the contents of old name list arrays into new name list arrays. The program listing is given in Figure 80.

Subroutine TDINVR

This subroutine inverts a non-singular matrix or solves a set of linear equations. The program listing is given in Figure 81.

SECTION IV

DESIGN PROGRAM (KONPACT-2)

The data produced by KONPACT-1 are utilized by KONPACT-2. KONPACT-2 contains two Air Force-owned synthesis programs, DIAK and FFOC. The DIAK (Doubly Iterative Algorithm developed by Konar) program computes optimal controller gains for full state feedback. FFOC (Fixed Form Optimal Control) simplifies these gains to specified measurements. KONPACT-2 interfaces with FLEXSTAB through the LSA program to evaluate performances of the closed loop system.

In this section, a description of KONPACT-2 program is presented in terms of overlay structure, flow charts, and program listings. The DIAK and FFOC programs are fully documented in Reference 2 and only the program listings are given here for completeness. Modularization and variable dimensioning of DIAK and FFOC programs are beyond the scope of this contract.

OVERLAY STRUCTURE

The KONPACT-2 program consists of a main overlay and three primary overlays. The overlay structure and the subroutines in each overlay is given in Figure 82. The subroutine summary consisting of name, description, reference, overlay position, and interrelationship is given in Table 14.

Table 14. KONPACT-2 Subroutine Summary

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship	
					Calls	Called by
MAIN	Sets up system dimensions and scratch array dimensions.		83	0.0	KORG2	
KORG2	Organizes input data and calls the primary overlays.	87	88	0.0	IDRO IDPR ERRM	MAIN
IDRO	Reorganizes the input data.			010		KORG2
IDPR	Prints input data.			0.0		KORG2
ERRM	Prints error message.			0.0		KORG2 DATAK
TDINVR	Inverts a nonsingular matrix or solves a set of linear equations.			0.0		DLAK CALI FFOC GCAL CAL
MP	Prints matrix data.		110	0.0		DLAK STRIC RESP FFOC
OUTP	Writes nonzero elements of a matrix on a data file.		111	0.0		DLAK FFOC
INPT	Reads nonzero elements of a matrix.			0.0		DLAK FFOC
ZERO	Initializes (or zeros) the elements of a matrix.			0.0		FFOC DDIAK DFFOC DLSA

Table 14. KONPACT-2 Subroutine Summary (Continued)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship	
					Calls	Called by
POLES	Computes the eignvalues of a matrix.		112	0.0	HESSEN QRCALL	DIAK FFOC
HESSEN	Reduces a matrix to upper Hessenberg form by Gaussian elimination.		113	0.0		POLES
QRCALL	Computes eigenvalues of an upper Hessenberg matrix.		114	0.0	QR	POLES
QR	Performs a double QR iteration on a real matrix.		115	0.0		QRCALL
DIAK	Computes optimal state feedback gains for a linear time-invariant system with a quadratic cost function.		84	1.0	INPT SHUFL MP STRIC TDINVR CALJ OUTP TIMER POLES	
TIMER	Computes time response.		89	1.0	SGUST	DIAK
SGUST	Computes step gust input.		90	1.0		TIMER
CALJ	Solves square Lyapunov equation.		91	1.0	TDINVR	COVAR COSTAT DIAK
STRIC	Computes stable set of starting gains for DIAK.		92	1.0	MP	DIAK
SHUFL	Reorders columns and rows of a matrix.		93	1.0		DIAK
GRAN	Generates random numbers.		118	1.0		TIMER

Table 14. KONPACT-2 Subroutine Summary (Continued)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship	
					Calls	Called by
FPOC	Computes simplified controller gains for a linear time-invariant system with a quadratic cost function.		85	2,0	POLES OUTP COVAR TRANS COSTAT RESP UNSCR MP ZERO INPT SHUF TDINVR	
SHUF	Reorders rows and columns of matrices.		94	2,0		FPOC
RESP	Computes covariances for disturbance inputs.		95	2,0		FPOC
COVAR	Computes covariance matrix.		96	2,0	CAL GCAL	RESP
COSTAT	Computes costate matrix.		97	2,0	CAL GCAL	FPOC
TRANS	Computes gradient transformation matrix.		98	2,0		FPOC
UNSCR	Transforms the gradient transformation matrix.		99	2,0		FPOC
GCAL	Solves rectangular Lyapunov equation.		100	2,0	TDINVR	COVAR COSTAT
CAL	Solves square Lyapunov equation		101	2,0	TDINVR	COVAR COSTAT
DATAK	Sets up array start indices and checks if scratch array size is sufficient.		86	3,0	DATAK DEFDC DESA FINK DERRM ERRM	

Table 14. KONPACT-2 Subroutine Summary (Concluded)

Subroutine	Description	Flow Chart Fig. #	Program Listing Fig. #	Overlay	Inter-relationship	
					Calls	Called by
DDIAK	Prepare data file for DLAK program.	102	103	3.0	ZERO FILE MPRS WTP	DATAK
DFFOC	Prepares data file for FFOC program.	104	105	3.0	ZERO FILE MPRS WTP	DATAK
DLSA	Prepares data for FINK program.	106	107	3.0	ZERO FILE MPRS INPTM	DATAK
FINK	Converts state space data into frequency domain data for LSA program.	108	109	3.0	MPRS	DATAK
MPRS	Prints matrix data.			3.0		DDIAK DFFOC DLSA FINK
FILE	Locates and inserts system labels on disc files and writes end of data mark.			3.0		DDIAK DFFOC DLSA
INPTM	Reads nonzero elements of a matrix.		116	3.0		DDIAK DFFOC DLSA
WTP	Writes nonzero elements of a matrix on a data file.		117	3.0		DDIAK DFFOC
DERRM	Prints an error message when the dimensions for scratch arrays are not sufficient.			3.0		DATAK

DESCRIPTION OF MAIN PROGRAMS

Program MAIN

This is the main program for overlay (0,0). This program assigns the various file numbers used in KONPACT-2. Maximum system dimensions and scratch array dimensions are set in this program. (Note that scratch arrays should be defined in DATAK program.) The program calls the organizing subroutine KOR2. The program listing is given in Figure 83.

Program DIAK

This is the main program for overlay (1,0). This program computes optimal state feedback gains for a linear time-invariant system with quadratic cost function. The program listing is given in Figure 84.

Program FFOC

This is the main program for overlay (2,0). This program computes simplified controller gains for a linear time-invariant system with a quadratic cost function. The program listing is given in Figure 85.

Program DATAK

This is the main program for overlay (3,0). The scratch arrays are defined here. The program computes the required scratch array dimensions and checks if the scratch array sizes are sufficient. The program calls the data preparation subroutines DDIAK, DFFOC, DLSA and FINK. The program listing is given in Figure 86. The dynamic storage map is given in Table 15.

Table 15. Dynamic Storage Map for Program DATAK

Calling Program Array	Array Start Index	Called Program Array	Maximum Dimension	Remarks
S1(M1)	M1 = 1	A(NXM, NXM)	NXM	Defined in MAIN
S1(M2)	M2 = M1 + NXM * NXM	B(NXM, NUM)	NUM	Defined in MAIN
S1(M3)	M3 = M2 + NXM * NUM	C(NRM, NXM)	NRM	Defined in MAIN
S1(M4)	M4 = M3 + NRM * NXM	D(NRM, NUM)		
S2(N1)	N1 = 1	B1(NXM, NUM)		
S2(N2)	N2 = N1 + NXM * NUM	B2(NXM, NUM)		
S2(N3)	N3 = N2 + NXM * NUM	C1(NRM, NXM)		
S2(N4)	N4 = N3 + NRM * NXM	C3(NRM, NXM)		
S2(N5)	N5 = N4 + NRM * NXM	D11(NRM, NUM)		
S2(N6)	N6 = N5 + NRM * NUM	BK(NUM, NRM)		
S2(N7)	N7 = N6 + NUM * NRM	BKC3(NUM, NXM)		
S2(K1)	K1 = 1	CC(NXRM, NXRUM)	NXRUM = NXM + NRM	Calculated in DATAK
S2(K2)	K2 = K1 + NXRM * NXRUM	NAME(NXRUM)	NXRUM = NXRM + NUM	Calculated in DATAK

DESCRIPTION OF BASIC SUBROUTINES

Subroutine KORG2

This subroutine organizes the execution of KONPACT-2 program. The input data cards for KONPACT-2 program are read and printed by the subroutine. The print specification cards are read in this subroutine, and the print control parameter IPRINT is set for the printed output options of the KONPACT-2 program. Under the control of the input data this subroutine calls the overlay loader to load the required primary overlay into central memory for execution. The flow chart is given in Figure 87 and the program listing is given in Figure 88.

Subroutine TIMER

This subroutine computes the time response for step command inputs and step gust inputs. The program listing is given in Figure 89.

Subroutine SGUST

This subroutine computes step gust input. The program listing is given in Figure 90.

Subroutine CAL1

This subroutine solves square Lyapunov equation. The program listing is given in Figure 91.

Subroutine STRIC

This subroutine computes a stable set of starting gains for DIAK. The program listing is given in Figure 92.

Subroutine SHUFL

This subroutine reorders the columns and rows of a matrix. The program listing is given in Figure 93.

Subroutine SHUF

This subroutine records rows and columns of matrices. The program listing is given in Figure 94.

Subroutine RESP

This subroutine computes covariances for disturbance inputs. The program listing is given in Figure 95.

Subroutine COVAR

This subroutine computes the covariance matrix. The program listing is given in Figure 96.

Subroutine COSTAT

This subroutine computes the costate matrix. The program listing is given in Figure 97.

Subroutine TRANS

This subroutine computes the gradient transformation matrix. The program listing is given in Figure 98.

Subroutine UNSCR

This subroutine transforms the gradient transformation matrix. The program listing is given in Figure 99.

Subroutine GCAL

This subroutine solves the rectangular Lyapunov equation. The program listing is given in Figure 100.

Subroutine CAL

This subroutine solves the square Lyapunov equation. The program listing is given in Figure 101.

Subroutine DDIAK

This subroutine reads data from cards or from file QDATA and prepares data file SCRTCH for DIAK subprogram. The flow chart is given in Figure 102 and the program listing is given in Figure 103.

Subroutine DFFOC

This subroutine reads data from cards or from file QDATA and prepares data file SCRTCH for FFOC subprogram. The flow chart is given in Figure 104 and the program listing is given in Figure 105.

Subroutine DLSA

This subroutine reads data from files QDATA, DDATA, and FDATA and prepares open loop or closed loop state space data. The flow chart is given in Figure 106, and the program listing is given in Figure 107.

Subroutine FINK

This subroutine uses the state space data obtained by the DLSA subroutine, computes the frequency domain data, and writes it on file SDSTP for the LSA program. The flow chart is given in Figure 108, and the program listing is given in Figure 109.

DESCRIPTION OF AUXILIARY SUBROUTINES

Subroutine MP

This subroutine prints matrix data. The program listing is given in Figure 110.

Subroutine OUTP

This subroutine writes the nonzero elements of a matrix on a data file. The program listing is given in Figure 111.

Subroutine POLES

This subroutine computes the eigenvalues of a matrix. The program listing is given in Figure 112.

Subroutine HESSEN

This subroutine computes the upper Hessenberg form of a matrix by Gaussian elimination. The program listing is given in Figure 113.

Subroutine QRCALL

This subroutine computes the eigenvalues of an upper Hessenberg form matrix. The program listing is given in Figure 114.

Subroutine QR

This subroutine performs a double QR iteration on a real matrix. The program listing is given in Figure 115.

Subroutine INPTM

This subroutine reads nonzero elements of a matrix. The program listing is given in Figure 116.

Subroutine WTP

This subroutine writes the nonzero elements of a matrix on a data file. The program listing is given in Figure 117.

Function GRAN

This function subroutine generates random numbers. The program listing is given in Figure 118.

For documentation of subroutines IDRO, IDPR, ERRM, TDINVR, INPT, ZERO, MPRS, FILE, and DERRM the reader is referred to Section III.

SECTION V

CONCLUSIONS AND RECOMMENDATIONS

The scope of this program was to develop programs to interface the level 2.01.00 FLEXSTAB with DIAK/FFOC optimal control programs. The theory and algorithms for the interface are presented in Volume I. Two demonstration examples are given in Volume III to show the data mechanics of the interface. A brief documentation of the interface program KONPACT is provided in this volume.

RECOMMENDATIONS FOR FUTURE SOFTWARE DEVELOPMENT WORK

- Full documentation of KONPACT should be made
- DIAK/FFOC programs should be modularized and variable dimensioned
- Faster algorithms should be used to reduce design time
- Reduced Controller Software (FFOC) should be augmented with the minimal order observer design capability

CONCLUSIONS

A large-scale software - KONPACT - for the design and analysis of active control systems is briefly documented in this volume. The work reported in Volumes I, II and III established the total dynamic system approach for the design and analysis.

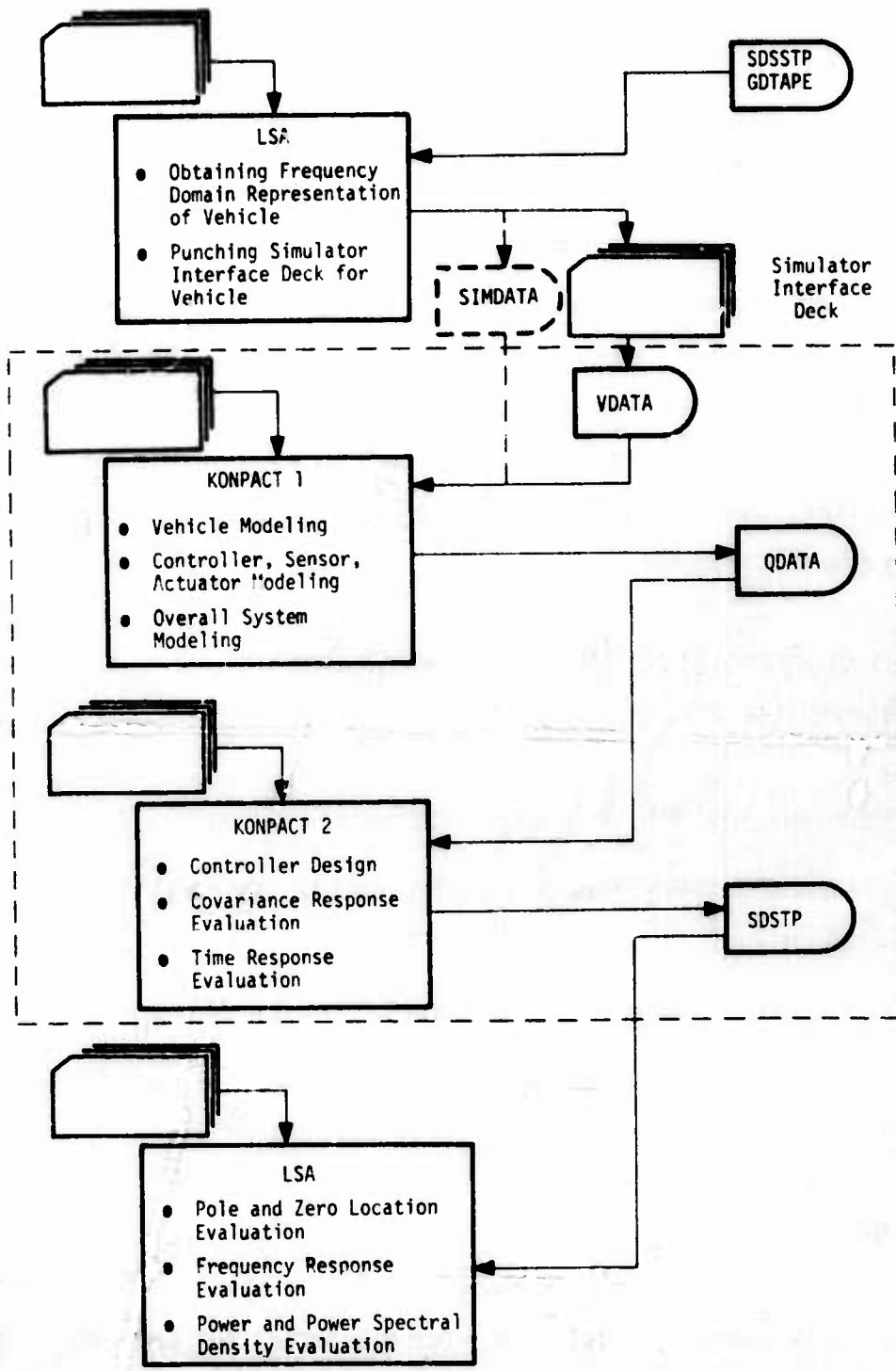


Figure 1. Interface Between LSA and KONPACT Programs

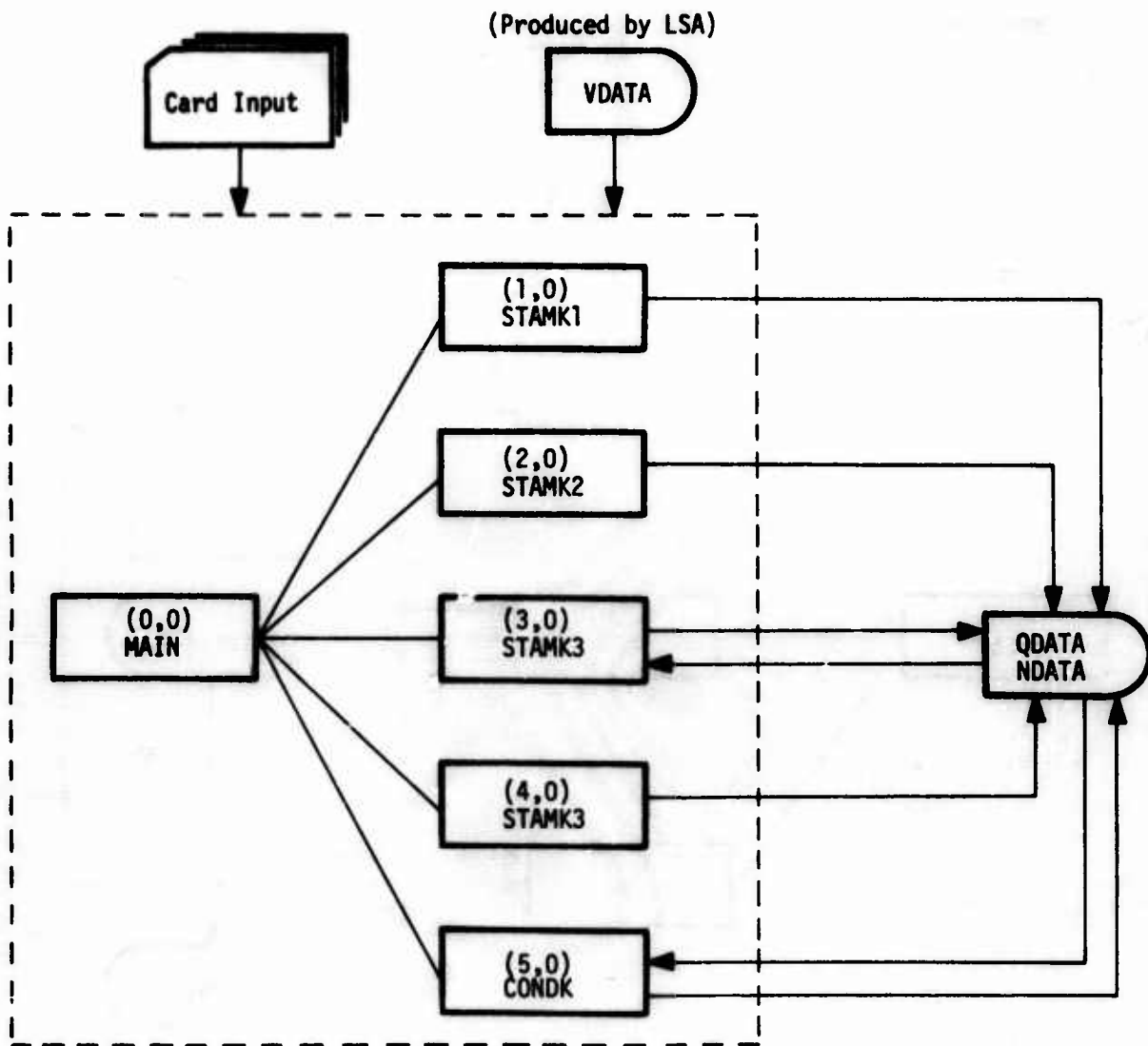


Figure 2. Overlay Structure of KONPACT-1

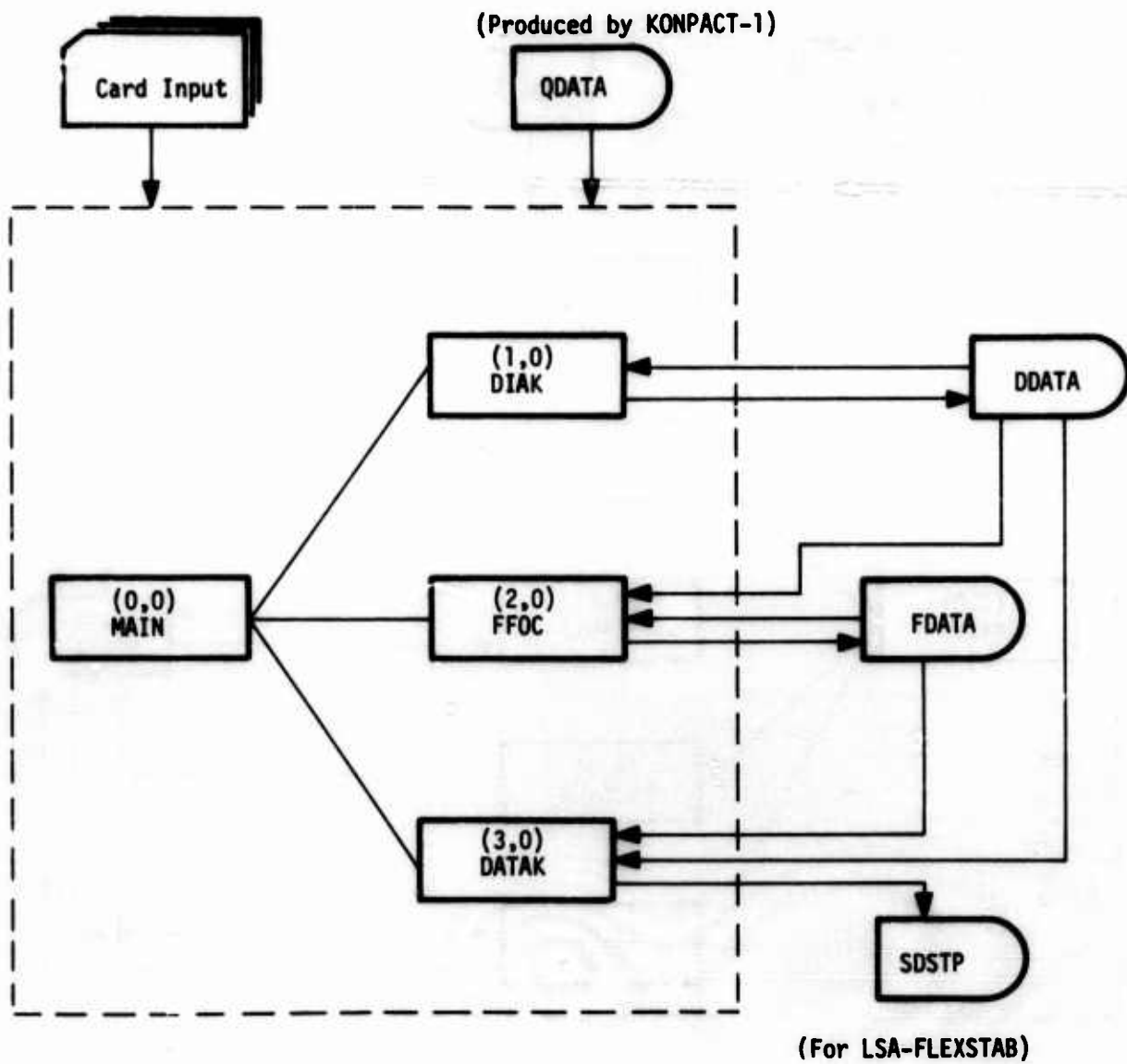


Figure 3. Overlay Structure of KONPACT-2

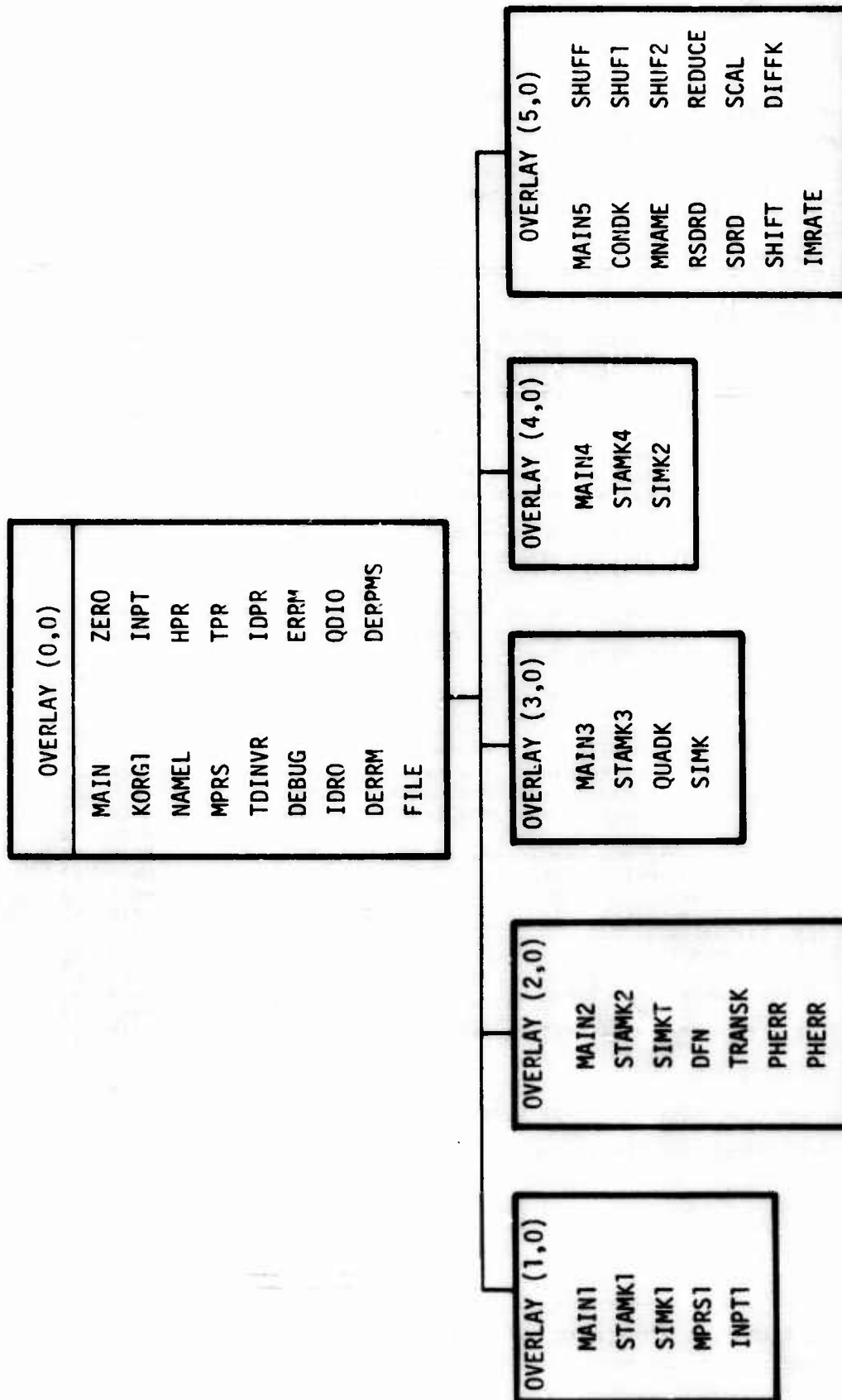


Figure 4. Overlay Structure and Subroutines in KONPACT-1

	OVERLAY(KONT,1)	MAIN1 2
	PROGRAM MAIN1	MAIN1 3
C		MAIN1 4
C	PURPOSE - TO SET UP DIMENSIONS AND CALL STAMK1	MAIN1 5
C	ANALYSIS - A F KODAR / J K HADASH - THE HONEYWELL INC	MAIN1 6
C	DATE WRITTEN - 1974	MAIN1 7
C		MAIN1 8
C	SUBPROGRAMS CALLED	MAIN1 9
C	DEBUG	MAIN1 10
C	STAMK1	MAIN1 11
C	DEJRM	MAIN1 12
C		MAIN1 13
	COMMON /DIM/ MS1,MS2,MS3,MS4,MAX4,MAX3,MAX2,MAX1,NRM,NUM,NYM	MAIN1 14
	MM,MR,AR,MR,MS,MM,MTF,MS1,MT	MAIN1 15
	COMMON /INOUT/ IP,IPR,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JO,J	MAIN1 16
	COMMON /SC1/ S1(1)	MAIN1 17
C	DIMENSION V(MAX4),F(MAX4,MAX4),U(NUM)	MAIN1 18
	COMMON /SC2/ S2(1)	MAIN1 19
C	DIMENSION A(NX4,NX2),B(NX4,NUM),C(NRM,NX4),D(NRM,NUM)	MAIN1 20
	COMMON /SC3/ S3(1)	MAIN1 21
C	DIMENSION NNS(NX4),VNS(NX4,2),DESS(NX4,10),UNITS(NX4,4)	MAIN1 22
C	DIMENSION NNO(IPM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)	MAIN1 23
C	DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)	MAIN1 24
	IF(IPRINT.EQ.6)CALL DEBUG(1.4,MAIN,4H1 01.0,1H)	MAIN1 25
		MAIN1 26
C	COMPUTE ARRAY START IDEAS	MAIN1 27
C		MAIN1 28
C	FOR V,W,F,U	MAIN1 29
C		MAIN1 30
	N1=1 N2=N1+MAX4 N3=N2+MAX4 N4=N3+MAX4*MAX4	MAIN1 31
	N5=N4+NUM	MAIN1 32
		MAIN1 33
C	FOR A,B,C,D	MAIN1 34
C		MAIN1 35
	M1=1 M2=M1+NX4*NX4 M3=M2+VX4*NUM M4=M3+VNM*NX4	MAIN1 36
	M5=M4+VNM*NUM	MAIN1 37
		MAIN1 38
C	FOR NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI	MAIN1 39
C		MAIN1 40
	L1=1 L2=L1+NX4 L3=L2+NX4*2 L4=L3+VX4*10 L5=L4+NX4*4	MAIN1 41
	L6=L5+NRM L7=L6+VNM*2 L8=L7+NRM*10 L9=L8+VNM*4	MAIN1 42
	L10=L9+NUM L11=L10+NUM*2 L12=L11+NUM*10 L13=L12+NUM*4	MAIN1 43
		MAIN1 44
C	CHECK IF SCRATCH ARRAY SIZES ARE SUFFICIENT	MAIN1 45
C		MAIN1 46
	IF((M5.GT.MS1).OR.(M5.GT.MS2).OR.(L13.GT.VS3))	MAIN1 47
	1CALL DEJRM(M5,M5,L13,MS4,MS1,MS2,MS3,MS4),C,4HMAIN,4H1 01W)	MAIN1 48
	IF(IPRINT.EQ.6)CALL DEBUG(2.4,MAIN,4H1 01.0,1H)	MAIN1 49
		MAIN1 50
C	CALL SUBROUTINE STAMK1	MAIN1 51
C		MAIN1 52
	CALL STAMK1(S1(N1),S1(N2),S1(N3),S1(N4),S2(M1),S2(M2),S2(M3),	MAIN1 53
	S2(M4),S3(L1),S3(L2),S3(L3),S3(L4),S3(L5),S3(L6),S3(L7),	MAIN1 54
	S3(L8),S3(L9),S3(L10),S3(L11),S3(L12),MAX4,MAX3,MAX2,MAX1,NRM,NUM,	MAIN1 55
	3NYM,MM,MS1,MS2,MS3,MS4,NH)	MAIN1 56
	IF(IPRINT.EQ.6)CALL DEBUG(3.4,MAIN,4H1 01.0,1H)	MAIN1 57
		MAIN1 58
C	RETURN TO MAIN OVERLAY	MAIN1 59
C		MAIN1 60
C	END	MAIN1 61

Figure 6. Program MAIN1 Program Listing

OVERLAY(KON1,2,1)	MAIN2 2
PROGRAM MAIN2	MAIN2 3
C	MAIN2 4
C	MAIN2 5
PURPOSE - TO SET UP DIMENSIONS AND CALL STAMP2	MAIN2 6
C	MAIN2 7
ANALYSTS - A F KONER / J K KARESH - THE MONEYBELL INC	MAIN2 8
C	MAIN2 9
DATE WRITTEN - 1975	MAIN2 10
C	MAIN2 11
SUBPROGRAMS CALLED	MAIN2 12
C	MAIN2 13
DEFUG	MAIN2 14
C	MAIN2 15
STAMP2	MAIN2 16
C	MAIN2 17
DEFUG	MAIN2 18
C	MAIN2 19
COMMON /INOUT/ IP,IM,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JO,JS	MAIN2 20
COMMON /DIM/ MS1,MS2,MS3,MS4,MAXM,MAXA,MAXV,NR4,NUM,NYM	MAIN2 21
1,MM,MD,MO,MP,MR,NR,MS,MN,MTEH,MST,MT	MAIN2 22
COMMON /SC1/ S1(1)	MAIN2 23
C	MAIN2 24
DIMENSION V(MAXM),W(MAXM),F(MAXM,MAXM)	MAIN2 25
C	MAIN2 26
DIMENSION XDOT(MST,MTEH),X(MST,MTEH),X1(1,MTEH),UI(1,MTEH)	MAIN2 27
C	MAIN2 28
DIMENSION U(NUM),NMX(MTEH),NNM(MTEH),VNI(MTEH)	MAIN2 29
COMMON /SC2/ S2(1)	MAIN2 30
C	MAIN2 31
DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NRM),D(NM,NUM)	MAIN2 32
C	MAIN2 33
DIMENSION AT(MST,MST,MTEH),AT(MST,1,MTEH)	MAIN2 34
C	MAIN2 35
DIMENSION CT(1,MST,MTEH),DT(1,1,MTEH)	MAIN2 36
C	MAIN2 37
DIMENSION P(MTEH,MTEH),D(MTEH,NUM),R(NRM,MTEH),S(NRM,NUM)	MAIN2 38
C	MAIN2 39
DIMENSION PRINT(2,MT),MS(2,MT,MTEH)	MAIN2 40
C	MAIN2 41
COMMON /SC3/ S3(1)	MAIN2 42
C	MAIN2 43
DIMENSION NNS(NXM),VNS(NXM,2),DESS(NX4,10),UNIT(NXM,4)	MAIN2 44
C	MAIN2 45
DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)	MAIN2 46
C	MAIN2 47
DIMENSION NNI(NUM),VNI(NUM,2),DESI(NM,10),UNITI(NUM,4)	MAIN2 48
C	MAIN2 49
IF(IPRINT.EQ.6)CALL DEFUG(1.4,MAIN,4M2 2.0,1M)	MAIN2 50
C	MAIN2 51
COMPUTE ARRAY START INDICES	MAIN2 52
C	MAIN2 53
C	MAIN2 54
FOR V,W,F,XDOT,X,R1,UI,U,NMX,NRM,NUM	MAIN2 55
C	MAIN2 56
C	MAIN2 57
N1=N1+MST*MTEH & N2=N2+MAXM & N3=N3+MAXM & N4=N3+MAXM*MAXM	MAIN2 58
C	MAIN2 59
N5=N4+MST*MTEH & N6=N5+MST*MTEH & N7=N6+MTEH & N8=N7+MTEH	MAIN2 60
C	MAIN2 61
N9=N8+NUM & N10=N9+MTEH & N11=N10+MTEH & N12=N11+MTEH	MAIN2 62
C	MAIN2 63
FOR A,P,C,D	MAIN2 64
C	MAIN2 65
C	MAIN2 66
N1=N1+NXM*NXM & N2=N2+NXM*NUM & N3=N2+NXM*NUM & N4=N3+NUM*NUM	MAIN2 67
C	MAIN2 68
N5=N4+NUM*NUM	MAIN2 69
C	MAIN2 70
FOR AT,DT,CT,DT,P,O,R,S,PRINT,MS	MAIN2 71
C	MAIN2 72
C	MAIN2 73
X1=1 & K2=K1+MST*MST*MTEH & K3=K2+MST*MTEH & K4=K3+MST*MTEH	MAIN2 74
C	MAIN2 75
K5=K4+MTEH & K6=K5+MTEH*MTEH & K7=K6+MTEH*NUM & K8=K7+NUM*MTEH	MAIN2 76
C	MAIN2 77
K9=K8+NUM*NUM & K10=K9+2*MT & K11=K10+2*MT*MTEH	MAIN2 78
C	MAIN2 79
FOR NNS,VNS,DESS,UNIT,S,NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI	MAIN2 80
C	MAIN2 81
C	MAIN2 82
L1=1 & L2=L1+NXM & L3=L2+NXM*2 & L4=L3+NXM*10 & L5=L4+NXM*6	MAIN2 83
C	MAIN2 84
L6=L5+NUM & L7=L6+NUM*2 & L8=L7+NUM*10 & L9=L8+NUM*6	MAIN2 85
C	MAIN2 86
L10=L9+NUM & L11=L10+NUM*2 & L12=L11+NUM*10 & L13=L12+NUM*6	MAIN2 87
C	MAIN2 88
C	MAIN2 89
CHECK IF SCRATCH ARRAY SIZES ARE SUFFICIENT	MAIN2 90
C	MAIN2 91
C	MAIN2 92
MKS11=MS	MAIN2 93
IF(K11.GT.MS)MKS11=M11	MAIN2 94
IF((N12.GT.MS11).OR.(MS.GT.MS21).OR.(K11.GT.MS21).OR.(L13.GT.MS31))	MAIN2 95
CALL DEFUG(N12,MKS11,L13,MS4,MS1,MS2,MS3,MS4,2.0,4,MAIN,4M2 1M)	MAIN2 96
IF(IPRINT.EQ.6)CALL DEFUG(1.4,MAIN,4M2 2.0,1M)	MAIN2 97

Figure 7. Program MAIN2 Program Listing

C		MAIN2 65
C	CALL SUBROUTINE STAMP2	MAIN2 66
C		MAIN2 67
	CALL STAMP2(S1(N1),S1(N2),S1(N3),S1(N4),S1(N5),S1(N6),S1(N7),	MAIN2 68
	1S1(N8),S1(N9),S1(N10),S1(N11),S2(M1),S2(M2),S2(M3),S2(M4),S2(K1),	MAIN2 69
	2S2(K2),S2(K3),S2(K4),S2(K5),S2(K6),S2(K7),S2(K8),S2(K9),S2(K10),	MAIN2 70
	3S3(L1),S3(L2),S3(L3),S3(L4),S3(L5),S3(L6),S3(L7),S3(L8),	MAIN2 71
	4S3(L9),S3(L10),S3(L11),S3(L12),MAXN,MAXM,NXM,NRM,SUM,VYM,MH,MTFB,	MAIN2 72
	5MST,MT,MS1,MS2,MS3,MS4,MR)	MAIN2 73
	IF(IIP)INT.F0.6)CALL DEMIG(3.64*MAIN.6M2 .2.0.10)	MAIN2 74
C		MAIN2 75
C	RETURN TO MAIN OVERLAY	MAIN2 76
C		MAIN2 77
	END	MAIN2 78

Figure 7. Program MAIN2 Program Listing (Concluded)

	OVERLAY(KON) 3.)	MAIN3 2
	PROGRAM MAIN3	MAIN3 3
C		MAIN3 4
C	PURPOSE - TO SET UP DIMENSIONS AND CALL STACKS	MAIN3 5
C	ANALYSIS - A F KONGR / J K MARSH - THE HONEYWELL INC	MAIN3 6
C	DATE WRITTEN - 1975	MAIN3 7
C		MAIN3 8
C	SUBPROGRAMS CALLED	MAIN3 9
C	DEFIN	MAIN3 10
C	STACKS	MAIN3 11
C		MAIN3 12
C	COMMON / I/O UT/ IP, IP, PRINT, INSERT, LOCATE, NULL, MARK(20), JN, JU, JS	MAIN3 13
C	COMMON / DIM/ MS1, MS2, MS3, MS4, MAX, MAX4, XAM, NRM, NUM, NYM	MAIN3 14
C	1, MM, MO, MO, MR, MR, MR, MS, MN, MTR, MST, MT	MAIN3 15
C	COMMON / SC1/ S1(1)	MAIN3 16
C	DIMENSION V(MAX), F(MAN, YAA4)	MAIN3 17
C	DIMENSION XDOT(NM, MR), X(NM, MR), ZI(NRM, MR), UI(NUM, MB)	MAIN3 18
C	DIMENSION RIN(MMR), U(NM), NIX(MR), VNP(MR), NNI(MR)	MAIN3 19
C	COMMON / SC2/ S2(1)	MAIN3 20
C	DIMENSION A(NM, NM), C(NM, NM), D(NM, NM)	MAIN3 21
C	COMMON / SC3/ S3(1)	MAIN3 22
C	DIMENSION VNS(NM, 2), VNS(NM, 2), DESS(NM, 10), UNITS(NM, 4)	MAIN3 23
C	DIMENSION VNO(NRM, 2), VNO(NRM, 2), DESO(NR, 10), UNITS(NRM, 4)	MAIN3 24
C	DIMENSION VNI(NUM, 2), VNI(NUM, 2), DESI(NUM, 10), UNITS(NUM, 4)	MAIN3 25
C		MAIN3 26
C	PRINT SYSTEM DIMENSIONS IF NEEDED	MAIN3 27
C		MAIN3 28
C	IF (IP, INT, EQ, 6) WRITE (IP, 145) MS1, MS2, MS3, MS4, MAX, MAX4	MAIN3 29
C	1, XAM, RM, NUM, NYM, MM, MO, MR, MR, MR, MS, MN, MTR, MST, MT	MAIN3 30
C	155 FORMAT (1X, 15 (TS, 1X))	MAIN3 31
C		MAIN3 32
C	COMPUTE MAXIMUM SIZE FOR RI:	MAIN3 33
C		MAIN3 34
C	NPM=NR*MR	MAIN3 35
C		MAIN3 36
C	COMPUTE ARRAY START INDICES	MAIN3 37
C		MAIN3 38
C	FOR V=M, F=XDOT, X, RT, U, RIN, I, NM, NRM, NYM	MAIN3 39
C		MAIN3 40
C	N1=1 N2=N1+MAX N3=N2+MAX N4=N3+MAX*MAX	MAIN3 41
C	N5=N4+XAM*MR N6=N5+XAM*MR N7=N6+NR*MR N8=N7+NUM*MR	MAIN3 42
C	N9=N8+NUM N10=N9+NR*MR N11=N10+MR N12=N11+MR N13=N12+MB	MAIN3 43
C		MAIN3 44
C	FOR A=P, C, D	MAIN3 45
C		MAIN3 46
C	M1=1 M2=M1+XAM*NM M3=M2+XAM*NUM M4=M3+NR*XAM	MAIN3 47
C	M5=M4+NR*NUM	MAIN3 48
C		MAIN3 49
C	FOR N=S, VNS, DESS, UNITS, NNO, VNO, DESO, UNITS, NNI, VNI, DESI, UNITS	MAIN3 50
C		MAIN3 51
C	L1=1 L2=L1+NM L3=L2+XV*MR L4=L3+XAM*10 L5=L4+XAM*4	MAIN3 52
C	L6=L5+NR*MR L7=L6+NR*MR L8=L7+NR*MR L9=L8+NR*MR	MAIN3 53
C	L10=L9+NUM L11=L10+NUM*2 L12=L11+NUM*10 L13=L12+NUM*4	MAIN3 54
C		MAIN3 55
C	PRINT ARRAY OVERLAPPING NUMBERS IF NEEDED	MAIN3 56
C		MAIN3 57
C	IF (IP, INT, EQ, 6) WRITE (IP, 145) M1, M2, N3, V4, N5, V6	MAIN3 58
C	1, N7, N8, N9, N10, N11, N12, N13	MAIN3 59
C	IF (IP, INT, EQ, 6) WRITE (IP, 145) M1, M2, M3, M4, M5	MAIN3 60
C	IF (IP, INT, EQ, 6) WRITE (IP, 145) L1, L2, L3, L4, L5, L6	MAIN3 61
C	1, L7, L8, L9, L10, L11, L12, L13	MAIN3 62
C		MAIN3 63
C	CHECK IF SCRATCH ARRAY SIZES ARE SUFFICIENT	MAIN3 64

Figure 8. Program MAIN3 Program Listing

```

C                                     MAIN3 65
C   IF (IN1.GT.MS1).OR.(M2.GT.MS2).OR.(L13.GT.MS3)   MAIN3 66
C   1CALL MCHRM(113.MK50.L13.MS4.M1.MS2.MS3.MS4.3.7.4HMAIN.4H3 .I) MAIN3 67
C                                     MAIN3 68
C   CALL SUBROUTINE STANK3                               MAIN3 69
C                                     MAIN3 70
C   CALL STANK3(S1(N1).S1(N2).S1(N3).S1(N4).S1(N5).S1(N6).S1(N7). MAIN3 71
C   1S1(N8).S1(N9).S1(N10).S1(N11).S1(N12).S2(M1).S2(M2).S2(M3).S2(M4). MAIN3 72
C   2S3(L1).S3(L2).S3(L3).S3(L4).S3(L5).S3(L6).S3(L7).S3(L8). MAIN3 73
C   3S3(L9).S3(L10).S3(L11).S3(L12).MAIN.MAM.NAM.NEM.NUM.NYM.MH. MAIN3 74
C   4MN.MM.MP.MQ.MR.MS1.MS2.MS3.45.NH.NV.MH) MAIN3 75
C                                     MAIN3 76
C   RETURN TO MAIN OVERLAY                             MAIN3 77
C                                     MAIN3 78
C   END                                                 MAIN3 79

```

Figure 8. Program MAIN3 Program Listing (Concluded)

	OVERLAY(KOMI,4,0)	MAIN4 2
	PROGRAM MAIN4	MAIN4 3
C		MAIN4 4
C	PURPOSE - TO SET UP DIMENSIONS AND CALL STAMK4	MAIN4 5
C	ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC	MAIN4 6
C	DATE WRITTEN - 1975	MAIN4 7
C		MAIN4 8
C	SUBPROGRAMS CALLED	MAIN4 9
C	DEPRM	MAIN4 10
C	STAMK4	MAIN4 11
C		MAIN4 12
	COMMON /DIM/ MS1,MS2,MS3,MS4,MAXN,MAXY,MAXZ,NRM,NUM,NYM	MAIN4 13
	,MM,MP,MQ,MR,MS,MN,MTFH,MST,MT	MAIN4 14
	COMMON /INOUT/ IR,IR,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JO,JS	MAIN4 15
	COMMON /SC1/ S1(1)	MAIN4 16
C	DIMENSION V(MAXN),W(MAXN),F(MAXN,MAXM),U(NUM)	MAIN4 17
	COMMON /SC2/ S2(1)	MAIN4 18
C	DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM)	MAIN4 19
	COMMON /SC3/ S3(1)	MAIN4 20
C	DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)	MAIN4 21
C	DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)	MAIN4 22
C	DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)	MAIN4 23
C		MAIN4 24
C	COMPUTE ARRAY START INDEXES	MAIN4 25
C		MAIN4 26
C	FOR V,W,F,U	MAIN4 27
C		MAIN4 28
	N1=1 & N2=N1*MAXN & N3=N2*MAXM & N4=N3*MAXN*MAXM	MAIN4 29
	N5=N4*NUM	MAIN4 30
C		MAIN4 31
C	FOR A,R,C,D	MAIN4 32
C		MAIN4 33
	M1=1 & M2=M1*NXM*NXM & M3=M2*NUM & M4=M3*NRM*NXM	MAIN4 34
	M5=M4*NRM*NUM	MAIN4 35
C		MAIN4 36
C	FOR NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI	MAIN4 37
C		MAIN4 38
	L1=1 & L2=L1*NXM & L3=L2*NXM*2 & L4=L3*NUM*10 & L5=L4*NXM*4	MAIN4 39
	L6=L5*NRM & L7=L6*NRM*2 & L8=L7*NRM*10 & L9=L8*NRM*4	MAIN4 40
	L10=L9*NUM & L11=L10*NUM*2 & L12=L11*NUM*10 & L13=L12*NUM*4	MAIN4 41
C		MAIN4 42
C	CHECK IF SCRATCH APRAY SIZES ARE SUFFICIENT	MAIN4 43
C		MAIN4 44
	IF((N4.GT.MS1).OR.(M5.GT.MS2).OR.(L13.GT.MS3))	MAIN4 45
	ICALL DERRM(N5,M5,L13,MS4,MS1,MS2,MS3,MS4,4,0,4,MAIN,4,44 .,I)	MAIN4 46
C		MAIN4 47
C	CALL SUBROUTINE STAMK4	MAIN4 48
C		MAIN4 49
	CALL STAMK4(S1(N1),S1(N2),S1(N3),S1(N4),S2(M1),S2(M2),S2(M3),	MAIN4 50
	S2(M4),S3(L1),S3(L2),S3(L3),S3(L4),S3(L5),S3(L6),S3(L7),	MAIN4 51
	S3(L8),S3(L9),S3(L10),S3(L11),S3(L12),MAXN,MAXY,NXM,NRM,NUM,	MAIN4 52
	JNYM,M,MS1,MS2,MS3,MS4,NH)	MAIN4 53
C		MAIN4 54
C	RETURN TO MAIN OVERLAY	MAIN4 55
C		MAIN4 56
	END	MAIN4 57

Figure 9. Program MAIN4 Program Listing

```

OVERLAY(KONT=5,0)
PROGRAM MAINS
MAINS 2

PURPOSE - TO SET UP DIMENSIONS AND CALL CONDK
MAINS 3
ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC
MAINS 4
DATE WRITTEN - 1975
MAINS 5

SUBPROGRAMS CALLED
MAINS 6
  DERRM
MAINS 7
  RESPK
MAINS 8
  DEBUG
MAINS 9

COMMON /INGO1/ IR,IRW,IPRINT,INSERT,LOCATE, NULL, MARK(20),JN,JO,JS
MAINS 10
COMMON /DIMP/ MS1,MS2,MS3,MS4,MAXN,MAXM,NAM,NUM,NYM
MAINS 11
COMMON /MOP/ MP,MR,MR,MR,MS,MM,MTFB,MSI,MT
MAINS 12
COMMON /SYS/ SCDF,SDS(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)
MAINS 13
COMMON /PHEAD(20)
MAINS 14
COMMON /SC2/ S2(1)
MAINS 15
DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
MAINS 16
DIMENSION CM(NRM,NYM),DM(NRM,NUM)
MAINS 17
COMMON /SC3/ S3(1)
MAINS 18
DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
MAINS 19
DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
MAINS 20
DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
MAINS 21
DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
MAINS 22
DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
MAINS 23
DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
MAINS 24
COMMON /SC1/ S1(1)
MAINS 25
DIMENSION DUMMY1(NDM1,NDM2),DUMMY2(VDM21,VDM22),DUMMY3(NUM)
MAINS 26
DIMENSION ES(NXM,NUM),ER(NRM,NUM)
MAINS 27
DIMENSION NSHJFS(NXM),NSHJFO(NRM),NSHJFI(NUM)
MAINS 28
DIMENSION CS(NRM,NXM),DS(NRM,NUM),CW(NRM,NXM),DW(NRM,NUM)
MAINS 29
DIMENSION IRS(NRM),O(NRM,NRM)
MAINS 30
IF(IPRINT.EQ.6)CALL DEBUG(1.4HMAIN.4H5 .5.0.1W)
MAINS 31
NXUM=IXM+NUM $ MWORD=17 $ NRSM=1
MAINS 32
NDM1=MAX0(MWORD,NXM,NRM,NRSM)
MAINS 33
NDM2=MAX0(NXUM,NRM)
MAINS 34
NDM21=MAX0(NRM,NXM,NRSM)
MAINS 35
NDM22=MAX0(NXM,NUM,NRM)
MAINS 36
MAINS 37
MAINS 38
MAINS 39
MAINS 40
MAINS 41
PRINT ERROR MESSAGE IF DIMENSION OF SCRATCH ARRAYS ARE INSUFFICIENT
MAINS 42
MAINS 43
M1=1 $ M2=M1+NXM+NXM $ M3=M2+NXM+NUM $ M4=M3+NRM+VXM
MAINS 44
M5=M4+NRM+NUM $ M6=M5+NRM+NXM $ M7=M6+NRM+NUM
MAINS 45
N1=1 $ N2=N1+NXM $ N3=N2+NXM*2 $ N4=N3+NXM*10 $ N5=N4+NXM*6
MAINS 46
N6=N5+NRM $ N7=N6+NRM*2 $ N8=N7+NRM*10 $ N9=N8+NRM*4
MAINS 47
N10=N9+NUM $ N11=N10+NUM*2 $ N12=N11+VDM*10 $ N13=N12+NUM*4
MAINS 48
N14=N13+NXM $ N15=N14+NXM*2 $ N16=N15+NXM*10 $ N17=N16+NXM*4
MAINS 49
N18=N17+NRM $ N19=N18+NRM*2 $ N20=N19+NRM*10 $ N21=N20+NRM*4
MAINS 50
N22=N21+NUM $ N23=N22+NUM*2 $ N24=N23+NUM*10 $ N25=N24+NUM*4
MAINS 51
L1=1 $ L2=L1+NDM11+NDM12 $ L3=L2+NDM21+NDM22 $ L4=L3+NUM
MAINS 52
L5=L4+NXM+NUM $ L6=L5+NRM+NUM $ L7=L6+NXM $ L8=L7+NRM
MAINS 53
L9=L8+NUM $ L10=L9+NRM+NXM $ L11=L10+VDM+NUM
MAINS 54
L12=L11+NRM+NXM $ L13=L12+NRM+NUM $ L14=L13+NRM
MAINS 55
L15=L14+NRM+NRM
MAINS 56
IF((L15.GT.MS1).OR.(M7.GT.MS2).OR.(N25.GT.MS3))
MAINS 57
CALL DERRM(L15,M7,N25,MS4,MS1,MS2,MS3,MS4,5.0,4HMAIN.4H5 .1W)
MAINS 58
IF(IPRINT.EQ.6)CALL DEBUG(2.4HMAIN.4H5 .5.0.1W)
MAINS 59
MAINS 60
CALL SUBROUTINE CONDK
MAINS 61
MAINS 62
CALL CONDK(S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),
MAINS 63
S3(N1),S3(N2),S3(N3),S3(N4),S3(N5),S3(N6))
MAINS 64

```

Figure 10. Program MAIN5 Program Listing

	2S3(N7),S3(N8),S3(N9),S3(N10),S3(N11),S3(N12),	MAINS 65
	3S3(N13),S3(N14),S3(N15),S3(N16),S3(N17),S3(N18),	MAINS 66
	4S3(N19),S3(N20),S3(N21),S3(N22),S3(N23),S3(N24),	MAINS 67
	5S1(L1),S1(L2),S1(L3),S1(L4),S1(L5),S1(L6),	MAINS 68
	6S1(L7),S1(L8),S1(L9),S1(L10),S1(L11),S1(L12),	MAINS 69
	7S1(L13),S1(L14),NXM,NRM,NUM,NDM11,NDM12,NDM21,NDM22)	MAINS 70
	IF(IPRINT.EQ.6)CALL DFRUG(3,4,MAIN,4M5 .S,0,FW)	MAINS 71
C		MAINS 72
C	RETURN TO MAIN OVEPLAY	MAINS 73
C		MAINS 74
	END	MAINS 75

Figure 10. Program MAIN5 Program Listing (Concluded)

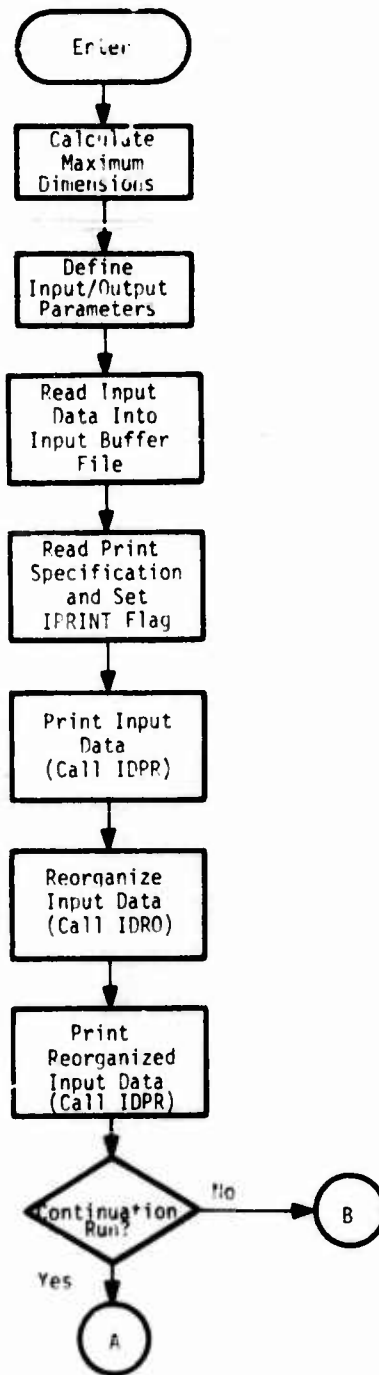


Figure 11. Subroutine KORGI Flow Chart

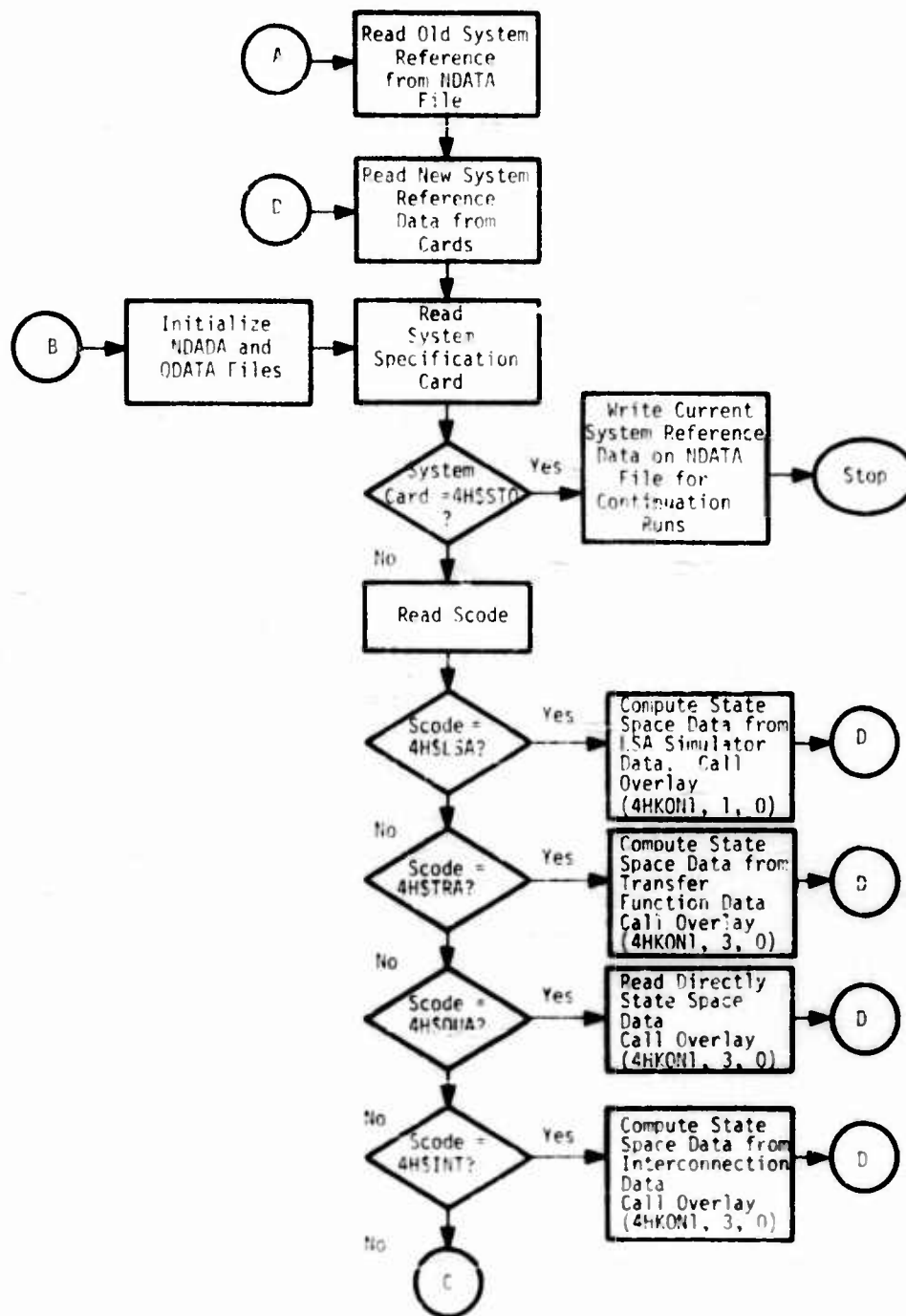


Figure 11. Subroutine KORGI Flow Chart (Continued)

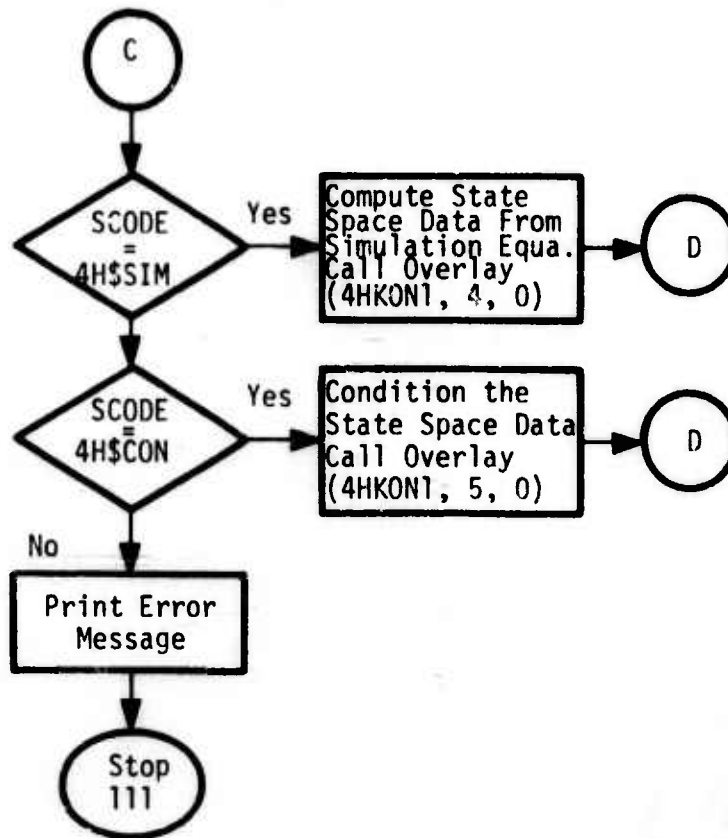


Figure 11. Subroutine KORGI Flow Chart (Concluded)

C	SUBROUTINE KORGI	KORGI 2	
C		KORGI 3	
C	ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC	KORGI 4	
C	PURPOSE - TO ORGANIZE EXECUTION OF KONPACT-1 PROGRAMS	KORGI 5	
C	DATE WRITTEN - JULY 1975	KORGI 6	
C		KORGI 7	
C	SUBPROGRAMS CALLED	KORGI 8	
C	MPW	KORGI 9	
C	IDPO	KORGI 10	
C	IDPP	KORGI 11	
C	FILE	KORGI 12	
C		KORGI 13	
C	LABELLED COMMON LIST	KORGI 14	
C	IR	FILE NUMBER FOR CARD READER	KORGI 15
C	IW	FILE NUMBER FOR LINE PRINTER	KORGI 16
C	IPRINT	PRINT CONTROL FLAG	KORGI 17
C	INSERT	HOLLFRITH INSE	KORGI 18
C	LOCATE	HOLLFRITH LOCA	KORGI 19
C	NULL	HOLLFRITH NULL	KORGI 20
C	MARK	HOLLFRITH SS..S	KORGI 21
C	JN	FILE NO FOR NAME LIST DATA FILE	KORGI 22
C	JQ	FILE NO FOR QUADRUPLE DATA FILE	KORGI 23
C	JS	FILE NO FOR SCRATCH FILE	KORGI 24
C	IHEAD	LABEL NAME	KORGI 25
C	SCODE	SYSTEM CODE WORD	KORGI 26
C	SDES	SYSTEM DESCRIPTION	KORGI 27
C	MSYS	PRESENT SYSTEM NUMBER	KORGI 28
C	HEAD	PRESENT SYSTEM HEADING	KORGI 29
C	NSYS	SYSTEM NUMBERS	KORGI 30
C	SHEAD	SYSTEM HEADINGS	KORGI 31
C	PHEAD	PREVIOUS SYSTEM HEADING	KORGI 32
C		KORGI 33	
C	COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)	KORGI 34	
C	1,PHEAD(20)	KORGI 35	
C	COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS	KORGI 36	
C	COMMON /DIM/ MS1,MS2,MS3,MS4,MAXN,MAXM,NXM,NRM,NUM,NYM	KORGI 37	
C	1,MM,MP,MQ,MR,MR,NB,MS,MN,MTFB,MST,MT	KORGI 38	
C	DIMENSION CARD(70),LAREL(70),AHEAD(20,20)	KORGI 39	
C	INTEGR HINSE,HLOCA,HNULL,HNDLR	KORGI 40	
C	DATA HINSE,HLOCA,HNULL,HNDLR/4HINSE,4HLOCA,4HNULL,4HSSSS/	KORGI 41	
C	DATA HSTOP,MSYST,HFM,HC,HCONT/4HSTOP,4MSYST,4HFM .2HC .4HCONT/	KORGI 42	
C	DATA HPRIN,HTHIN,HFRYT/4HPRIN,4HTHIN,4HFRYT/	KORGI 43	
C	DATA HTPUT,HNAL,HPIIT/4HTPUT,4HNAL .4HPUT /	KORGI 44	
C	DATA HDLSA,HDCON,HNOUA/4HLSA,4HSCON,4HSOUA/	KORGI 45	
C	DATA HTRA,HDIRT,HDSIM/4HSTRA,4HDIRT,4HDSIM/	KORGI 46	
C	DATA HREFE,HRENC,HFBRR,HRRBR/4HREFE,4HPENC,4HE .4H /	KORGI 47	
C	DATA HEND/4HEND /	KORGI 48	
C		KORGI 49	
C	MAXIMUM DIMENSIONS FOR V,W AND F	KORGI 50	
C		KORGI 51	
C	MAXN=JN+NYM+NRM & MAXM=NXM+NYM+NXM+NYM	KORGI 52	
C		KORGI 53	
C	MAXIMUM DIMENSIONS FOR COMBINING TRANSFER FUNCTION BLOCKS	KORGI 54	
C		KORGI 55	
C	MST=5 & MT=6	KORGI 56	
C		KORGI 57	
C	MAXIMUM DIMENSIONS FOR COMBINING SUBSYSTEMS	KORGI 58	
C		KORGI 59	
C	MM=NRM	KORGI 60	
C	IF(MM,LT,NUM)MM=NUM	KORGI 61	
C	MS=NXM & MQ=MR & MR=MR & MP=MM+? & NN=MM+MM	KORGI 62	
C		KORGI 63	
C	MAXIMUM SYSTEM NUMBER	KORGI 64	

Figure 12. Subroutine KORGI Program Listing

C	NR=9	KORG1 65
C		KORG1 66
C	DEFINE INPUT/OUTPUT PARAMETERS	KORG1 67
C		KORG1 68
	IR=5 & I=9 & IPRINT=4 & JN=7 & JQ=8 & JS=3	KORG1 69
	INSERT=MINSE & LOCATE=MLOCA & NULL=MNULL	KORG1 70
	DO 10 I=1,20	KORG1 71
100	MARK(I)=MDOLR	KORG1 72
	LAHEL(1)=HREFE	KORG1 73
	LAHEL(2)=HRENC	KORG1 74
	LAHEL(3)=HEERRR	KORG1 75
	DO 105 I=4,20	KORG1 76
105	LAHEL(I)=MHRRR	KORG1 77
C		KORG1 78
C	READ INPUT DATA INTO INPUT DATA BUFFER FILE	KORG1 79
C		KORG1 80
	REWIND IR	KORG1 81
110	CONTINUE	KORG1 82
	READ(A,120)CARD	KORG1 83
	IF(EOF(6))140,115	KORG1 84
115	CONTINUE	KORG1 85
	WRITE(IR,120)CARD	KORG1 86
120	FORMAT(20A4)	KORG1 87
	GO TO 110	KORG1 88
140	CONTINUE	KORG1 89
	ENDFILE IR	KORG1 90
	REWIND IR	KORG1 91
C		KORG1 92
C	READ PRINT SPECIFICATION AND SET IPRINT	KORG1 93
C		KORG1 94
C		KORG1 95
142	CONTINUE	KORG1 96
	READ(IR,170)CARD	KORG1 97
	DECODE(4,143,CARD(1))CC,DUMMY	KORG1 98
143	FORMAT(A2,A2)	KORG1 99
	IF(CC,EO,MC)GO TO 142	KORG1100
	IF(CARD(1).NE.MPRIN)GO TO 152	KORG1101
	IF(CARD(3).EQ.MTHIN)IPRINT=3	KORG1102
	IF(CARD(3).EQ.MTHIN)GO TO 142	KORG1103
	IF(CARD(3).EQ.HERYT)IPRINT=4	KORG1104
	IF(CARD(3).EQ.HERYT)GO TO 142	KORG1105
	IF(CARD(3).NE.MPUT)GO TO 144	KORG1106
	IF(IPRINT.EQ.1)IPRINT=5	KORG1107
	IF(IPRINT.EQ.5)GO TO 142	KORG1108
	IPRINT=3	KORG1109
	GO TO 142	KORG1110
144	CONTINUE	KORG1111
	IF(CARD(3).NE.MNAL)GO TO 144	KORG1112
	IF(IPRINT.EQ.1)IPRINT=4	KORG1113
	IF(IPRINT.EQ.4)GO TO 142	KORG1114
	IPRINT=2	KORG1115
	GO TO 142	KORG1116
145	CONTINUE	KORG1117
	IF(CARD(3).NE.MPUT)GO TO 144	KORG1118
	IF(IPRINT.EQ.4)IPRINT=1	KORG1119
	IF(IPRINT.EQ.1)GO TO 142	KORG1120
	IF(IPRINT.EQ.2)IPRINT=4	KORG1121
	IF(IPRINT.EQ.3)IPRINT=5	KORG1122
	IF(IPRINT.EQ.4)GO TO 142	KORG1123
	IF(IPRINT.EQ.5)GO TO 142	KORG1124
	IPRINT=1	KORG1125
	GO TO 142	KORG1126
C		KORG1127
C	PRINT ERROR MESSAGE	KORG1128
C		KORG1129
146	CONTINUE	KORG1130

Figure 12. Subroutine KORGI Program Listing (Continued)

	WRITE(IW,150)	KORG1131
	150 FORMAT(IH1,/// <i>1X,30H</i> PRINT CARD SPECIFICATION ERROR,/// <i>1X,143H</i> INPUT AND FINAL OUTPUT DATA WILL BE PRINTED)	KORG1132
	IPRINT=4	KORG1133
C		KORG1134
C	PRINT INPUT DATA	KORG1135
C		KORG1136
	152 CONTINUE	KORG1137
	REWIND IR	KORG1138
	IF((IPRINT.NE.1).AND.(IPRINT.LT.4))GO TO 15A	KORG1139
	WRITE(IW,154)	KORG1140
	154 FORMAT(IH1,/// <i>1X,24H</i> *** INPUT DATA CARDS ***,/// <i>1</i>)	KORG1141
	CALL IDPR(IR,IW)	KORG1142
	REWIND IR	KORG1143
	15A CONTINUE	KORG1144
C		KORG1145
C	REORGANIZE INPUT DATA	KORG1146
C		KORG1147
C	CALL IDRO(IR,IW,JS)	KORG1148
C		KORG1149
C	PRINT REORGANIZED INPUT DATA	KORG1150
C		KORG1151
	IF(IPRINT.LT.6)GO TO 164	KORG1152
	WRITE(IW,160)	KORG1153
	160 FORMAT(IH1,/// <i>1X,30H</i> *** REORGANIZED INPUT DATA ***,/// <i>1</i>)	KORG1154
	CALL IDPR(IR,IW)	KORG1155
C		KORG1156
C	READ INITIALIZING INSTRUCTIONS	KORG1157
C		KORG1158
	164 CONTINUE	KORG1159
	ISYS=	KORG1160
	DO 164 I=1,9	KORG1161
	DO 164 J=1,20	KORG1162
	166 SHEAD(I,J)=HRRRR	KORG1163
	168 CONTINUE	KORG1164
	READ(IR,170)CARD	KORG1165
	170 FORMAT(20A4)	KORG1166
	IF(CARD(1).EQ.HPRIM)GO TO 16B	KORG1167
	IF(CARD(1).NE.HCONT)GO TO 175	KORG1168
	CALL FILE(JN,LOCATE,LABEL)	KORG1169
	READ(N)((SHEAD(I,J),J=1,20),I=1,9)	KORG1170
	CALL FILE(JN,NULL,LABEL)	KORG1171
	WRITE(IW,430)	KORG1172
	WRITE(IW,440)((SHEAD(I,J),J=1,20),I=1,9)	KORG1173
	GO TO 180	KORG1174
	175 CONTINUE	KORG1175
	CALL FILE(JN,INSERT,MARK)	KORG1176
	CALL FILE(JO,INSERT,MARK)	KORG1177
	GO TO 190	KORG1178
C		KORG1179
C	READ SYSTEM REFERENCE DATA	KORG1180
C		KORG1181
	180 CONTINUE	KORG1182
	READ(IR,170)CARD	KORG1183
	IF(CARD(1).NE.HREFE)GO TO 190	KORG1184
	183 CONTINUE	KORG1185
	READ(IR,170)CARD	KORG1186
	IF(CARD(1).EQ.HEND)GO TO 185	KORG1187
	DECODE(4,220,CARD(1))DI,NSYSNO,DZ	KORG1188
	DO 185 I=1,20	KORG1189
	185 SHEAD(NSYSNO,I)=CARD(I)	KORG1190
	GO TO 183	KORG1191
	188 CONTINUE	KORG1192
	WRITE(IW,430)	KORG1193
	WRITE(IW,440)((SHEAD(I,J),J=1,20),I=1,9)	KORG1194
C		KORG1195
		KORG1196

Figure 12. Subroutine KORGI Program Listing (Continued)

C	READ SYSTEM SPECIFICATION CARD	KORG1197
C	READ(IR,17)ICARD	KORG1198
190	CONTINUE	KORG1199
	IF(CARD(1).EQ.MSTOP)GO TO 400	KORG1200
	IF(CARD(1).NE.MSYST)GO TO 240	KORG1201
	IF(IPRINT.LT.6)GO TO 210	KORG1202
	CALL MPRICARD,IW	KORG1203
	WRITE(IW,200)MS1,MS2,MS3,MS4,MAXN,MAXM	KORG1204
	I,NXM,NRM,NUM,NYM,MM,MP,MQ,MR,MR,NR,MS,MY,MTFB,MST,MT	KORG1205
200	FORMAT(1X,15(15,1X))	KORG1206
210	CONTINUE	KORG1207
	DECODE(4,220,CARD(1))D1,NSYSNO,D2	KORG1208
220	FORMAT(A2,11,A1)	KORG1209
	IF(NSYSNO.GT.NR)GO TO 260	KORG1210
	ISYS=ISYS+1	KORG1211
	IF(ISYS.GT.20)GO TO 260	KORG1212
	DO 24 I=1,5	KORG1213
	II=5+I	KORG1214
240	SOES(I)=CARD(II)	KORG1215
	DO 245 I=1,20	KORG1216
245	PHEAD(I)=SHEAD(NSYSNO,I)	KORG1217
	DO 25 I=1,20	KORG1218
	HEAD(I)=CARD(I)	KORG1219
	AHEAD(ISYS,I)=CARD(I)	KORG1220
250	SHEAD(NSYSNO,I)=CARD(I)	KORG1221
	NSYS(ISYS)=NSYSNO	KORG1222
	IF(IPRINT.LT.6)GO TO 256	KORG1223
	WRITE(IW,253)CARD	KORG1224
	WRITE(IW,253)HEAD	KORG1225
	WRITE(IW,253)PHEAD	KORG1226
	WRITE(IW,253)(SHEAD(NSYSNO,I),I=1,20)	KORG1227
253	FORMAT(1X,20A4)	KORG1228
256	CONTINUE	KORG1229
	READ(IR,17)SCODE	KORG1230
	IF(SCODE.EQ.MOLSA)GO TO 300	KORG1231
	IF(SCODE.EQ.MOTRA)GO TO 320	KORG1232
	IF(SCODE.EQ.MOQUA)GO TO 340	KORG1233
	IF(SCODE.EQ.MDINT)GO TO 340	KORG1234
	IF(SCODE.EQ.MDSIM)GO TO 360	KORG1235
	IF(SCODE.EQ.MDCON)GO TO 380	KORG1236
C		KORG1237
C	PRINT ERROR MESSAGE	KORG1238
C		KORG1239
260	CONTINUE	KORG1240
	WRITE(IW,280)	KORG1241
280	FORMAT(1M,/,1X,3)SYSTEM CARD SPECIFICATION ERROR)	KORG1242
	WRITE(IW,290)CARD	KORG1243
	WRITE(IW,290)SCODE	KORG1244
290	FORMAT(1X,20A4)	KORG1245
	WRITE(IW,295)NSYSNO,NR	KORG1246
295	FORMAT(1X,I2,1X,I2)	KORG1247
	STOP III	KORG1248
C		KORG1249
C	CALL OVERLAY LOADER TO LOAD REQUIRED PROGRAMS FOR EXECUTION	KORG1250
C		KORG1251
300	CONTINUE	KORG1252
	CALL OVERLAY(4MKON1,1,0)	KORG1253
	GO TO 180	KORG1254
320	CONTINUE	KORG1255
	CALL OVERLAY(4MKON1,2,0)	KORG1256
	GO TO 180	KORG1257
340	CONTINUE	KORG1258
	CALL OVERLAY(4MKON1,3,0)	KORG1259
	GO TO 180	KORG1260
360	CONTINUE	KORG1261
		KORG1262

Figure 12. Subroutine KORG1 Program Listing (Continued)

CALL OVERLAY(4MKON),4.0)	KORG1263
GO TO 180	KORG1264
380 CONTINUE	KORG1265
CALL OVERLAY(4MKON),5.0)	KORG1266
GO TO 180	KORG1267
C	KORG1268
C WRITE SYSTEM LABELS ON NFILE FOR CONTINUATION RUNS	KORG1269
C	KORG1270
400 CONTINUE	KORG1271
CALL FILE(JN,INSERT,LABEL)	KORG1272
WRITE(JN)((SHEAD(I,J),J=1,20),I=1,9)	KORG1273
CALL FILE(JN,INSERT,MARK)	KORG1274
WRITE(IM,430)	KORG1275
430 FORMAT(1M,/// <i>1X,34M</i> *** REFERENCE OF SYSTEM LABELS ***/// <i>)</i>	KORG1276
WRITE(IM,440)((SHEAD(I,J),J=1,20),I=1,9)	KORG1277
440 FORMAT(/// <i>1X,20A4</i> /// <i>)</i>	KORG1278
WRITE(IM,450)	KORG1279
450 FORMAT(1M,/// <i>1X,41M</i> *** LIST OF SYSTEM LABELS CREATED IN THIS, 18M RUN: ***/// <i>)</i>	KORG1280
WRITE(IM,440)((AHEAD(I,J),J=1,20),I=1,ISYS)	KORG1281
STOP	KORG1282
END	KORG1283
	KORG1284

Figure 12. Subroutine KORGI Program Listing (Concluded)

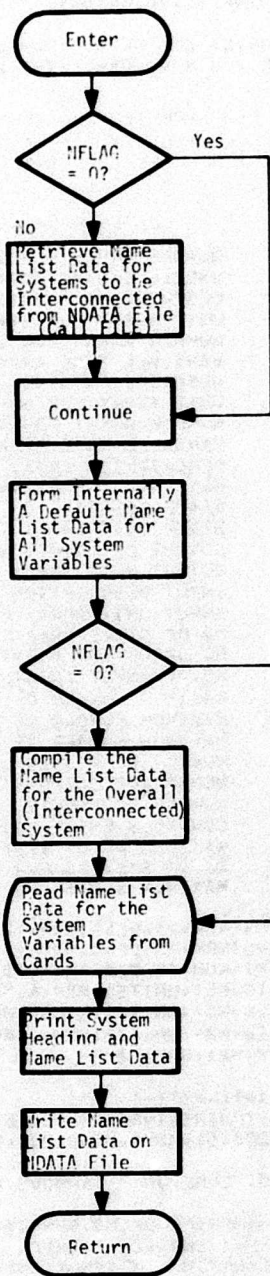


Figure 13. Subroutine NAMEL Flow Chart


```

REWIND JS
NRI=0
IF(NFLAG.EQ.0)GO TO 390
IF(IPRINT.EQ.6)CALL DERUG(1,4HNAME,4HL .0,0,1W)
C
C RETRIEVE NAME LIST DATA OF SUBSYSTEMS FOR COMBINING FROM FILE
C NDATA
C
DO 10 I=1,20
10 CARD(I)=HEAD(I)
DO 120 N=1,NR
NSY=NSYS(N)
DO 40 I=1,20
40 HEAD(I)=SHEAD(NSY,I)
CALL FILE(JN,LOCATF,HEAD)
READ(IN)NXN,NRN,NUN
1 (NNS(I),(VNS(I,J),J=1,2),
2 (DESS(I,J),J=1,10),(UNITS(I,J),J=1,4),I=1,NXN),
3 (NNO(I),(VNO(I,J),J=1,2),
4 (DESO(I,J),J=1,10),(UNITO(I,J),J=1,4),I=1,NRN),
5 (NNI(I),(VNI(I,J),J=1,2),
6 (DESI(I,J),J=1,10),(UNITI(I,J),J=1,4),I=1,NUN)
IF(IPRINT.EQ.6)CALL DERUG(2,4HNAME,4HL .0,0,1W)
NXX(N)=NXN
NRR(N)=NRN
NUU(N)=NUN
C
C IF THE SUBSYSTEM IS AN IMPLICIT MODEL, THEN SET NRI=NRN
C
IF(NSY.EQ.NR)NRI=NRN
DO 60 I=1,NXN
DO 50 J=1,10
50 DESS(I,J,N)=DESS(I,J)
DO 60 J=1,4
60 UNITS(I,J,N)=UNITS(I,J)
DO 80 I=1,NRN
DO 70 J=1,10
70 DESO(I,J,N)=DESO(I,J)
DO 80 J=1,4
80 UNITO(I,J,N)=UNITO(I,J)
DO 100 I=1,NUN
DO 90 J=1,10
90 DESI(I,J,N)=DESI(I,J)
DO 100 J=1,4
100 UNITI(I,J,N)=UNITI(I,J)
120 CONTINUE
DO 130 I=1,20
130 HEAD(I)=CARD(I)
380 CONTINUE
IF(IPRINT.EQ.6)CALL DERUG(3,4HNAME,4HL .0,0,1W)
C
C FORM A DEFAULT NAME LIST TABLE FOR THE SYSTEM
C
C FORM NAME LIST FOR STATES
C
DO 500 II=1,NX
NNS(II)=II
ENCOFF(4,420,VNS(II,1))HXP,II
420 FORMAT(A2,I2)
VNS(II,2)=HP
UNITS(II,1)=HBLANK
UNITS(II,2)=HBLANK
UNITS(II,3)=HBLANK
UNITS(II,4)=HBLANK
J=0
JFLAG=0
NAMEL 65
NAMEL 66
NAMEL 67
NAMEL 68
NAMEL 69
NAMEL 70
NAMEL 71
NAMEL 72
NAMEL 73
NAMEL 74
NAMEL 75
NAMEL 76
NAMEL 77
NAMEL 78
NAMEL 79
NAMEL 80
NAMEL 81
NAMEL 82
NAMEL 83
NAMEL 84
NAMEL 85
NAMEL 86
NAMEL 87
NAMEL 88
NAMEL 89
NAMEL 90
NAMEL 91
NAMEL 92
NAMEL 93
NAMEL 94
NAMEL 95
NAMEL 96
NAMEL 97
NAMEL 98
NAMEL 99
NAMEL100
NAMEL101
NAMEL102
NAMEL103
NAMEL104
NAMEL105
NAMEL106
NAMEL107
NAMEL108
NAMEL109
NAMEL110
NAMEL111
NAMEL112
NAMEL113
NAMEL114
NAMEL115
NAMEL116
NAMEL117
NAMEL118
NAMEL119
NAMEL120
NAMEL121
NAMEL122
NAMEL123
NAMEL124
NAMEL125
NAMEL126
NAMEL127
NAMEL128
NAMEL129
NAMEL130

```

Figure 14. Subroutine NAMEL Program Listing (Continued)

440	CONTINUE	NAMEL131
	IF(J.GF.5)GO TO 460	NAMEL132
	IF(JFLAG.EQ.1)GO TO 440	NAMEL133
	J=J+1	NAMEL134
	IF(SDFS(J).EQ.HBLANK)JFLAG=JFLAG+1	NAMEL135
	DESS(1,J)=SDES(J)	NAMEL136
	GO TO 440	NAMEL137
460	CONTINUE	NAMEL138
	J=J+1	NAMEL139
	DESS(1,J)=MSTAT	NAMEL140
	J=J+1	NAMEL141
	ENCODE(4,470,DESS(1,J))ME,11,HH	NAMEL142
470	FORMAT(A),12,41)	NAMEL143
480	CONTINUE	NAMEL144
	J=J+1	NAMEL145
	IF(J.GT.10)GO TO 500	NAMEL146
	DESS(1,J)=HBLANK	NAMEL147
	GO TO 480	NAMEL148
500	CONTINUE	NAMEL149
	IF(IPRINT.EQ.6)CALL DFHUG(4,4,NAME,4ML,0,0,1W)	NAMEL150
C		NAMEL151
C	FORM NAME LIST FOR OUTPUTS	NAMEL152
C		NAMEL153
	NRJ=NR-NR1	NAMEL154
	DO 70 I=1,NR	NAMEL155
	VNO(I)=I	NAMEL156
	ENCODE(4,420,VNO(I),1),HRP,11	NAMEL157
	VNO(I,2)=HP	NAMEL158
	UNITO(I,1)=HBLANK	NAMEL159
	UNITO(I,2)=HBLANK	NAMEL160
	UNITO(I,3)=HBLANK	NAMEL161
	UNITO(I,4)=HBLANK	NAMEL162
	J=0	NAMEL163
	JFLAG=0	NAMEL164
C		NAMEL165
C	FORM NAME LIST FOR THE IMPLICIT MODEL ERROR RESPONSES	NAMEL166
C		NAMEL167
	IF(I.LE.NRJ)GO TO 640	NAMEL168
	NJ=I-NRJ	NAMEL169
	DECODE(4,610,DESO(NJ,1,KR),11,1,1)	NAMEL170
610	FORMAT(A3,A1)	NAMEL171
	ENCODE(4,615,DESO(11,1))ME,11	NAMEL172
615	FORMAT(A1,A3)	NAMEL173
	ENCODE(4,615,DESO(11,2))11,HHHH	NAMEL174
	DESO(11,3)=HMODE	NAMEL175
	DESO(11,4)=HMRFO	NAMEL176
	DESO(11,5)=HLLRE	NAMEL177
	DESO(11,6)=HRROR	NAMEL178
	DESO(11,7)=HBLANK	NAMEL179
	DESO(11,8)=HBLANK	NAMEL180
	DESO(11,9)=HBLANK	NAMEL181
	DESO(11,10)=HBLANK	NAMEL182
	DO 62 J=1,4	NAMEL183
620	UNITO(11,J)=UNITO(NJ,J,KR)	NAMEL184
	GO TO 730	NAMEL185
640	CONTINUE	NAMEL186
	IF(J.GF.5)GO TO 660	NAMEL187
	IF(JFLAG.EQ.1)GO TO 640	NAMEL188
	J=J+1	NAMEL189
	IF(SDFS(J).EQ.HBLANK)JFLAG=JFLAG+1	NAMEL190
	DESO(11,J)=SDES(J)	NAMEL191
	GO TO 640	NAMEL192
660	CONTINUE	NAMEL193
	J=J+1	NAMEL194
	DESO(11,J)=HQUITP	NAMEL195
	J=J+1	NAMEL196

Figure 14. Subroutine NAMEL Program Listing (Continued)

ENCODE(4,420,DESO(II,J))MUT,II	NAMEL197
680 CONTINUE	NAMEL198
J=J+1	NAMEL199
IF(J.GT.10)GO TO 700	NAMEL200
DESO(II,J)=HBLANK	NAMEL201
GO TO 680	NAMEL202
700 CONTINUE	NAMEL203
IF(IPRINT.EQ.6)CALL DFRUG(5,4HNAME,4HL .0,0,1W)	NAMEL204
C	NAMEL205
C FORM NAME LIST FOR INPUTS	NAMEL206
C	NAMEL207
DO 900 II=1,NU	NAMEL208
NNI(II)=II	NAMEL209
ENCODE(4,420,VNI(II,1))MUP,II	NAMEL210
VNI(II,2)=HP	NAMEL211
UNITI(II,1)=HBLANK	NAMEL212
UNITI(II,2)=HBLANK	NAMEL213
UNITI(II,3)=HBLANK	NAMEL214
UNITI(II,4)=HBLANK	NAMEL215
J=0	NAMEL216
JFLAG=0	NAMEL217
840 CONTINUE	NAMEL218
IF(J.GE.5)GO TO 860	NAMEL219
IF(JFLAG.EQ.1)GO TO 860	NAMEL220
J=J+1	NAMEL221
IF(SDES(J).EQ.HBLANK)JFLAG=JFLAG+1	NAMEL222
DESI(II,J)=SDES(J)	NAMEL223
GO TO 840	NAMEL224
860 CONTINUE	NAMEL225
J=J+1	NAMEL226
DESI(II,J)=HINPU	NAMEL227
J=J+1	NAMEL228
ENCODE(4,470,DESI(II,J))MT,II,MB	NAMEL229
880 CONTINUE	NAMEL230
J=J+1	NAMEL231
IF(J.GT.10)GO TO 900	NAMEL232
DESI(II,J)=HBLANK	NAMEL233
GO TO 880	NAMEL234
900 CONTINUE	NAMEL235
IF(IPRINT.EQ.6)CALL DFRUG(6,4HNAME,4HL .0,0,1W)	NAMEL236
IF(INFLAG.EQ.0)GO TO 1220	NAMEL237
C	NAMEL238
C COMBINE THE NAME LIST DATA OF SUBSYSTEMS AND OBTAIN THE NAME LIST	NAMEL239
C DATA FOR THE COMBINED SYSTEM	NAMEL240
C	NAMEL241
1000 CONTINUE	NAMEL242
II=0	NAMEL243
DO 1040 K=1,KB	NAMEL244
NXXK=1,XX(K)	NAMEL245
DO 1040 I=1,NXXK	NAMEL246
II=II+1	NAMEL247
NNS(II)=II	NAMEL248
ENCODE(4,420,VNS(II,1))MXP,II	NAMEL249
VNS(II,2)=HP	NAMEL250
DO 1020 J=1,10	NAMEL251
1020 DESS(II,J)=DESSS(I,J,K)	NAMEL252
DO 1040 J=1,4	NAMEL253
1040 UNITS(II,J)=UNITSS(I,J,K)	NAMEL254
C	NAMEL255
C READ NAME LIST DATA FOR OUTPUTS OBTAINABLE FROM	NAMEL256
C INTERCONNECTION EQUATIONS WRITTEN ON SCRATCH FILE JS	NAMEL257
C BY SUBROUTINE SIMK	NAMEL258
C	NAMEL259
READ(15,160)CARD	NAMEL260
IF(CARD(1).NE.HOUTP)GO TO 1320	NAMEL261
1050 CONTINUE	NAMEL262

Figure 14. Subroutine NAMEL Program Listing (Continued)

<pre> 1060 READ(JS,1060)II,K,I FORMAT(3I2) IF(II,EQ,-1)GO TO 1110 NNO(II)=II ENCODE(4,420,VNO(II,1))HRP,II VNO(II,2)=HP DO 1080 J=1,10 1080 DESO(II,J)=DESOO(I,J,K) DO 1100 J=1,4 1100 UNITO(II,J)=UNITOO(I,J,K) GO TO 1050 1110 CONTINUE IF(IP=INT.EQ.6)CALL DEBUG(7,4HNAME,4HL ,0,0,1W) C C READ NAME LIST DATA FOR INPUTS OBTAINABLE FROM C INTERCONNECTION EQUATIONS WRITTEN ON SCRATCH FILE JS C BY SUBROUTINE SIMK C READ(JS,160)CARD IF(CARD(1).NE.MINPI)GO TO 1320 1120 CONTINUE READ(JS,1060)II,K,I IF(II,EQ,-1)GO TO 1170 NNI(II)=II ENCODE(4,420,VNI(II,1))HUP,II VNI(II,2)=HP DO 1140 J=1,10 1140 DESI(II,J)=DESI(I,J,K) DO 1160 J=1,4 1160 UNITI(II,J)=UNITII(I,J,K) GO TO 1120 1170 CONTINUE READ(JS,160)CARD IF(CARD(1).NE.MEND)GO TO 1320 C C READ NAME LIST DATA FROM CARDS C 1220 CONTINUE IF(IP=INT.EQ.6)CALL DEBUG(8,4HNAME,4HL ,0,0,1W) READ(IR,160)CARD 160 FORMAT(20A4) IF(CARD(1).EQ.MEND)GO TO 1340 IF(CARD(1).EQ.MSTAT)GO TO 1240 IF(CARD(1).EQ.MOUTP)GO TO 1260 IF(CARD(1).EQ.MINPI)GO TO 1300 GO TO 200 C C READ NAME LIST DATA FOR STATES C 1240 CONTINUE READ(IR,280)NNNN,(VN(J),J=1,2),(DES(J),J=1,10),(UNIT(J),J=1,4) 280 FORMAT(12,6X,2A4,4X,10A4,4X,4A4) IF(NNNN,EQ,-1)GO TO 1220 NNS(NNNN)=NNNN DO 1245 J=1,2 1245 VNS(NNNN,J)=VN(J) DO 1250 J=1,10 1250 DESS(NNNN,J)=DES(J) DO 1255 J=1,4 1255 UNITS(NNNN,J)=UNIT(J) GO TO 1240 C C READ NAME LIST DATA FOR OUTPUTS C 1260 CONTINUE READ(IR,280)NNNN,(VN(J),J=1,2),(DES(J),J=1,10),(UNIT(J),J=1,4) </pre>	<pre> NAMEL263 NAMEL264 NAMEL265 NAMEL266 NAMEL267 NAMEL268 NAMEL269 NAMEL270 NAMEL271 NAMEL272 NAMEL273 NAMEL274 NAMEL275 NAMEL276 NAMEL277 NAMEL278 NAMEL279 NAMEL280 NAMEL281 NAMEL282 NAMEL283 NAMEL284 NAMEL285 NAMEL286 NAMEL287 NAMEL288 NAMEL289 NAMEL290 NAMEL291 NAMEL292 NAMEL293 NAMEL294 NAMEL295 NAMEL296 NAMEL297 NAMEL298 NAMEL299 NAMEL300 NAMEL301 NAMEL302 NAMEL303 NAMEL304 NAMEL305 NAMEL306 NAMEL307 NAMEL308 NAMEL309 NAMEL310 NAMEL311 NAMEL312 NAMEL313 NAMEL314 NAMEL315 NAMEL316 NAMEL317 NAMEL318 NAMEL319 NAMEL320 NAMEL321 NAMEL322 NAMEL323 NAMEL324 NAMEL325 NAMEL326 NAMEL327 NAMEL328 </pre>
--	--

Figure 14. Subroutine NAMEL Program Listing (Continued)

```

IF (NNNN.EQ.-1) GO TO 1220
NNO (N'NNN)=NNNN
DO 1245 J=1,2
1265 VNO (N'NNN,J)=VN (J)
DO 1270 J=1,10
1270 DESO (N'NNN,J)=DES (J)
DO 1275 J=1,4
1275 UNITO (N'NNN,J)=UNIT (J)
GO TO 1260
C
C READ NAME LIST DATA FOR INPUTS
C
1300 CONTINUE
READ (1R,280) NNNN, (VN (J), J=1,2), (DES (J), J=1,10), (UNIT (J), J=1,4)
IF (NNNN.EQ.-1) GO TO 1220
NNI (N'NNN)=NNNN
DO 1285 J=1,2
1285 VNI (N'NNN,J)=VN (J)
DO 1290 J=1,10
1290 DESI (N'NNN,J)=DES (J)
DO 1295 J=1,4
1295 UNITI (N'NNN,J)=UNIT (J)
GO TO 1300
1340 CONTINUE
IF (IPRINT.EQ.6) CALL DEBUG (9,4HNAME,4HL ,0,0,1W)
C
C PRINT HEADING AND NAME LIST DATA
C
IF (IPRINT.LT.2) GO TO 1540
CALL HPR (HEAD,1W)
WRITE (9,1360) NX,NP,NU
1360 FORMAT (//,1X,18HNUMBER OF STATES =,12,/,1X,
18HNUMBER OF OUTPUTS=,12,/,1X,18HNUMBER OF INPUTS =,12,/)
WRITE (1W,1380)
1380 FORMAT (//,20X,23H*** NAME LIST TABLE ***,/)
WRITE (1W,1400)
1400 FORMAT (/,1X,8HVARIABLE,6H NAME ,6X,13H DESCRIPTION ,
131X,6H UNIT ,/)
IF (IPRINT.EQ.6) CALL DEBUG (10,4HNAME,4HL ,0,0,1W)
C
C PRINT NAME LIST DATA FOR STATES
C
WRITE (1W,1460)
1460 FORMAT (/,1X,6HSTATE ,/)
WRITE (1W,1480) (NNS (I), (VNS (I,J), J=1,2), (DESS (I,J), J=1,10),
I
(UNITS (I,J), J=1,4), I=1,NX)
1480 FORMAT (1X,12,6X,2A4,4X,10A4,4X,4A4)
C
C PRINT NAME LIST DATA FOR OUTPUTS
C
WRITE (1W,1500)
1500 FORMAT (/,1X,6HOUTPUT,/)
WRITE (1W,1480) (NNO (I), (VNO (I,J), J=1,2), (DESO (I,J), J=1,10),
I
(UNITO (I,J), J=1,4), I=1,NR)
C
C PRINT NAME LIST DATA FOR INPUTS
C
WRITE (1W,1520)
1520 FORMAT (/,1X,6HINPUT ,/)
WRITE (1W,1480) (NNI (I), (VNI (I,J), J=1,2), (DESI (I,J), J=1,10),
I
(UNITI (I,J), J=1,4), I=1,NU)
1540 CONTINUE
IF (IPRINT.EQ.6) CALL DEBUG (11,4HNAME,4HL ,0,0,1W)
C
C WRITE NAME LIST DATA ON DISK FILE
C

```

```

NAMEL329
NAMEL330
NAMEL331
NAMEL332
NAMEL333
NAMEL334
NAMEL335
NAMEL336
NAMEL337
NAMEL338
NAMEL339
NAMEL340
NAMEL341
NAMEL342
NAMEL343
NAMEL344
NAMEL345
NAMEL346
NAMEL347
NAMEL348
NAMEL349
NAMEL350
NAMEL351
NAMEL352
NAMEL353
NAMEL354
NAMEL355
NAMEL356
NAMEL357
NAMEL358
NAMEL359
NAMEL360
NAMEL361
NAMEL362
NAMEL363
NAMEL364
NAMEL365
NAMEL366
NAMEL367
NAMEL368
NAMEL369
NAMEL370
NAMEL371
NAMEL372
NAMEL373
NAMEL374
NAMEL375
NAMEL376
NAMEL377
NAMEL378
NAMEL379
NAMEL380
NAMEL381
NAMEL382
NAMEL383
NAMEL384
NAMEL385
NAMEL386
NAMEL387
NAMEL388
NAMEL389
NAMEL390
NAMEL391
NAMEL392
NAMEL393
NAMEL394

```

Figure 14. Subroutine NAMEL Program Listing (Continued)

CALL FILE(JN,INSERT,HEAD)	NAMEL395
WRITE(JN)NX,NR,NU,	NAMEL396
1 (NNS(I),(VNS(I,J),J=1,2),	NAMEL397
2 (DESS(I,J),J=1,10),(UNITS(I,J),J=1,4),I=1,NX),	NAMEL398
3 (NNO(I),(VNO(I,J),J=1,2),	NAMEL399
4 (DESO(I,J),J=1,10),(UNITO(I,J),J=1,4),I=1,NR),	NAMEL400
5 (NNI(I),(VNI(I,J),J=1,2),	NAMEL401
6 (DESI(I,J),J=1,10),(UNITI(I,J),J=1,4),I=1,NU)	NAMEL402
CALL FILE(JN,INSERT,MARK)	NAMEL403
IF(IPRINT.EQ.6)CALL DEBUG(12,4HNAME,4ML ,0,0,IW)	NAMEL404
RETURN	NAMEL405
C PRINT ERROR MESSEGE	NAMEL406
C	NAMEL407
C	NAMEL408
200 CONTINUE	NAMEL409
WRITE(IW,220)	NAMEL410
220 FORMAT(1H1,/,1X,37HDATA CONTROL CARD SPECIFICATION ERROR)	NAMEL411
STOP 111	NAMEL412
1320 CONTINUE	NAMEL413
WRITE(IW,1330)	NAMEL414
1330 FORMAT(1H1,/,1X,36HERROR IN DATA PROVIDED BY SIMK)	NAMEL415
STOP 111	NAMEL416
END	NAMEL417

Figure 14. Subroutine NAMEL Program Listing (Concluded)

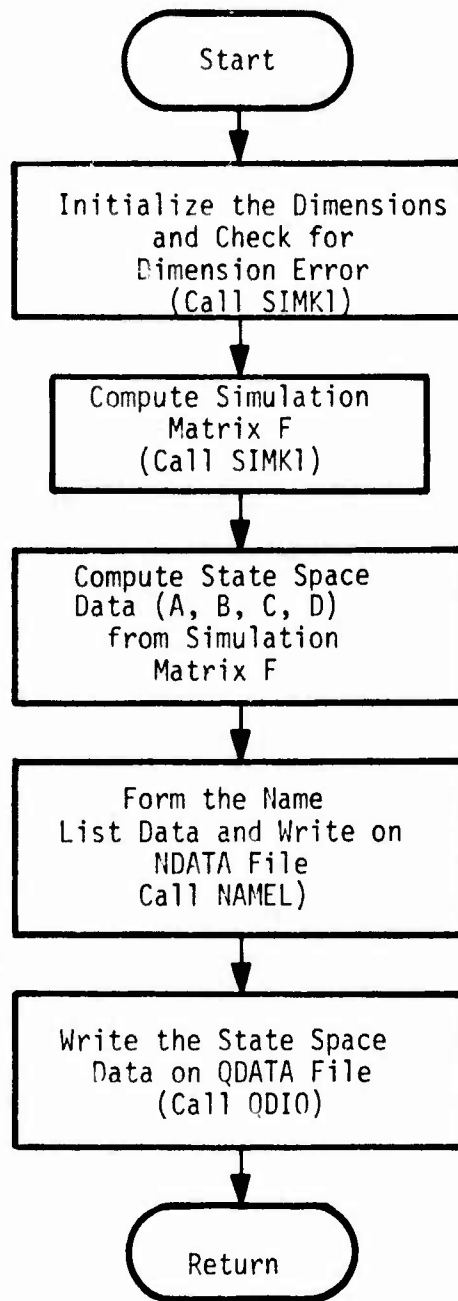


Figure 15. Subroutine STAMK1 Flow Chart


```

COMMON /SCI/ SI(1)
C DIMENSION DESSS(NM,10,MB),UNITSS(NM,4,MR)
C DIMENSION DFS00(NM,10,MB),UNIT00(NM,4,MR)
C DIMENSION DESII(NUM,10,MB),UNITII(NUM,4,MR)
C DIMENSION NXX(MR),NRR(MR),NUU(MB)
IF(IPRINT.EQ.6)CALL DFRUG(1.4HSTAM,4HK1 .1.0,1W)
L1=1 & L2=L1+NXM*MR*10 & L3=L2+NXM*MR*4 & L4=L3+NRM*MR*10
L5=L4+NRM*MR*4 & L6=L5+NUM*MR*10 & L7=L6+NUM*MR*4
L8=L7*MB & L9=L8*MR & L10=L9*MR
IF(L1.GT.MS1)
ICALL DERRM(L10,MS2,MS3,MS4,MS1,MS2,MS3,MS4,1.0,4HSTAM,4HK1 .1W)
IF(IPRINT.EQ.6)CALL DFRUG(2.4HSTAM,4HK1 .1.0,1W)
NR1=0 & NR2=0 & NR3=0 & NU1=0 & NU2=0 & NU3=0
NXA=0 & NRA=0 & NUA=0
EPSF=1.0E-30 & T=0.0 & NFLAG=0
IF((IPRINT.EQ.7).OR.(IPRINT.GT.4))CALL MPR(HEAD,1W)
C
C INITIALIZING CALL TO SUBROUTINE STAMK1
C
INIT=
NX=0 & NY=0 & NP=0 & NU=0
N1=1 & N2=N1*NX & N3=N2*NY & N4=N3*NX
CALL S[MK1](W(N1),W(N2),W(N3),W(N4),V(N1),V(N2),V(N3),
INX,NY,NR,NU,INIT,T,MS1,MS2,MS3,MS4)
IF(IPRINT.EQ.6)CALL DEBUG(3.4HSTAM,4HK1 .1.0,1W)
C
C CHECK FOR DIMENSION ERROR
C
INIT = 1
M=2*NX*NY*NU
N=NX*Y*NR
IF((NX.GT.NXM).OR.(NR.GT.NRM).OR.(NU.GT.NUM).OR.(NY.GT.NYM))
ICALL DERRMS(NX,NR,NU,NY,NXM,NRM,NUM,NYM,1.0,4HSTAM,4HK1 .1W)
N1=1 & N2=N1*NX & N3=N2*NY & N4=N3*NX
DO 101 J=1,M
101 W(J)=1.
DO 501 J=1,M
W(J)=1.
CALL S[MK1](W(N1),W(N2),W(N3),W(N4),V(N1),V(N2),V(N3),
INX,NY,NR,NU,INIT,T,MS1,MS2,MS3,MS4)
W(J)=1.6
DO 501 I=1,N
501 F(I,J)=V(I)
C
C COMPUTE THE SIMULATION MATRIX
C
NV=NX*NY
IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,4,4,T,4HSTM)
DO 51 I=1,NV
DO 52 J=1,NV
52 F(I,J)=-F(I,J)
51 F(I,I)=F(I,I)+1.
CALL TDINVR(ISOL,IDSOL,NV,-4,F,MAXN,KDIM,DET)
IR=NV+1
IE=NV+NR
JR=18
JE=M
DO 53 I=IR,IE
DO 53 J=JR,JE
DO 53 K=1,NV
53 F(I,J)=F(I,J)+F(I,K)*F(K,J)
DO 53 I=1,IE
DO 53 J=1,JE
IF(ABS(F(I,J)).LE.EPSF) F(I,J) = 0.0
CONTINUE
530 IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,4,4,T,4HSTM)

```

```

STAMK165
STAMK166
STAMK167
STAMK168
STAMK169
STAMK170
STAMK171
STAMK172
STAMK173
STAMK174
STAMK175
STAMK176
STAMK177
STAMK178
STAMK179
STAMK180
STAMK181
STAMK182
STAMK183
STAMK184
STAMK185
STAMK186
STAMK187
STAMK188
STAMK189
STAMK190
STAMK191
STAMK192
STAMK193
STAMK194
STAMK195
STAMK196
STAMK197
STAMK198
STAMK199
STAMK100
STAMK101
STAMK102
STAMK103
STAMK104
STAMK105
STAMK106
STAMK107
STAMK108
STAMK109
STAMK110
STAMK111
STAMK112
STAMK113
STAMK114
STAMK115
STAMK116
STAMK117
STAMK118
STAMK119
STAMK120
STAMK121
STAMK122
STAMK123
STAMK124
STAMK125
STAMK126
STAMK127
STAMK128
STAMK129
STAMK130

```

Figure 16. Subroutine STAMK1 Program Listing (Continued)

C		STAMK131
C	FORM A,B,C,D MATRICES	STAMK132
C		STAMK133
	J1=NV*1	STAMK134
	J2=NV*NX	STAMK135
	J3=J1*NX	STAMK136
	J4=J2*NU	STAMK137
	I1=NV*1	STAMK138
	I2=NV*NR	STAMK139
	DO 6001 I=1,NX	STAMK140
	DO 6001 J=J1,J2	STAMK141
	JJ=J-J*1	STAMK142
6001	A(I,J)=F(I,J)	STAMK143
	DO 6002 I=1,NX	STAMK144
	DO 6002 J=J3,J4	STAMK145
	JJ=J-J*1	STAMK146
6002	B(I,J)=F(I,J)	STAMK147
	DO 6003 I=I1,I2	STAMK148
	IJ=I-I*1	STAMK149
	DO 6003 J=J1,J2	STAMK150
	JJ=J-J*1	STAMK151
6003	C(IJ,JJ)=F(I,J)	STAMK152
	DO 6004 I=I1,I2	STAMK153
	IJ=I-I*1	STAMK154
	DO 6004 J=J3,J4	STAMK155
	JJ=J-J*1	STAMK156
6004	D(IJ,JJ)=F(I,J)	STAMK157
	IF(IPRINT.EQ.6)CALL DEBUG(4,4HSTAM,4H<1 .1,0,1W)	STAMK158
C		STAMK159
C	READ AND UPDATE NAME LIST DATA	STAMK160
C		STAMK161
	KB=NMAX	STAMK162
	CALL NAMEL(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,	STAMK163
	IDESI,UNITI,S1(L1),S1(L2),S1(L3),S1(L4),S1(L5),S1(L6),	STAMK164
	2S1(L7),S1(L8),S1(L9),NAM,NRM,NUM,NX,NR,NU,NFLAG,MB,KB,NB)	STAMK165
	IF(IPRINT.EQ.6)CALL DEBUG(5,4HSTAM,4H<1 .1,0,1W)	STAMK166
C		STAMK167
C	WRITE QUADRUPLE DATA ON FILE ODATA	STAMK168
C		STAMK169
	IO=0	STAMK170
	MFLAG=2	STAMK171
	NXA=NX \$ NRA=NR \$ NUA=NU	STAMK172
	CALL DDIO(A,B,C,D,A,NX,NR,NI,NM,NRM,NUM,NXA,NRA,NUA,	STAMK173
	1NR1,NR2,NR3,NU1,NU2,NU3,T,IO,IPRINT,IN,JO,HEAD,MARK,	STAMK174
	ZLOCATF,NULL,INSERT,MFLAG)	STAMK175
	IF(IPRINT.EQ.6)CALL DEBUG(6,4HSTAM,4H<1 .1,0,1W)	STAMK176
	RETURN	STAMK177
	END	STAMK178

Figure 16. Subroutine STAMK1 Program Listing (Concluded)

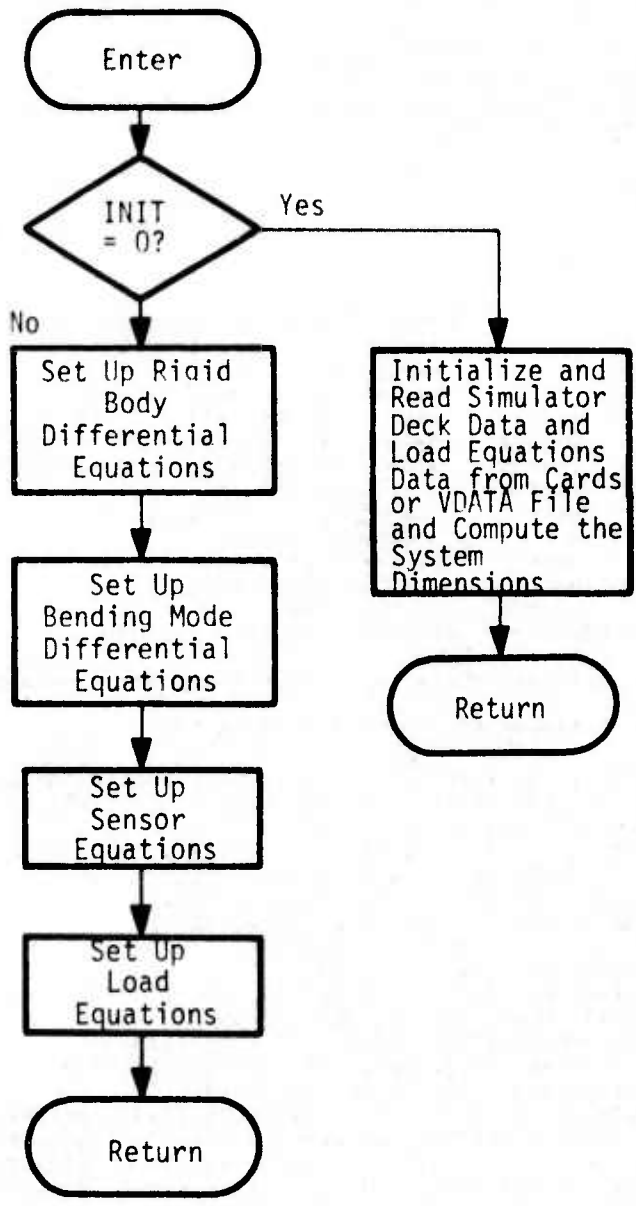


Figure 17. Subroutine SIMK1 Flow Chart

```

SUBROUTINE SIMK1(XDOT,Y,X,U,XDOTL,YL,RL,NX,NY,NR,NU,INIT,T,
1MS1,MS2,MS3,MS4)
C
C
C PURPOSE - TO READ SIMULATOR MATRIX DATA FROM LSA AND
C TO IMPLEMENT STANDARD LSA EQUATIONS
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C DATE WRITTEN - MAY 1975
C
C SUBPROGRAMS CALLED
C   DEBUG
C   INPT1
C   MPWS1
C
C ARGUMENTS LIST
C   XDOT      ARRAY FOR STATE DERIVATIVES
C   Y         ARRAY FOR Y EQUATIONS
C   X         ARRAY FOR STATES
C   U         ARRAY FOR EXTERNAL INPUTS
C   XDOTL     OUTPUT  ARRAY FOR DERIVATIVE OF STATE
C   YL        OUTPUT  ARRAY FOR Y EQUATION VARIABLES
C   RL        OUTPUT  ARRAY FOR EXTERNAL RESPONSE VARIABLES
C   NX        OUTPUT  NUMBER OF STATES
C   NY        OUTPUT  NUMBER OF Y EQUATIONS
C   NR        OUTPUT  NUMBER OF OUTPUTS
C   NU        OUTPUT  NUMBER OF INPUTS
C   INIT      INPUT   INITIAL MODE FLAG
C   T         OUTPUT  SAMPLE TIME
C
C OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
C DIMENSION XDOT(NX),Y(NY),X(NX),U(NU),XDOTL(NX),YL(NY),RL(NR)
C
C DIMENSION STATEMENT FOR THE MATRIX DATA FROM LSA
C
COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NUILL,MARK(20),JN,JQ,JS
REAL LVPO,LVP1,LRO,LR1,LUF0,LUE1,LUE2,LDELS0,LDELS1,LDELS2
REAL LUG0,LUG1,LVG0,LVG1,LWG0,LWG1
COMMON /SC2/ VPVP0(6,6),VPVP1(6,6),VPR0(6,3),VPR1(6,3)
1.VPUF0(6,3),VPUF1(6,3),VPIE2(6,3),VPDELS0(6,3)
2.VPDELS1(6,3),VPDELS2(6,3),VPIG0(6,3),VPIG1(6,3)
3.VPVG0(6,3),VPVG1(6,3),VPWG0(6,3),VPWG1(6,3)
4.RVPO(3,6),RVPI(3,6),PR0(3,3),PRI(3,3)
5.UFVP0(30,6),UFVP1(30,6),UEP0(30,3)
6.UFR1(30,3),UFEU0(30,30),UEUE1(30,30),UEUE2(30,30)
7.UEDEL0(30,3),UEDELS1(30,3),UEDELS2(30,3)
8.UEUG0(30,3),UEUG1(30,3),UEVG0(30,3),UEVG1(30,3)
9.UEWG0(30,3),UEWG1(30,3),TVPO(9,6),TVPI(9,6),TRO(9,3)
A.TRI(9,3),TUF0(9,30),TUE1(9,30),TUF2(9,30)
B.TDELS0(9,3),TDELS1(9,3),TDELS2(9,3),TUG0(9,3),TUG1(9,3)
C.TVG0(9,3),TVG1(9,3),TWG0(9,3),TWG1(9,3),LVP0(15,6),LVP1(15,6)
D.LRO(15,3),LR1(15,3),LUF0(15,30),LUE1(15,30),LUE2(15,30)
E.LDELS0(15,3),LDELS1(15,3),LDELS2(15,3),LUG0(15,3),LUG1(15,3)
F.LVG0(15,3),LVG1(15,3),LWG0(15,3),LWG1(15,3)
G.RANDING(1,3),UNITY(30,30)
DIMENSION JHEAD(120),IDRM(120),IDCM(120)
DIMENSION ICARD(8),IHEAD(A)
DIMENSION SC(1)
EQUIVALENCE (SC(1),VPVP0(1,1))
IF(IPRINT.EQ.6)CALL DEBUG(1,4HSIMK,4H)  .1,0,IW)
IF(INIT.NE.0) GO TO 150
C
C INITIALIZE AND SET MAX DIMENSIONS FOR SIMULATOR MATRIX DATA
C
IEND=10HEND
NXVP=0 % NXR=0 % NXUE=0
SIMK1 2
SIMK1 3
SIMK1 4
SIMK1 5
SIMK1 6
SIMK1 7
SIMK1 8
SIMK1 9
SIMK1 10
SIMK1 11
SIMK1 12
SIMK1 13
SIMK1 14
SIMK1 15
SIMK1 16
SIMK1 17
SIMK1 18
SIMK1 19
SIMK1 20
SIMK1 21
SIMK1 22
SIMK1 23
SIMK1 24
SIMK1 25
SIMK1 26
SIMK1 27
SIMK1 28
SIMK1 29
SIMK1 30
SIMK1 31
SIMK1 32
SIMK1 33
SIMK1 34
SIMK1 35
SIMK1 36
SIMK1 37
SIMK1 38
SIMK1 39
SIMK1 40
SIMK1 41
SIMK1 42
SIMK1 43
SIMK1 44
SIMK1 45
SIMK1 46
SIMK1 47
SIMK1 48
SIMK1 49
SIMK1 50
SIMK1 51
SIMK1 52
SIMK1 53
SIMK1 54
SIMK1 55
SIMK1 56
SIMK1 57
SIMK1 58
SIMK1 59
SIMK1 60
SIMK1 61
SIMK1 62
SIMK1 63
SIMK1 64
SIMK1 65

```

Figure 18. Subroutine SIMK1 Program Listing


```

JMFAD(101)=10HL/VG0      & JMFAD(102)=10HL/VG1      SIMK1132
JMFAD(103)=10HL/WG0      & JMFAD(104)=10HL/WG1      SIMK1133
JMFAD(105)=10HL/UGS0     & JMFAD(106)=10HL/UGS1     SIMK1134
JMFAD(107)=10HL/VGS0     & JMFAD(108)=10HL/VGS1     SIMK1135
JMFAD(109)=10HL/WGS0     & JMFAD(110)=10HL/WGS1     SIMK1136
C                                                                    SIMK1137
C MISCELLANEOUS MATRIX NAMES                                       SIMK1138
C                                                                    SIMK1139
C JMFAD(111)=10M(HANDING) & JMFAD(112)=10M*FINISHED*           SIMK1140
C                                                                    SIMK1141
C SET UP MAX ROW AND COL DIMENSIONS FOR SIMULATOR MATRIX DATA   SIMK1142
C                                                                    SIMK1143
DO 4 I=1,16                                                         SIMK1144
  I1=I                                                                SIMK1145
  I2=16*I                                                            SIMK1146
  I3=32*I                                                            SIMK1147
  I4=48*I                                                            SIMK1148
  I5=64*I                                                            SIMK1149
  IDRM(I1)=NXVPM & IDRM(I2)=NXRM & IDRM(I3)=NXUEM               SIMK1150
  IDRM(I4)=NRTM & IDRM(I5)=NRLM                                    SIMK1151
4 CONTINUE                                                            SIMK1152
DO 6 I=1,5                                                           SIMK1153
  J=(16*I-16)                                                        SIMK1154
  IDCM(J+1)=NXVPM & IDCM(J+2)=NXVPM                               SIMK1155
  IDCM(J+3)=NXRM & IDCM(J+4)=NXRM                                 SIMK1156
  IDCM(J+5)=NXUFM & IDCM(J+6)=NXUFM & IDCM(J+7)=NXUFM         SIMK1157
  IDCM(J+8)=NUCM & IDCM(J+9)=NUCM & IDCM(J+10)=NUCM           SIMK1158
  IDCM(J+11)=NUGM & IDCM(J+12)=NUGM & IDCM(J+13)=NUGM        SIMK1159
  IDCM(J+14)=NUGM & IDCM(J+15)=NUGM & IDCM(J+16)=NUGM        SIMK1160
6 CONTINUE                                                            SIMK1161
  IDRM(81)=NRM & IDCM(81)=NLM                                       SIMK1162
  IDRM(82)=30 & IDCM(82)=30                                          SIMK1163
C                                                                    SIMK1164
C CHECK IF SCRATCH ARRAY SIZE IS SUFFICIENT                          SIMK1165
C                                                                    SIMK1166
N=0                                                                    SIMK1167
DO 8 I=1,20                                                         SIMK1168
8 N=N+IDRM(I)*IDCM(I)                                               SIMK1169
DO 9 I=33,82                                                         SIMK1170
9 N=N+IDRM(I)*IDCM(I)                                               SIMK1171
  IF(N.GT.MS2)                                                        SIMK1172
  ICALL DFRM(MS1,N,MS3,MS4,MS1,MS2,MS3,MS4,1,0,4HSIMK,4HI  .IW) SIMK1173
  IF(IPRINT.EQ.6)CALL DFRUG(2,4HSIMK,4HI  .1,0, IW)                SIMK1174
C                                                                    SIMK1175
C INITIALIZE THE MEMORY WHERE SIMULATOR MATRIX DATA IS STORED    SIMK1176
C                                                                    SIMK1177
DO 10 I=1,N                                                         SIMK1178
10 SC(I)=0.0                                                         SIMK1179
C                                                                    SIMK1180
C READ LSA SIMULATOR DECK IDENTIFICATION CARD                       SIMK1181
C                                                                    SIMK1182
12 CONTINUE                                                            SIMK1183
  READ(IP,16)ICARD                                                    SIMK1184
16 FORMAT(8A10)                                                       SIMK1185
  IF(ICARD(1).EQ.IEND)RETURN                                          SIMK1186
  IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))WRITE(IW,22)                   SIMK1187
  IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))WRITE(IW,24)ICARD              SIMK1188
22 FORMAT(//,20X,27M*** LSA - FLEXSTAR DATA ***,//)               SIMK1189
24 FORMAT(//,1X,8A10,//)                                             SIMK1190
  DO 28 I=1,8                                                         SIMK1191
28 IHEAD(I)=ICARD(I)                                                 SIMK1192
  READ(IR,16)ICARD                                                    SIMK1193
  MHEAD=ICARD(1)                                                      SIMK1194
  DECODE(10,30,ICARD(2))NROW,NCOL                                    SIMK1195
30 FORMAT(2I5)                                                        SIMK1196
  IF(ICARD(1).NE.IEND)GO TO 52                                       SIMK1197

```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

	JR=IR	SIMK1198
	IR=4	SIMK1199
C		SIMK1200
C	LOCATE SIMULATOR DECK DATA	SIMK1201
C		SIMK1202
	32 CONTINUE	SIMK1203
	READ(IR,16)ICARD	SIMK1204
	IF(EOF(IR))36,44	SIMK1205
	36 CONTINUE	SIMK1206
C		SIMK1207
C	PRINT ERROR MESSAGE	SIMK1208
C		SIMK1209
	WRITE(IW,40)IP	SIMK1210
	40 FORMAT(1H1,/,/,1X,3AHVEHICLE DATA CANNOT BE FOUND ON FILE= ,I2)	SIMK1211
	STOP 111	SIMK1212
C		SIMK1213
C	READ MATRIX NAME CARD	SIMK1214
C		SIMK1215
	44 CONTINUE	SIMK1216
	IF(IHEAD(1).NE.ICARD(1))GO TO 32	SIMK1217
	READ(IR,16)ICARD	SIMK1218
	MHEAD=ICARD(1)	SIMK1219
	DECODE(10,30,ICARD(2))NROW,NCOL	SIMK1220
	GO TO 52	SIMK1221
	48 CONTINUE	SIMK1222
	READ(IR,16)ICARD	SIMK1223
	MHEAD=ICARD(1)	SIMK1224
	DECODE(10,30,ICARD(2))NROW,NCOL	SIMK1225
	CONTINUE	SIMK1226
	52 IF(IPRINT.EQ.6)CALL DEBUG(3,4HSIMK,4H1 ,),0,IW)	SIMK1227
	DO 54 I=1,112	SIMK1228
	IF(MHEAD.EQ.JHEAD(I))GO TO 58	SIMK1229
	54 CONTINUE	SIMK1230
C		SIMK1231
C	PRINT ERROR MESSAGE	SIMK1232
C		SIMK1233
	55 CONTINUE	SIMK1234
	WRITE(IW,56)	SIMK1235
	56 FORMAT(1H1,/,/,1X,1QHERROR IN INPUT DATA)	SIMK1236
	STOP 111	SIMK1237
C		SIMK1238
C	READ AND PRINT LSA SIMULATOR DECK DATA	SIMK1239
C		SIMK1240
	58 CONTINUE	SIMK1241
	IF(I.GF.112)GO TO 98	SIMK1242
C		SIMK1243
C	COMPUTE II FROM I SO THAT STEADY GUST COEFF MATRICES ARE	SIMK1244
C	STORED IN THE SAME LOCATIONS AS THE GUST COEFF MATRICES	SIMK1245
C		SIMK1246
	II=I	SIMK1247
	IF((I.GT.16).AND.(I.LE.26))II=I-6	SIMK1248
	IF((I.GT.26).AND.(I.LE.44))GO TO 55	SIMK1249
	IF((I.GT.44).AND.(I.LE.60))II=I-12	SIMK1250
	IF((I.GT.60).AND.(I.LE.82))II=I-18	SIMK1251
	IF((I.GT.82).AND.(I.LE.104))II=I-24	SIMK1252
	IF((I.GT.104).AND.(I.LE.112))II=I-30	SIMK1253
C		SIMK1254
C	COMPUTE ARRAY START INDEX FOR SIMULATOR MATRIX DATA	SIMK1255
C		SIMK1256
	N=1	SIMK1257
	IM1=II-1	SIMK1258
	IF(II.NE.0.33)IM1=20	SIMK1259
	IF(IM1.EQ.0)GO TO 70	SIMK1260
	JM1=IM1	SIMK1261
	IF(IM1.GT.20)JM1=20	SIMK1262
	DO 60 J=1,JM1	SIMK1263

Figure 18. Subroutine SIMK1 Program Listing (Continued)

60	N=N*IDRM(J)*IDCM(J)	SIMK1264
	IF(IM1.LE.20)GO TO 70	SIMK1265
	DO 65 J=33,IM1	SIMK1266
65	N=N*IDRM(J)*IDCM(J)	SIMK1267
70	CONTINUE	SIMK1268
C		SIMK1269
C	READ AND PRINT SIMULATOR MATRIX DATA	SIMK1270
C		SIMK1271
	NROWM=IDRM(II) % NCOLM=IDCM(II)	SIMK1272
	CALL INPT1(SC(N),NROWM,NCOLM,NROW,NCOL,IR)	SIMK1273
	CALL MPRS1(SC(N),NROWM,NCOLM,NROW,NCOL,MHEAD)	SIMK1274
C		SIMK1275
C	COMPUTE STATE DIMENSION OF SIMULATOR MATRIX DATA READ	SIMK1276
C		SIMK1277
	IF((II.GT.00).AND.(II.LE.16))NXVP=NROW	SIMK1278
	IF((II.GT.16).AND.(II.LE.32))NXR=NROW	SIMK1279
	IF((II.GT.32).AND.(II.LE.48))NXUE=NROW	SIMK1280
	IF((II.GT.48).AND.(II.LE.64))NRT=NROW	SIMK1281
	IF((II.GT.64).AND.(II.LE.80))NRL=NROW	SIMK1282
C		SIMK1283
C	COMPUTE INPUT DIMENSION OF SIMULATOR MATRIX DATA READ	SIMK1284
C		SIMK1285
	DO 80 J=1,5	SIMK1286
	JJ=(J-1)*22+1	SIMK1287
	IF(JJ.LT.96)GO TO 80	SIMK1288
	IF(JJ.GT.110)GO TO 80	SIMK1289
	IF(JJ.EQ.96)NUC1=NCOL	SIMK1290
	IF(JJ.EQ.97)NUC2=NCOL	SIMK1291
	IF(JJ.EQ.98)NUC3=NCOL	SIMK1292
	IF(JJ.EQ.99)NUG0=NCOL	SIMK1293
	IF(JJ.EQ.100)NUG1=NCOL	SIMK1294
	IF(JJ.EQ.101)NVG0=NCOL	SIMK1295
	IF(JJ.EQ.102)NVG1=NCOL	SIMK1296
	IF(JJ.EQ.103)NWG0=NCOL	SIMK1297
	IF(JJ.EQ.104)NWG1=NCOL	SIMK1298
	IF(JJ.EQ.105)NUGS0=NCOL	SIMK1299
	IF(JJ.EQ.106)NUGS1=NCOL	SIMK1300
	IF(JJ.EQ.107)NVGS0=NCOL	SIMK1301
	IF(JJ.EQ.108)NVGS1=NCOL	SIMK1302
	IF(JJ.EQ.109)NWGS0=NCOL	SIMK1303
	IF(JJ.EQ.110)NWGS1=NCOL	SIMK1304
80	CONTINUE	SIMK1305
	GO TO 48	SIMK1306
C		SIMK1307
C	PRINT THE LAST CARD READ FROM SIMULATOR MATRIX DATA	SIMK1308
C		SIMK1309
98	CONTINUE	SIMK1310
	IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))WRITE(IW,100)MHEAD	SIMK1311
100	FORMAT(//,10X,A10,/))	SIMK1312
	IF(IPRINT.EQ.6)CALL DFBUG(4.4HSIMK,4H) *1.0,IW)	SIMK1313
C		SIMK1314
C	FORM THE UNITY MATRIX	SIMK1315
C		SIMK1316
	IF(NXUE.EQ.0)GO TO 134	SIMK1317
	DO 130 I=1,NXUF	SIMK1318
	DO 130 J=1,NXUF	SIMK1319
	UNITY(I,J)=0.0	SIMK1320
130	UNITY(I,I)=1.0	SIMK1321
134	CONTINUE	SIMK1322
C		SIMK1323
C	CHECK FOR DIMENSTION ERROR	SIMK1324
C		SIMK1325
	IF((NXVP.LE.NXVPM).AND.(NXR.LE.NXPM).AND.(NXUE.LE.NXUEM)	SIMK1326
	1.AND.(NUC1.LE.NUCM).AND.(NUC2.LE.NUCM).AND.(NUC3.LE.NUCM)	SIMK1327
	2.AND.(NUG0.LE.NUGM).AND.(NUG1.LE.NUGM)	SIMK1328
	3.AND.(NVG0.LE.NVGM).AND.(NVG1.LE.NVGM)	SIMK1329

Figure 18. Subroutine SIMK1 Program Listing (Continued)

	4.AND.(NWG0.LE.NUGM).AND.(NWG1.LE.NUGM)	SIMK1330
	5.AND.(NUGS0.LE.NUGM).AND.(NUGS1.LE.NUGM)	SIMK1331
	6.AND.(NVGS0.LE.NUGM).AND.(NVGS1.LE.NUGM)	SIMK1332
	7.AND.(NWGS0.LE.NUGM).AND.(NWGS1.LE.NUGM)	SIMK1333
	8.AND.(NRT.LE.NRTM).AND.(NRL.LE.NRLM).AND.(NB.LE.NBM).AND	SIMK1334
	9.(NL.LE.NLM)GO TO 13A	SIMK1335
	WRITE(IW,136)	SIMK1336
	136 FORMAT(1H1,/,1X,47HDIMENSION OF LSA DATA EXCEEDS THAT USED IN	SIMK1337
	1.16HSURROUTINE SIMK1)	SIMK1338
	STOP 111	SIMK1339
	138 CONTINUE	SIMK1340
C		SIMK1341
C	COMPUTE SYSTEM DIMENSIONS	SIMK1342
C		SIMK1343
	NX=NXVP+NXR+NXUE*2	SIMK1344
	NU=NUC1+NUC2+NUC3+NUG0+NUG1+NVG0+NVG1+NWG0+NWG1	SIMK1345
	1+NUGS0+NUGS1+NVGS0+NVGS1+NWGS0+NWGS1	SIMK1346
	NQ=NRT+NRL	SIMK1347
	IF(IR.NE.4)GO TO 12	SIMK1348
	IR=JR	SIMK1349
	IF(IPRINT.EQ.6)CALL DFBUG(5,4HSIMK,4H: .1.0.1W1	SIMK1350
	RETURN	SIMK1351
C		SIMK1352
C	PRINT ERROR MESSAGE	SIMK1353
C		SIMK1354
	140 CONTINUE	SIMK1355
	WRITE(IW,145)	SIMK1356
	145 FORMAT(1H1,/,1X,35HDIMENSION ERROR IN SURROUTINE SIMK1)	SIMK1357
	STOP 111	SIMK1358
	150 CONTINUE	SIMK1359
C		SIMK1360
C	DIFFERENTIAL EQUATIONS FOR RIGID BODY VELOCITIES	SIMK1361
C		SIMK1362
	IF(NXVP.LE.0)GO TO 264	SIMK1363
	DO 260 I=1,NXVP	SIMK1364
	XDOTL(I)=0.0	SIMK1365
C		SIMK1366
C	FROM RIGID BODY VELOCITIES	SIMK1367
C		SIMK1368
	DO 152 K=1,NXVP	SIMK1369
	152 XDOTL(I)=XDOTL(I)+VPVP0(I,K)*X(K)+VPVP1(I,K)*XDOT(K)	SIMK1370
	IF(NXR.LE.0)GO TO 160	SIMK1371
C		SIMK1372
C	FROM RIGID BODY ATTITUDES	SIMK1373
C		SIMK1374
	DO 156 K=1,NXR	SIMK1375
	KK=NXVP+K	SIMK1376
	156 XDOTL(I)=XDOTL(I)+VPR0(I,K)*X(KK)+VPR1(I,K)*XDOT(KK)	SIMK1377
	160 CONTINUE	SIMK1378
	IF(NXUF.LE.0)GO TO 16A	SIMK1379
C		SIMK1380
C	FROM HENDING MODES	SIMK1381
C		SIMK1382
	DO 164 K=1,NXUE	SIMK1383
	KK=NXVP+NXR+K	SIMK1384
	KKK=NXVP+NXR+NXUE+K	SIMK1385
	164 XDOTL(I)=XDOTL(I)+VPUE0(I,K)*X(KK)+VPUE1(I,K)*X(KKK)	SIMK1386
	1+VPUE2(I,K)*XDOT(KKK)	SIMK1387
	168 CONTINUE	SIMK1388
C		SIMK1389
C	FROM CONTROL SURFACE INPUTS	SIMK1390
C		SIMK1391
	IF(NUC1.LE.0)GO TO 184	SIMK1392
	DO 172 K=1,NUC1	SIMK1393
	172 XDOTL(I)=XDOTL(I)+VPDFLS0(I,K)*U(K)	SIMK1394
	IF(NUC2.LE.0)GO TO 184	SIMK1395

Figure 18. Subroutine SIMK1 Program Listing (Continued)

	DO 176 K=1,NUC2	SIMK1396
	KK=NUC1*K	SIMK1397
176	XDOTL(I)=XDOTL(I)+VPDFLS1(I,K)*U(KK)	SIMK1398
	IF(NUC3.LE.0)GO TO 184	SIMK1399
	DO 180 K=1,NUC3	SIMK1400
	KK=NUC1+NUC2*K	SIMK1401
180	XDOTL(I)=XDOTL(I)+VPDFLS2(I,K)*U(KK)	SIMK1402
184	CONTINUE	SIMK1403
	MU=NUC1+NUC2+NUC3	SIMK1404
	II=I	SIMK1405
C		SIMK1406
C	FROM U-GUST INPUTS	SIMK1407
C		SIMK1408
	IF(NUG0.LE.0)GO TO 196	SIMK1409
	DO 188 K=1,NUG0	SIMK1410
	KK=MU*K	SIMK1411
188	XDOTL(II)=XDOTL(II)+VPUG0(I,K)*U(KK)	SIMK1412
	IF(NUG1.LE.0)GO TO 196	SIMK1413
	DO 192 K=1,NUG1	SIMK1414
	KK=MU+NUG0*K	SIMK1415
192	XDOTL(II)=XDOTL(II)+VPUG1(I,K)*U(KK)	SIMK1416
196	CONTINUE	SIMK1417
	MU=MU+NUG0+NUG1	SIMK1418
C		SIMK1419
C	FROM V-GUST INPUTS	SIMK1420
C		SIMK1421
	IF(NVG0.LE.0)GO TO 208	SIMK1422
	DO 200 K=1,NVG0	SIMK1423
	KK=MU*K	SIMK1424
200	XDOTL(II)=XDOTL(II)+VPVG0(I,K)*U(KK)	SIMK1425
	IF(NVG1.LE.0)GO TO 208	SIMK1426
	DO 204 K=1,NVG1	SIMK1427
	KK=MU+NVG0*K	SIMK1428
204	XDOTL(II)=XDOTL(II)+VPVG1(I,K)*U(KK)	SIMK1429
208	CONTINUE	SIMK1430
	MU=MU+NVG0+NVG1	SIMK1431
C		SIMK1432
C	FROM W-GUST INPUTS	SIMK1433
C		SIMK1434
	IF(NWG0.LE.0)GO TO 220	SIMK1435
	DO 212 K=1,NWG0	SIMK1436
	KK=MU*K	SIMK1437
212	XDOTL(II)=XDOTL(II)+VPWG0(I,K)*U(KK)	SIMK1438
	IF(NWG1.LE.0)GO TO 220	SIMK1439
	DO 216 K=1,NWG1	SIMK1440
	KK=MU+NWG0*K	SIMK1441
216	XDOTL(II)=XDOTL(II)+VPWG1(I,K)*U(KK)	SIMK1442
220	CONTINUE	SIMK1443
	MU=MU+NWG0+NWG1	SIMK1444
C		SIMK1445
C	FROM STEADY U-GUST INPUTS	SIMK1446
C		SIMK1447
	IF(NUGS0.LE.0)GO TO 232	SIMK1448
	DO 224 K=1,NUGS0	SIMK1449
	KK=MU*K	SIMK1450
224	XDOTL(II)=XDOTL(II)+VPUG0(I,K)*U(KK)	SIMK1451
	IF(NUGS1.LE.0)GO TO 232	SIMK1452
	DO 228 K=1,NUGS1	SIMK1453
	KK=MU+NUGS0*K	SIMK1454
228	XDOTL(II)=XDOTL(II)+VPUG1(I,K)*U(KK)	SIMK1455
232	CONTINUE	SIMK1456
	MU=MU+NUGS0+NUGS1	SIMK1457
C		SIMK1458
C	FROM STEADY V-GUST INPUTS	SIMK1459
C		SIMK1460
	IF(NVGS0.LE.0)GO TO 244	SIMK1461

Figure 18. Subroutine SIMK1 Program Listing (Continued)

	DO 236 K=1,NVGS0	SIMK1462
	KK=MU*K	SIMK1463
236	XDOTL(II)=XDOTL(II)+VPVGO(I,K)*U(KK)	SIMK1464
	IF(NVGS1.LE.0)GO TO 244	SIMK1465
	DO 240 K=1,NVGS1	SIMK1466
	KK=MU*NVGS0*K	SIMK1467
240	XDOTL(II)=XDOTL(II)+VPVGI(I,K)*U(KK)	SIMK1468
244	CONTINUE	SIMK1469
	MU=MU*NVGS0+NVGS1	SIMK1470
C		SIMK1471
C	FROM STEADY W-GUST INPUTS	SIMK1472
C		SIMK1473
	IF(NWGS0.LE.0)GO TO 256	SIMK1474
	DO 248 K=1,NWGS0	SIMK1475
	KK=MU*K	SIMK1476
248	XDOTL(II)=XDOTL(II)+VPWGO(I,K)*U(KK)	SIMK1477
	IF(NWGS1.LE.0)GO TO 256	SIMK1478
	DO 252 K=1,NWGS1	SIMK1479
	KK=MU*NWGS0*K	SIMK1480
252	XDOTL(II)=XDOTL(II)+VPWGI(I,K)*U(KK)	SIMK1481
256	CONTINUE	SIMK1482
	MU=MU*NWGS0+NWGS1	SIMK1483
260	CONTINUE	SIMK1484
264	CONTINUE	SIMK1485
C		SIMK1486
C	DIFFERENTIAL EQUATIONS FOR RIGID BODY ATTITUDES	SIMK1487
C		SIMK1488
	IF(NXVP.LE.0)GO TO 280	SIMK1489
	DO 272 I=1,NXVP	SIMK1490
	II=NXVP+I	SIMK1491
	XDOTL(II)=0.0	SIMK1492
	IF(NXVP.LE.0)GO TO 272	SIMK1493
C		SIMK1494
C	FROM RIGID BODY VELOCITIES	SIMK1495
C		SIMK1496
	DO 268 K=1,NXVP	SIMK1497
268	XDOTL(II)=XDOTL(II)+RVPO(I,K)*X(K)+RVP1(I,K)*XDOT(K)	SIMK1498
272	CONTINUE	SIMK1499
C		SIMK1500
C	FROM RIGID BODY ATTITUDES	SIMK1501
C		SIMK1502
	DO 276 K=1,NXVP	SIMK1503
	KK=NXVP+K	SIMK1504
276	XDOTL(II)=XDOTL(II)+RPO(I,K)*X(KK)+RRI(I,K)*XDOT(KK)	SIMK1505
280	CONTINUE	SIMK1506
C		SIMK1507
C	DIFFERENTIAL EQUATIONS FOR BENDING MODE DISPLACEMENTS AND RATES	SIMK1508
C		SIMK1509
	IF(NXUF.LE.0)GO TO 396	SIMK1510
	DO 284 I=1,NXUF	SIMK1511
	II=NXVP+NXR+I	SIMK1512
	XDOTL(II)=0.0	SIMK1513
	DO 284 K=1,NXUF	SIMK1514
	KK=NXVP+NXR+NXUF+K	SIMK1515
284	XDOTL(II)=XDOTL(II)+UNITY(I,K)*X(KK)	SIMK1516
	DO 392 I=1,NXUF	SIMK1517
	II=NXVP+NXR+NXUF+I	SIMK1518
	XDOTL(II)=0.0	SIMK1519
	IF(NXVP.LE.0)GO TO 292	SIMK1520
C		SIMK1521
C	FROM RIGID BODY VELOCITIES	SIMK1522
C		SIMK1523
	DO 288 K=1,NXVP	SIMK1524
288	XDOTL(II)=XDOTL(II)+UEVPO(I,K)*X(K)+UEVPI(I,K)*XDOT(K)	SIMK1525
292	CONTINUE	SIMK1526
	IF(NXR.LE.0)GO TO 298	SIMK1527

Figure 18. Subroutine SIMK1 Program Listing (Continued)

C		SIMK1528
C	FROM RIGID BODY ATTITUDES	SIMK1529
C		SIMK1530
	DO 296 K=1,NXR	SIMK1531
	KK=NXVP*K	SIMK1532
296	XDOTL(II)=XDOTL(II)+UFRO(I,K)*X(KK)+UER1(I,K)*XDOT(KK)	SIMK1533
298	CONTINUE	SIMK1534
C		SIMK1535
C	FROM HENDING MODES	SIMK1536
C		SIMK1537
	DO 300 K=1,NXUF	SIMK1538
	KK=NXVP*NXR*K	SIMK1539
	KKK=NXVP*NXR*NXUE*K	SIMK1540
300	XDOTL(II)=XDOTL(II)+UFUFO(I,K)*X(KK)+UFUE1(I,K)*X(KKK)	SIMK1541
	+UFUE2(I,K)*XDOT(KKK)	SIMK1542
C		SIMK1543
C	FROM CONTROL SURFACE INPUTS	SIMK1544
C		SIMK1545
	IF(NUC1.LE.0)GO TO 316	SIMK1546
	DO 304 K=1,NUC1	SIMK1547
304	XDOTL(II)=XDOTL(II)+UEDELS0(I,K)*U(K)	SIMK1548
	IF(NUC2.LE.0)GO TO 316	SIMK1549
	DO 308 K=1,NUC2	SIMK1550
	KK=NUC1*K	SIMK1551
308	XDOTL(II)=XDOTL(II)+UFDELS1(I,K)*U(KK)	SIMK1552
	IF(NUC3.LE.0)GO TO 316	SIMK1553
	DO 312 K=1,NUC3	SIMK1554
	KK=NUC1+NUC2*K	SIMK1555
312	XDOTL(II)=XDOTL(II)+UFDELS2(I,K)*U(KK)	SIMK1556
316	CONTINUE	SIMK1557
	MU=NUC1+NUC2+NUC3	SIMK1558
C		SIMK1559
C	FROM U-GUST INPUTS	SIMK1560
C		SIMK1561
	IF(NUG0.LE.0)GO TO 328	SIMK1562
	DO 320 K=1,NUG0	SIMK1563
	KK=MU*K	SIMK1564
320	XDOTL(II)=XDOTL(II)+UFUG0(I,K)*U(KK)	SIMK1565
	IF(NUG1.LE.0)GO TO 328	SIMK1566
	DO 324 K=1,NUG1	SIMK1567
	KK=MU+NUG0*K	SIMK1568
324	XDOTL(II)=XDOTL(II)+UFUG1(I,K)*U(KK)	SIMK1569
328	CONTINUE	SIMK1570
	MU=MU+NUG0+NUG1	SIMK1571
C		SIMK1572
C	FROM V-GUST INPUTS	SIMK1573
C		SIMK1574
	IF(NVGO.LE.0)GO TO 340	SIMK1575
	DO 332 K=1,NVGO	SIMK1576
	KK=MU*K	SIMK1577
332	XDOTL(II)=XDOTL(II)+UFVGO(I,K)*U(KK)	SIMK1578
	IF(NVG1.LE.0)GO TO 340	SIMK1579
	DO 336 K=1,NVG1	SIMK1580
	KK=MU+NVGO*K	SIMK1581
336	XDOTL(II)=XDOTL(II)+UFVG1(I,K)*U(KK)	SIMK1582
340	CONTINUE	SIMK1583
	MU=MU+NVGO+NVG1	SIMK1584
C		SIMK1585
C	FROM W-GUST INPUTS	SIMK1586
C		SIMK1587
	IF(NWGO.LE.0)GO TO 352	SIMK1588
	DO 344 K=1,NWGO	SIMK1589
	KK=MU*K	SIMK1590
344	XDOTL(II)=XDOTL(II)+UEWGO(I,K)*U(KK)	SIMK1591
	IF(NWG1.LE.0)GO TO 352	SIMK1592
	DO 348 K=1,NWG1	SIMK1593

Figure 18. Subroutine SIMK1 Program Listing (Continued)

	KK=MU*NWGO*K	SIMK1594
	348 XDOTL (II)=XDOTL (II)+UEWGI (I,K)*U(KK)	SIMK1595
	352 CONTINUE	SIMK1596
	MU=MU*NWGO*NWGI	SIMK1597
C		SIMK1598
C	FROM STEADY U-GUST INPUTS	SIMK1599
C		SIMK1600
	IF (NUGSO.LE.0)GO TO 364	SIMK1601
	DO 356 K=1,NUGSO	SIMK1602
	KK=MU*K	SIMK1603
	356 XDOTL (II)=XDOTL (II)+UEUGO (I,K)*U(KK)	SIMK1604
	IF (NUGS1.LE.0)GO TO 364	SIMK1605
	DO 360 K=1,NUGS1	SIMK1606
	KK=MU*NUGSO*K	SIMK1607
	360 XDOTL (II)=XDOTL (II)+UEUGI (I,K)*U(KK)	SIMK1608
	364 CONTINUE	SIMK1609
	MU=MU*NUGSO*NUGS1	SIMK1610
C		SIMK1611
C	FROM STEADY V-GUST INPUTS	SIMK1612
C		SIMK1613
	IF (NVGSO.LE.0)GO TO 376	SIMK1614
	DO 368 K=1,NVGSO	SIMK1615
	KK=MU*K	SIMK1616
	368 XDOTL (II)=XDOTL (II)+UEVGO (I,K)*U(KK)	SIMK1617
	IF (NVGS1.LE.0)GO TO 376	SIMK1618
	DO 372 K=1,NVGS1	SIMK1619
	KK=MU*NVGSO*K	SIMK1620
	372 XDOTL (II)=XDOTL (II)+UEVGI (I,K)*U(KK)	SIMK1621
	376 CONTINUE	SIMK1622
	MU=MU*NVGSO*NVGS1	SIMK1623
C		SIMK1624
C	FROM STEADY W-GUST INPUTS	SIMK1625
C		SIMK1626
	IF (NWGSO.LE.0)GO TO 388	SIMK1627
	DO 380 K=1,NWGSO	SIMK1628
	KK=MU*K	SIMK1629
	380 XDOTL (II)=XDOTL (II)+UFWGO (I,K)*U(KK)	SIMK1630
	IF (NWGS1.LE.0)GO TO 388	SIMK1631
	DO 384 K=1,NWGS1	SIMK1632
	KK=MU*NWGSO*K	SIMK1633
	384 XDOTL (II)=XDOTL (II)+UFWGI (I,K)*U(KK)	SIMK1634
	388 CONTINUE	SIMK1635
	MU=MU*NWGSO*NWGS1	SIMK1636
	392 CONTINUE	SIMK1637
	396 CONTINUE	SIMK1638
C		SIMK1639
C	SENSOR EQUATIONS	SIMK1640
C		SIMK1641
	IF (NRT.LE.0)GO TO 516	SIMK1642
	DO 512 I=1,NRT	SIMK1643
	RL (I)=0.0	SIMK1644
	IF (NXVP.LE.0)GO TO 404	SIMK1645
C		SIMK1646
C	FROM RIGID BODY VELOCITIES	SIMK1647
C		SIMK1648
	DO 400 K=1,NXVP	SIMK1649
	400 RL (I)=RL (I)+TVPO (I,K)*X (K)+TVPI (I,K)*XDOT (K)	SIMK1650
	404 CONTINUE	SIMK1651
	IF (NXR.LE.0)GO TO 412	SIMK1652
C		SIMK1653
C	FROM RIGID BODY ATTITUDES	SIMK1654
C		SIMK1655
	DO 408 K=1,NXR	SIMK1656
	KK=NXVP*K	SIMK1657
	408 RL (I)=RL (I)+TR0 (I,K)*X (KK)+TR1 (I,K)*XDOT (KK)	SIMK1658
	412 CONTINUE	SIMK1659

Figure 18. Subroutine SIMK1 Program Listing (Continued)

	IF(NXUF.LE.0)GO TO 420	SIMK1660
C		SIMK1661
C	FROM HENDING MODES	SIMK1662
C		SIMK1663
	DO 416 K=1,NXUF	SIMK1664
	KK=NXVP+NXR*K	SIMK1665
	KKK=NXVP+NXR+NXUF*K	SIMK1666
416	RL(I)=RL(I)+TUF0(I,K)*X(KK)+TUF1(I,K)*X(KKK)+TUF2(I,K)*XDOT(KKK)	SIMK1667
420	CONTINUE	SIMK1668
C		SIMK1669
C	FROM CONTROL SURFACE INPUTS	SIMK1670
C		SIMK1671
	IF(NUC1.LE.0)GO TO 436	SIMK1672
	DO 424 K=1,NUC1	SIMK1673
424	RL(I)=PL(I)+TDFLS0(I,K)*U(K)	SIMK1674
	IF(NUC2.LE.0)GO TO 436	SIMK1675
	DO 424 K=1,NUC2	SIMK1676
	KK=NUC1*K	SIMK1677
428	RL(I)=RL(I)+TDFLS1(I,K)*U(K)	SIMK1678
	IF(NUC3.LE.0)GO TO 436	SIMK1679
	DO 432 K=1,NUC3	SIMK1680
	KK=NUC1+NUC2*K	SIMK1681
432	RL(I)=RL(I)+TDFLS2(I,K)*U(K)	SIMK1682
436	CONTINUE	SIMK1683
	MU=NUC1+NUC2+NUC3	SIMK1684
C		SIMK1685
C	FROM U-GUST INPUTS	SIMK1686
C		SIMK1687
	IF(NUG0.LE.0)GO TO 448	SIMK1688
	DO 440 K=1,NUG0	SIMK1689
	KK=MU*K	SIMK1690
440	RL(I)=PL(I)+TUG0(I,K)*U(KK)	SIMK1691
	IF(NUG1.LE.0)GO TO 448	SIMK1692
	DO 444 K=1,NUG1	SIMK1693
	KK=MU+NUG0*K	SIMK1694
444	RL(I)=RL(I)+TUG1(I,K)*U(KK)	SIMK1695
448	CONTINUE	SIMK1696
	MU=MU+NUG0+NUG1	SIMK1697
C		SIMK1698
C	FROM V-GUST INPUTS	SIMK1699
C		SIMK1700
	IF(NVG0.LE.0)GO TO 460	SIMK1701
	DO 452 K=1,NVG0	SIMK1702
	KK=MU*K	SIMK1703
452	RL(I)=RL(I)+TVG0(I,K)*U(KK)	SIMK1704
	IF(NVG1.LE.0)GO TO 460	SIMK1705
	DO 456 K=1,NVG1	SIMK1706
	KK=MU+NVG0*K	SIMK1707
456	RL(I)=RL(I)+TVG1(I,K)*U(KK)	SIMK1708
460	CONTINUE	SIMK1709
	MU=MU+NVG0+NVG1	SIMK1710
C		SIMK1711
C	FROM W-GUST INPUTS	SIMK1712
C		SIMK1713
	IF(NWG0.LE.0)GO TO 472	SIMK1714
	DO 464 K=1,NWG0	SIMK1715
	KK=MU*K	SIMK1716
464	RL(I)=RL(I)+TWG0(I,K)*U(KK)	SIMK1717
	IF(NWG1.LE.0)GO TO 472	SIMK1718
	DO 468 K=1,NWG1	SIMK1719
	KK=MU+NWG0*K	SIMK1720
468	RL(I)=RL(I)+TWG1(I,K)*U(KK)	SIMK1721
472	CONTINUE	SIMK1722
	MU=MU+NWG0+NWG1	SIMK1723
C		SIMK1724
C	FROM STEADY U-GUST INPUTS	SIMK1725

Figure 18. Subroutine SIMK1 Program Listing (Continued)

	IF (NUGS0.LE.0)GO TO 484	SIMK1724
	DO 476 K=1,NUGS0	SIMK1727
	KK=MU*K	SIMK1729
476	RL(I)=RL(I)+TUGO(I,K)*U(KK)	SIMK1729
	IF (NUGS1.LE.0)GO TO 484	SIMK1730
	DO 480 K=1,NUGS1	SIMK1731
	KK=MU*NUGS0*K	SIMK1732
480	RL(I)=RL(I)+TUG1(I,K)*U(KK)	SIMK1733
484	CONTINUE	SIMK1734
	MU=MU+NUGS0+NUGS1	SIMK1735
C		SIMK1736
C	FROM STEADY V-GUST INPUTS	SIMK1737
C		SIMK1738
	IF (NVGS0.LE.0)GO TO 496	SIMK1739
	DO 488 K=1,NVGS0	SIMK1740
	KK=MU*K	SIMK1741
488	RL(I)=RL(I)+TVGO(I,K)*U(KK)	SIMK1742
	IF (NVGS1.LE.0)GO TO 496	SIMK1743
	DO 492 K=1,NVGS1	SIMK1744
	KK=MU*NVGS0*K	SIMK1745
492	RL(I)=RL(I)+TVG1(I,K)*U(KK)	SIMK1746
496	CONTINUE	SIMK1747
	MU=MU+NVGS0+NVGS1	SIMK1749
C		SIMK1750
C	FROM STEADY W-GUST INPUTS	SIMK1751
C		SIMK1752
	IF (NWGS0.LE.0)GO TO 508	SIMK1753
	DO 500 K=1,NWGS0	SIMK1754
	KK=MU*K	SIMK1755
500	RL(I)=RL(I)+TWGO(I,K)*U(KK)	SIMK1756
	IF (NWGS1.LE.0)GO TO 508	SIMK1757
	DO 504 K=1,NWGS1	SIMK1758
	KK=MU*NWGS0*K	SIMK1759
504	RL(I)=RL(I)+TWG1(I,K)*U(KK)	SIMK1760
508	CONTINUE	SIMK1761
	MU=MU+NWGS0+NWGS1	SIMK1762
512	CONTINUE	SIMK1763
516	CONTINUE	SIMK1764
C		SIMK1765
C	LOAD EQUATIONS	SIMK1766
C		SIMK1767
	IF (NRL.LE.0)GO TO 716	SIMK1768
	DO 712 I=1,NRL	SIMK1769
	J=NRT*I	SIMK1770
	RL(J)=0.0	SIMK1771
	IF (NXVP.LE.0)GO TO 604	SIMK1772
C		SIMK1773
C	FROM RIGID BODY VELOCITIES	SIMK1774
C		SIMK1775
	DO 600 K=1,NXVP	SIMK1776
600	RL(J)=RL(J)+LVP0(I,K)*X(K)+LVP1(I,K)*XDOT(K)	SIMK1777
604	CONTINUE	SIMK1778
	IF (NXR.LE.0)GO TO 612	SIMK1779
C		SIMK1780
C	FROM RIGID BODY ATTITUDES	SIMK1781
C		SIMK1782
	DO 608 K=1,NXR	SIMK1783
	KK=NXVP*K	SIMK1784
608	RL(J)=RL(J)+LR0(I,K)*X(KK)+LR1(I,K)*XDOT(KK)	SIMK1785
612	CONTINUE	SIMK1786
	IF (NXUE.LE.0)GO TO 620	SIMK1787
C		SIMK1788
C	FROM BENDING MODES	SIMK1789
C		SIMK1790
	DO 616 K=1,NXUF	SIMK1791

Figure 18. Subroutine SIMK1 Program Listing (Continued)

	KK=NXP+NXR*K	SIMK1792
	KKK=NXP+NXR+NUF**	SIMK1793
616	PL(J)=PL(J)+LDF0(I,K)*X(KK)+LDF1(I,K)*X(KKK)+LDF2(I,K)*XDOT(KKK)	SIMK1794
620	CONTINUE	SIMK1795
C		SIMK1796
C	FROM CONTROL SURFACE INPUTS	SIMK1797
C		SIMK1798
	IF(NUC).LE.0)GO TO 636	SIMK1799
	DO 624 K=1,NUC1	SIMK1800
624	PL(J)=PL(J)+LDFL50(I,K)*U(K)	SIMK1801
	IF(NUC2.LE.0)GO TO 636	SIMK1802
	DO 624 K=1,NUC2	SIMK1803
	KK=NUC1*K	SIMK1804
628	PL(J)=PL(J)+LDFL51(I,K)*U(KK)	SIMK1805
	IF(NUC3.LE.0)GO TO 636	SIMK1806
	DO 632 K=1,NUC3	SIMK1807
	KK=NUC1*NUC2*K	SIMK1808
632	PL(J)=PL(J)+LDFL52(I,K)*U(KK)	SIMK1809
636	CONTINUE	SIMK1810
	MU=NUC1+NUC2+NUC3	SIMK1811
C		SIMK1812
C	FROM U-GUST INPUTS	SIMK1813
C		SIMK1814
	IF(NUG0.LE.0)GO TO 648	SIMK1815
	DO 640 K=1,NUG0	SIMK1816
	KK=MU*K	SIMK1817
640	PL(J)=PL(J)+LUG0(I,K)*U(KK)	SIMK1818
	IF(NUG1.LE.0)GO TO 648	SIMK1819
	DO 644 K=1,NUG1	SIMK1820
	KK=MU+NUG0*K	SIMK1821
644	PL(J)=PL(J)+LUG1(I,K)*U(KK)	SIMK1822
648	CONTINUE	SIMK1823
	MU=MU+NUG0+NUG1	SIMK1824
C		SIMK1825
C	FROM V-GUST INPUTS	SIMK1826
C		SIMK1827
	IF(NVG0.LE.0)GO TO 660	SIMK1828
	DO 652 K=1,NVG0	SIMK1829
	KK=MU*K	SIMK1830
652	PL(J)=PL(J)+LVG0(I,K)*U(KK)	SIMK1831
	IF(NVG1.LE.0)GO TO 660	SIMK1832
	DO 656 K=1,NVG1	SIMK1833
	KK=MU+NVG0*K	SIMK1834
656	PL(J)=PL(J)+LVG1(I,K)*U(KK)	SIMK1835
660	CONTINUE	SIMK1836
	MU=MU+NVG0+NVG1	SIMK1837
C		SIMK1838
C	FROM W-GUST INPUTS	SIMK1839
C		SIMK1840
	IF(NWG0.LE.0)GO TO 672	SIMK1841
	DO 664 K=1,NWG0	SIMK1842
	KK=MU*K	SIMK1843
664	PL(J)=PL(J)+LWG0(I,K)*U(KK)	SIMK1844
	IF(NWG1.LE.0)GO TO 672	SIMK1845
	DO 668 K=1,NWG1	SIMK1846
	KK=MU+NWG0*K	SIMK1847
668	PL(J)=PL(J)+LWG1(I,K)*U(KK)	SIMK1848
672	CONTINUE	SIMK1849
	MU=MU+NWG0+NWG1	SIMK1850
C		SIMK1851
C	FROM STEADY U-GUST INPUTS	SIMK1852
C		SIMK1853
	IF(NUGS0.LE.0)GO TO 684	SIMK1854
	DO 676 K=1,NUGS0	SIMK1855
	KK=MU*K	SIMK1856
676	PL(J)=PL(J)+LUG0(I,K)*U(KK)	SIMK1857

Figure 18. Subroutine SIMK1 Program Listing (Continued)

	IF (NVGS1.LE.0) GO TO 684	SIMK1854
	DO 680 K=1,NVGS1	SIMK1855
	KK=MU+NVGS0*K	SIMK1860
680	RL(J)=RL(J)+LUG1(I,K)*U(KK)	SIMK1861
684	CONTINUE	SIMK1862
	MU=MU+NVGS0*NVGS1	SIMK1863
C		SIMK1864
C	FROM STEADY V-GUST INPUTS	SIMK1865
C		SIMK1866
	IF (NVGS0.LE.0) GO TO 696	SIMK1867
	DO 684 K=1,NVGS0	SIMK1868
	KK=MU*K	SIMK1869
684	RL(J)=RL(J)+LVG0(I,K)*U(KK)	SIMK1870
	IF (NVGS1.LE.0) GO TO 696	SIMK1871
	DO 692 K=1,NVGS1	SIMK1872
	KK=MU+NVGS0*K	SIMK1873
692	RL(J)=RL(J)+LVG1(I,K)*U(KK)	SIMK1874
696	CONTINUE	SIMK1875
	MU=MU+NVGS0*NVGS1	SIMK1876
C		SIMK1877
C	FROM STEADY W-GUST INPUTS	SIMK1878
C		SIMK1879
	IF (NWGS0.LE.0) GO TO 708	SIMK1880
	DO 700 K=1,NWGS0	SIMK1881
	KK=MU*K	SIMK1882
700	RL(J)=RL(J)+LWG0(I,K)*U(KK)	SIMK1883
	IF (NWGS1.LE.0) GO TO 708	SIMK1884
	DO 704 K=1,NWGS1	SIMK1885
	KK=MU+NWGS0*K	SIMK1886
704	RL(J)=RL(J)+LWG1(I,K)*U(KK)	SIMK1887
708	CONTINUE	SIMK1888
	MU=MU+NWGS0*NWGS1	SIMK1889
712	CONTINUE	SIMK1890
716	CONTINUE	SIMK1891
	RETURN	SIMK1892
	END	SIMK1893

Figure 18. Subroutine SIMK1 Program Listing (Concluded)

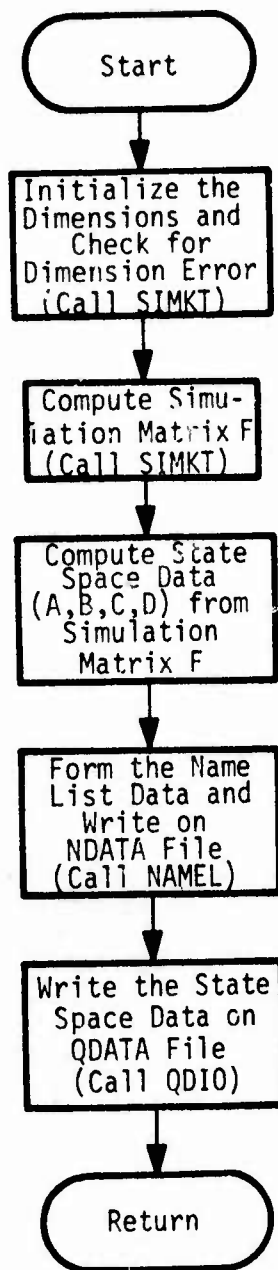


Figure 19. Subroutine STAMK2 Flow Chart

10	X(J,N)=0.	STAMK131
	DO 11 NN=1,NMAX	STAMK132
	MX=NN*(NN)	STAMK133
	DO 12 J=1,MX	STAMK134
12	RI(J,NN)=0.	STAMK135
	MX=NN*(NN)	STAMK136
	DO 13 J=1,MX	STAMK137
13	UI(J,NN)=0.	STAMK138
11	CONTINUE	STAMK139
	DO 14 I=1,NU	STAMK140
14	U(I)=0.	STAMK141
C		STAMK142
C	COMPUTE PARTIALS WRT STATE DERIVATIVES	STAMK143
C		STAMK144
	JJ=0	STAMK145
	DO 50 NN=1,NMAX	STAMK146
	MX=NN*(NN)	STAMK147
	DO 50 J=1,MX	STAMK148
	JJ=JJ+1	STAMK149
	XDOT(I,NN)=1.	STAMK150
	CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NUU,	STAMK151
	IAT,RT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MTFB,MST,	STAMK152
	ZMT,NUM,NRM,INIT,T)	STAMK153
	XDOT(J,NN)=0.	STAMK154
	DO 50 I=1,N	STAMK155
50	F(I,J)=V(I)	STAMK156
C		STAMK157
C	COMPUTE PARTIALS WRT INTERNAL OUTPUTS	STAMK158
C		STAMK159
	DO 10 NN=1,NMAX	STAMK160
	MX=NN*(NN)	STAMK161
	DO 10 J=1,MX	STAMK162
	JJ=JJ+1	STAMK163
	RI(J,NN)=1.	STAMK164
	CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NUU,	STAMK165
	IAT,RT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MTFB,MST,	STAMK166
	ZMT,NUM,NRM,INIT,T)	STAMK167
	RI(J,NN)=0.	STAMK168
	DO 10 I=1,N	STAMK169
100	F(I,J)=V(I)	STAMK170
C		STAMK171
C	COMPUTE PARTIALS WRT INTERNAL INPUTS	STAMK172
C		STAMK173
	DO 15 NN=1,NMAX	STAMK174
	MX=NN*(NN)	STAMK175
	DO 15 J=1,MX	STAMK176
	JJ=JJ+1	STAMK177
	UI(J,NN)=1.	STAMK178
	CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NUU,	STAMK179
	IAT,RT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MTFB,MST,	STAMK180
	ZMT,NUM,NRM,INIT,T)	STAMK181
	UI(J,NN)=0.	STAMK182
	DO 15 I=1,N	STAMK183
150	F(I,J)=V(I)	STAMK184
	IF(IPRINT.EQ.6)CALL DFRUG(4,4HSTAM,4H(2),2,0,1W)	STAMK185
C		STAMK186
C	COMPUTE PARTIALS WRT STATES	STAMK187
C		STAMK188
	DO 201 NN=1,NMAX	STAMK189
	MX=NN*(NN)	STAMK190
	DO 201 J=1,MX	STAMK191
	JJ=JJ+1	STAMK192
	X(J,NN)=1.	STAMK193
	CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NUU,	STAMK194
	IAT,RT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MTFB,MST,	STAMK195
	ZMT,NUM,NRM,INIT,T)	STAMK196

Figure 20. Subroutine STAMK2 Program Listing (Continued)

```

      X(J,NH)=0.
      DO 201 I=1,N
201  F(I,J)=V(I)
      IF(IP=INT.EQ.6)CALL DFHUG(5.4HSTAM,4HX2 ,2,0,1W)
C
C   COMPUTE PARTIALS WRT EXTERNAL INPUTS
C
      DO 251 J=1,NU
      JJ=JJ+1
      U(J)=1.
      CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NVR,NUU,
      IAT,BT,CT,DT,PRINT,HS,P,0,R,S,NX,NY,NR,NU,NMAX,MTFB,MST,
      2MT,NUM,NRM,INIT,T)
      U(J)=1.
      DO 251 I=1,N
251  F(I,J)=V(I)
8002 CONTINUE
C
C   COMPUTE THE SIMULATION MATRIX
C
      NV=NX+NY
      IF(IP=INT.EQ.6)CALL MPPS(F,MAXN,MAXM,V,M,T,4HSM )
      DO 51 I=1,NV
      DO 52 J=1,NV
52  F(I,J)=-F(I,J)
51  F(I,I)=F(I,I)+1.
      CALL TDINVR(ISOL,INDOL,NV,-M,F,MAXN,KDIM,DET)
      IR=NV+1
      IF=NV+NR
      JB=IB
      JE=M
      DO 53 I=IR,IF
      DO 53 J=JB,JE
      DO 53 K=1,NV
53  F(I,J)=F(I,J)+F(I,K)*F(K,J)
      DO 531 I=1,IE
      DO 53 J=1,JE
      IF(ABS(F(I,J)),LE.FPSF) F(I,J) = 0.0
530 CONTINUE
      IF(IP=INT.EQ.6)CALL MPRS(F,MAXN,MAXM,V,M,T,4HSMI)
C
C   FORM A,B,C,D MATRICES
C
      J1=NV+1
      J2=NV+NX
      J3=J1+NX
      J4=J2+NU
      I1=NV+1
      I2=NV+NR
      DO 60 1 I=1,NX
      DO 60 1 J=J1,J2
      JJ=J-1+1
6001 A(I,J)=F(I,J)
      DO 60 2 I=1,NX
      DO 60 2 J=J3,J4
      JJ=J-13+1
6002 B(I,J)=F(I,J)
      DO 60 3 I=I1,I2
      II=I-1+1
      DO 60 3 J=J1,J2
      JJ=J-1+1
6003 C(II,J)=F(I,J)
      DO 60 4 I=I1,I2
      II=I-1+1
      DO 60 4 J=J3,J4
      JJ=J-13+1
      STAMK197
      STAMK198
      STAMK199
      STAMK200
      STAMK201
      STAMK202
      STAMK203
      STAMK204
      STAMK205
      STAMK206
      STAMK207
      STAMK208
      STAMK209
      STAMK210
      STAMK211
      STAMK212
      STAMK213
      STAMK214
      STAMK215
      STAMK216
      STAMK217
      STAMK218
      STAMK219
      STAMK220
      STAMK221
      STAMK222
      STAMK223
      STAMK224
      STAMK225
      STAMK226
      STAMK227
      STAMK228
      STAMK229
      STAMK230
      STAMK231
      STAMK232
      STAMK233
      STAMK234
      STAMK235
      STAMK236
      STAMK237
      STAMK238
      STAMK239
      STAMK240
      STAMK241
      STAMK242
      STAMK243
      STAMK244
      STAMK245
      STAMK246
      STAMK247
      STAMK248
      STAMK249
      STAMK250
      STAMK251
      STAMK252
      STAMK253
      STAMK254
      STAMK255
      STAMK256
      STAMK257
      STAMK258
      STAMK259
      STAMK260
      STAMK261
      STAMK262

```

Figure 20. Subroutine STAMK2 Program Listing (Continued)

6004	D(I1, IJ)=F(I, J)	STAMK263
C		STAMK264
C	READ AND UPDATE NAME LIST DATA	STAMK265
C		STAMK266
	KR=NMAX	STAMK267
	CALL NAMEL (NNS, VNS, DESS, UNITS, NNO, VNO, DESO, UNITO, NNI, VNI,	STAMK268
	1 DESI, UNITI, S1(L1), S1(L2), S1(L3), S1(L4), S1(L5), S1(L6),	STAMK269
	2 S1(L7), S1(L8), S1(L9), NXM, NRM, NUM, NX, NR, NU, NFLAG, MB, KB, NB)	STAMK270
	IF (IPRINT.EQ.6) CALL DEBUG(6, 4HSTAM, 4HK2 .2, 0, IW)	STAMK271
C		STAMK272
C	WRITE QUADRUPE DATA ON FILE QDATA	STAMK273
C		STAMK274
	IQ=0	STAMK275
	MFLAG=?	STAMK276
	NXA=NX \$ NRA=NR \$ NUA=NU	STAMK277
	CALL QDIO (A, B, C, D, A, NX, NR, NU, NXM, NRM, NUM, NXA, NRA, NUA,	STAMK278
	1 NR1, NR2, NR3, NU1, NU2, NU3, T, IQ, IPRINT, I#, JO, HEAD, MARK,	STAMK279
	2 LOCATF, NULL, INSERT, MFLAG)	STAMK280
	IF (IPRINT.EQ.6) CALL DEBUG(7, 4HSTAM, 4HK2 .2, 0, IW)	STAMK281
	RETURN	STAMK282
	END	STAMK283

Figure 20. Subroutine STAMK2 Program Listing (Concluded)

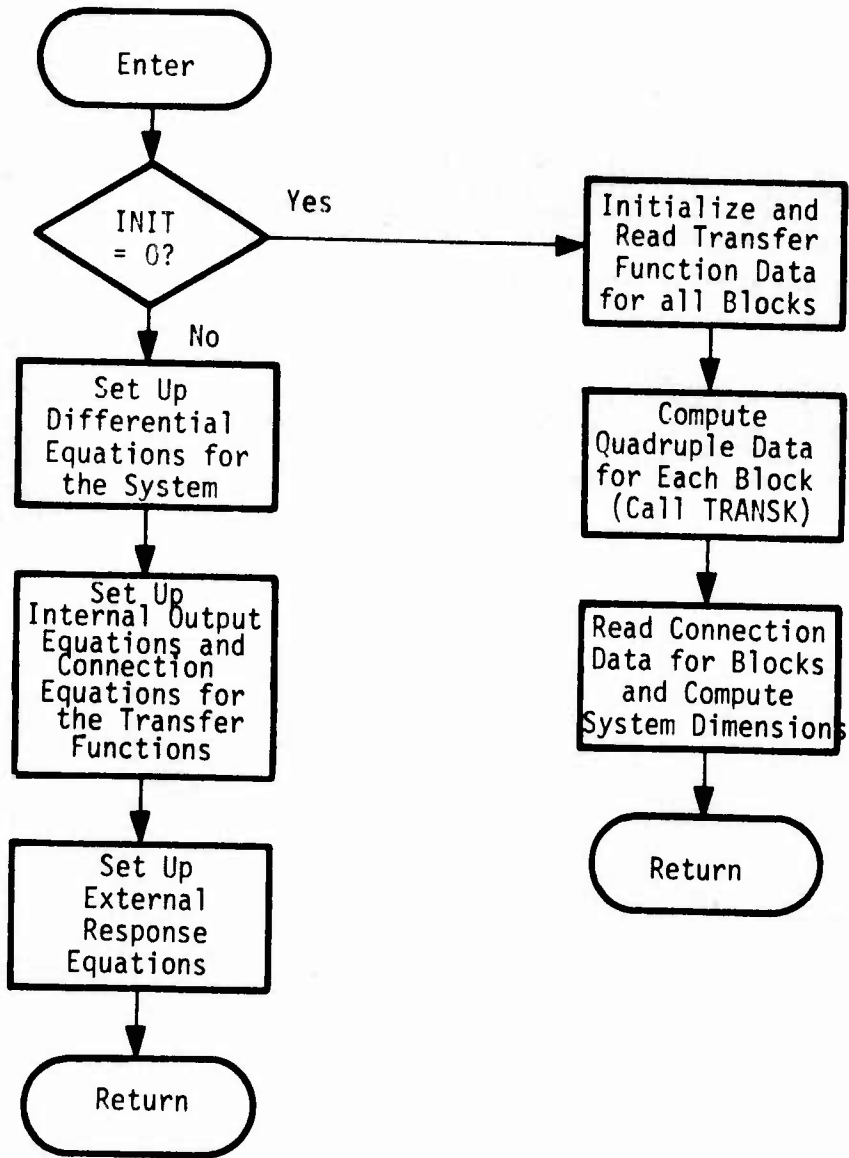


Figure 21. Subroutine SIMKT Flow Chart

1060	FORMAT(20A4)	SIMKT 65
	DECODE(4,1100,CARD(1))CC,DUMMY	SIMKT 66
1100	FORMAT(A1,A3)	SIMKT 67
	IF(CC.EQ.HC)GO TO 1040	SIMKT 68
	IF(CARD(1).EQ.HEND)GO TO 1400	SIMKT 69
	IF(CARD(1).NE.HBLOC)GO TO 1440	SIMKT 70
	BLK1=CARD(1)	SIMKT 71
	DECODE(4,1120,CARD(2))BLK2,NBKNO,DUMMY	SIMKT 72
1120	FORMAT(A1,I2,A1)	SIMKT 73
	BKD1=CARD(3)	SIMKT 74
	BKD2=CARD(4)	SIMKT 75
	IF(BLK2.NE.HK)GO TO 1440	SIMKT 76
	N=NBKNO	SIMKT 77
	NBK=NBK+1	SIMKT 78
	IF(NBK.GT.MTFB)GO TO 1190	SIMKT 79
	DO 1125 I=1,3	SIMKT 80
1125	IHEAD(I)=CARD(I)	SIMKT 81
	IF((BKD1.EQ.MDELA).AND.(BKD2.EQ.HY))GO TO 1300	SIMKT 82
	IF(IPRINT.EQ.6)CALL DEBUG(2,4HSIMK,4HT ,2,0,1W)	SIMKT 83
C		SIMKT 84
C	READ RATIONAL TRANSFER FUNCTION DATA	SIMKT 85
C		SIMKT 86
	CALL ZERO(IPRINT,2,MT)	SIMKT 87
	CALL INPT(IPRINT,2,MT)	SIMKT 88
	DO 1130 I=1,2	SIMKT 89
	DO 1130 J=1,MT	SIMKT 90
1130	MS(I,J,N)=PRINT(I,J)	SIMKT 91
	DO 1160 I=1,MT	SIMKT 92
	IF(MS(1,I,N).NE.0.0)NNX(N)=I-1	SIMKT 93
	IF(MS(2,I,N).NE.0.0)NNX(N)=I-1	SIMKT 94
1160	CONTINUE	SIMKT 95
	IF(MS(2,1,N).NE.0.0)GO TO 1200	SIMKT 96
C		SIMKT 97
C	PRINT ERROR MESSAGE	SIMKT 98
C		SIMKT 99
	WRITE(IW,1180)	SIMKT100
1180	FORMAT(1H1,/,1X,37HTRANSFER FUNCTION SPECIFICATION ERROR)	SIMKT101
	STOP 111	SIMKT102
1190	CONTINUE	SIMKT103
	WRITE(IW,1195)	SIMKT104
1195	FORMAT(1H1,/,29HTOO MANY BLOCKS FOR COMBINING)	SIMKT105
	STOP 111	SIMKT106
C		SIMKT107
C	PRINT THE TRANSFER FUNCTION	SIMKT108
C		SIMKT109
1200	CONTINUE	SIMKT110
	IF((IPRINT.NE.3).AND.(IPRINT.LT.5))GO TO 1040	SIMKT111
	NN1=NNX(N)+1	SIMKT112
	DO 1240 I=1,2	SIMKT113
	DO 1240 J=1,NN1	SIMKT114
1240	PRINT(I,J)=MS(I,J,N)	SIMKT115
	CALL TPR(IPRINT,NN1,MT,IHEAD,T,IW)	SIMKT116
	GO TO 1040	SIMKT117
1300	CONTINUE	SIMKT118
	IF(IPRINT.EQ.6)CALL DEBUG(3,4HSIMK,4HT ,2,0,1W)	SIMKT119
C		SIMKT120
C	READ TIME DELAY SPECIFICATION	SIMKT121
C		SIMKT122
	READ(I,1320)TO,XX,XR,UU,OMEGM,DELPHM,ND,NN	SIMKT123
1300	FORMAT(6E12,6,2I2)	SIMKT124
	IF(TD.GT.0.0)GO TO 1380	SIMKT125
	IF(UU.EQ.0.0)GO TO 1340	SIMKT126
	T=XX-XR/UU	SIMKT127
	IF(TD.LE.0.0)GO TO 1340	SIMKT128
	GO TO 1300	SIMKT129
1340	CONTINUE	SIMKT130

Figure 22. Subroutine SIMKT Program Listing (Continued)

C		SIMKT131
C	PRINT ERROR MESSEGE	SIMKT132
C		SIMKT133
	WRITE(IW,1360)	SIMKT134
1360	FORMAT(IH1,///,1X,30)TIME DELAY SPECIFICATION ERROR)	SIMKT135
	STOP 111	SIMKT136
1380	CONTINUE	SIMKT137
	IF(OMEGM.NE.0.0)GO TO 1400	SIMKT138
	IF((ND.EQ.0).OR.(NN.EQ.0))GO TO 1340	SIMKT139
	CALL DFN(HS.MT.MTFB.ND.NN.N.TD,IPRINT,IW)	SIMKT140
	NNX(N)=ND-1	SIMKT141
	GO TO 1200	SIMKT142
1400	CONTINUE	SIMKT143
	IF(DELPHM.LE.0.0)GO TO 1340	SIMKT144
	DO 1420 ND=2.5	SIMKT145
	NNX(N)=ND-1	SIMKT146
	NDM=ND	SIMKT147
	IF(ND.EQ.5)NDM=4	SIMKT148
	DO 1420 NN=1,NDM	SIMKT149
	CALL DFN(HS.MT.MTFB.ND.NN.N.TD,IPRINT,IW)	SIMKT150
	CALL PHERR(HS.MT.MTFB.ND.N.OMEGM.TD,DELPH,IPRINT,IW)	SIMKT151
	IF(DELPH.LE.DELPHM)GO TO 1200	SIMKT152
1420	CONTINUE	SIMKT153
	WRITE(IW,1430)DELPHM,DELPH	SIMKT154
1430	FORMAT(IH1,///,1X,30)TIME DELAY SPECIFICATION CANNOT BE MET,///,	SIMKT155
	11X,20)ALLOWED PHASE ERROR=,E12.6,///,	SIMKT156
	21X,20)ACTUAL PHASE ERROR =,E12.6,///	SIMKT157
	GO TO 1200	SIMKT158
1440	CONTINUE	SIMKT159
C		SIMKT160
C	PRINT ERROR MESSEGE	SIMKT161
C		SIMKT162
	WRITE(IW,1460)	SIMKT163
1460	FORMAT(IH1,///,1X,37)DATA CONTROL CARD SPECIFICATION ERROR)	SIMKT164
	STOP 111	SIMKT165
1480	CONTINUE	SIMKT166
	NMAX=NRK	SIMKT167
	IF(IPRINT.EQ.6)CALL DEBUG(4,4)SIMK,4MT .2,0,(IW)	SIMKT168
C		SIMKT169
C	COMPUTE QUADRUPLES FOR ALL BLOCKS	SIMKT170
C		SIMKT171
	DO 1540 N=1,NMAX	SIMKT172
	CALL TRANSK(NNX,NNR,NNU,AT,RT,CT,DT,PRINT,HS,	SIMKT173
	IMST,MT,MTFB,N,NUM,NRM,IPRINT,IW)	SIMKT174
1540	CONTINUE	SIMKT175
	NX=0	SIMKT176
	DO 1560 N=1,NMAX	SIMKT177
	NX=NX+NNX(N)	SIMKT178
	NNU(N)=1	SIMKT179
1560	NNR(N)=1	SIMKT180
	NY=2*NMAX	SIMKT181
C		SIMKT182
C	READ INTERCONNECTION QUADRUPLES AND PRINT THEM	SIMKT183
C		SIMKT184
1580	CONTINUE	SIMKT185
	READ(IR,1060)CARD	SIMKT186
	DECODE(4,1100,CARD(1))CC,DUMMY	SIMKT187
	IF(CC.EQ.HC)GO TO 1580	SIMKT188
	IF(CARD(1).EQ.HEND)GO TO 1600	SIMKT189
	IF((CARD(1).EQ.HUIR).AND.(CARD(2).EQ.HI))CALL INPT(P,MTFB,MTFB)	SIMKT190
	IF((CARD(1).EQ.HUIR).AND.(CARD(2).EQ.HI))GO TO 1580	SIMKT191
	IF(CARD(1).EQ.HUIU)CALL INPT(O,MTFB,NUM)	SIMKT192
	IF(CARD(1).EQ.HUIU)GO TO 1580	SIMKT193
	IF(CARD(1).EQ.HRRI)CALL INPT(R,NRM,MTFB)	SIMKT194
	IF(CARD(1).EQ.HRRI)GO TO 1580	SIMKT195
	IF(CARD(1).EQ.HRU)CALL INPT(S,NRM,NUM)	SIMKT196

Figure 22. Subroutine SIMKT Program Listing (Continued)

```

        IF (CAPD(1).EQ.HRU) GO TO 158)
        GO TO 1440
1600 CONTINUE
        IF (IPRINT.EQ.6) CALL DFRUG(5.4HSIMK,4MT  .2.0.1W)
C
C CALCULATE NR AND NU
C
        DO 1640 J=1,NUM
        DO 1620 I=1,NMAX
        IF (Q(I,J).NE.0.0) GO TO 1660
1620 CONTINUE
        DO 1640 I=1,NRM
        IF (S(I,J).NE.0.0) GO TO 1660
1640 CONTINUE
        NU=J-1
        GO TO 1680
1660 CONTINUE
        NU=NUM
1680 CONTINUE
        IF (NU.EQ.0) GO TO 1780
        DO 1740 I=1,NRM
        DO 1700 J=1,NMAX
        IF (R(I,J).NE.0.0) GO TO 1740
1700 CONTINUE
        DO 1720 J=1,NU
        IF (S(I,J).NE.0.0) GO TO 1740
1720 CONTINUE
        NR=I-1
        GO TO 1760
1740 CONTINUE
        NR=NR+
1760 CONTINUE
        IF (NR.GT.0) GO TO 1820
1780 CONTINUE
C
C PRINT ERROR MESSAGE
C
        WRITE (IW,1800)
1800 FORMAT (IH1,/,/.1X.35HINTERCONNECTION SPECIFICATION ERROR)
        STOP 111
1820 CONTINUE
        IF ((IPRINT.NE.3).AND.(IPRINT.LT.5)) GO TO 1860
        WRITE (IW,1840)
1840 FORMAT (/,/20X.36H*** CONNECTION DATA FOR BLOCKS **/,/)
        CALL MPRS (P,MTFB,MTFB,NMAX,NMAX,T,4HP )
        CALL MPRS (Q,MTFB,NUM,NMAX,NU,T,4HQ )
        CALL MPRS (R,NRM,MTFB,NR,NMAX,T,4HR )
        CALL MPRS (S,NRM,NUM,NR,NU,T,4HS )
1860 CONTINUE
        RETURN
1900 CONTINUE
C
C COMPUTE SUBSYSTEM STATES XDOT(N)=AN*XN +BN*UN
C
        II=0
        DO 251 N=1,NMAX
        MX=NX(N)
        DO 200 I=1,MX
        II=II+1
        XDOTL(II)=0.0
        NUX = NNU(N)
        DO 201 J=1,NUX
        XDOTL(II)=XDOTL(II)+BT(I,J,N)*UI(J,N)
201 DO 200 J=1,MX
200 XDOTL(II)=XDOTL(II)+AT(I,J,N)*X(J,N)
251 CONTINUE
SIMKT197
SIMKT198
SIMKT199
SIMKT200
SIMKT201
SIMKT202
SIMKT203
SIMKT204
SIMKT205
SIMKT206
SIMKT207
SIMKT208
SIMKT209
SIMKT210
SIMKT211
SIMKT212
SIMKT213
SIMKT214
SIMKT215
SIMKT216
SIMKT217
SIMKT218
SIMKT219
SIMKT220
SIMKT221
SIMKT222
SIMKT223
SIMKT224
SIMKT225
SIMKT226
SIMKT227
SIMKT228
SIMKT229
SIMKT230
SIMKT231
SIMKT232
SIMKT233
SIMKT234
SIMKT235
SIMKT236
SIMKT237
SIMKT238
SIMKT239
SIMKT240
SIMKT241
SIMKT242
SIMKT243
SIMKT244
SIMKT245
SIMKT246
SIMKT247
SIMKT248
SIMKT249
SIMKT250
SIMKT251
SIMKT252
SIMKT253
SIMKT254
SIMKT255
SIMKT256
SIMKT257
SIMKT258
SIMKT259
SIMKT260
SIMKT261
SIMKT262

```

Figure 22. Subroutine SIMKT Program Listing (Continued)

C		SIMKT263
C	COMPUTE INTERNAL OUTPUTS RIN=CN*AN*DN*UN	SIMKT264
C		SIMKT265
	II=0	SIMKT266
	DO 350 N=1,NMAX	SIMKT267
	MX=NNP(N)	SIMKT268
	DO 300 I=1,MX	SIMKT269
	II=II+1	SIMKT270
	YL(II)=0.0	SIMKT271
	MX1=NNX(N)	SIMKT272
	DO 301 J=1,MX1	SIMKT273
301	YL(II)=YL(II)+CT(I,J,N)*X(J,N)	SIMKT274
	NX1=NNU(N)	SIMKT275
	DO 300 J=1,NX1	SIMKT276
300	YL(II)=YL(II)+DT(I,J,N)*UI(J,N)	SIMKT277
350	CONTINUE	SIMKT278
C		SIMKT279
C	INTERCONNECTION EQUATIONS	SIMKT280
C		SIMKT281
	DO 240 I=1,NMAX	SIMKT282
	II=II+1	SIMKT283
	YL(II)=0.0	SIMKT284
	DO 230 J=1,NMAX	SIMKT285
230	YL(II)=YL(II)+P(I,J)*RI(I,J)	SIMKT286
	DO 240 J=1,NU	SIMKT287
240	YL(II)=YL(II)+Q(I,J)*U(J)	SIMKT288
C		SIMKT289
C	EXTERNAL RESPONS EQUATIONS	SIMKT290
C		SIMKT291
	II=0	SIMKT292
	DO 280 I=1,NR	SIMKT293
	II=II+1	SIMKT294
	RL(II)=0.0	SIMKT295
	DO 270 J=1,NMAX	SIMKT296
270	RL(II)=RL(II)+R(I,J)*RI(I,J)	SIMKT297
	DO 280 J=1,NU	SIMKT298
280	RL(II)=RL(II)+S(I,J)*U(J)	SIMKT299
	RETURN	SIMKT300
	END	SIMKT301

Figure 22. Subroutine SIMKT Program Listing (Concluded)

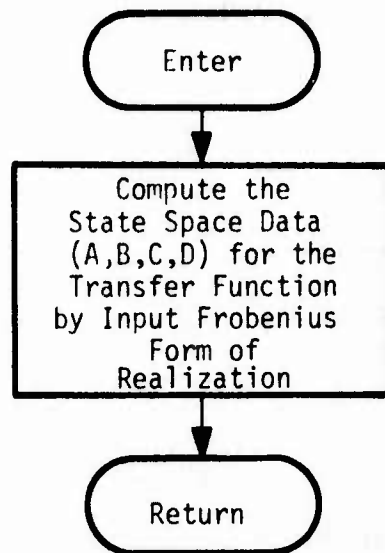


Figure 23. Subroutine TRANSK Flow Chart

	SUBROUTINE TRANSK (NNX,NNR,NNU,AT,RT,CT,DT,PRINT,HS, MST,MT,MTF,NUM,NRM,IPRINT,IW)	TRANSK 2
C		TRANSK 3
C	PURPOSE - TO COMPUTE QUADRUPLES FOR RATIONAL TRANSFER FUNCTION	TRANSK 4
C	ANALYSIS - A F KOVAR / J K MAHESH - THE HONEYWELL INC	TRANSK 5
C	DATE WRITTEN - 1970	TRANSK 6
C		TRANSK 7
C	SUBPROGRAMS CALLED	TRANSK 8
C	DEBUG	TRANSK 9
C		TRANSK10
C	ARGUMENTS LIST	TRANSK11
C	N INPUT TRANSFER FN BLOCK NO	TRANSK12
C	IPRINT INPUT PRINT CONTROL FLAG	TRANSK13
C	IW INPUT FILE NO FOR LINE PRINTED	TRANSK14
C	OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM	TRANSK15
C		TRANSK16
C	DIMENSION NNX(MTF),NNR(MTF),NNU(MTF)	TRANSK17
C	DIMENSION HS(2,MT,MTF),PRINT(2,MT)	TRANSK18
C	DIMENSION AT(MST,MST,MTF),RT(MST,1,MTF)	TRANSK19
C	DIMENSION CT(1,MST,MTF),DT(1,1,MTF)	TRANSK20
C		TRANSK21
C	ZERO OUT STORAGE SPACE	TRANSK22
C		TRANSK23
C	IF (IP-INT.EQ.5) CALL DEBUG(1.4-TRAN,4MSK .2,0.1W)	TRANSK24
C	MX=NNX(N)	TRANSK25
C	MXM=NX(N)-1	TRANSK26
C	MXP=NX(N)+1	TRANSK27
C	DO 1 I=1,MX	TRANSK28
C	DO 1 J=1,MX	TRANSK29
C	AT(I,I,N)=0.0	TRANSK30
C	DO 2 I=1,MX	TRANSK31
C	J=1	TRANSK32
C	RT(I,I,N)=0.0	TRANSK33
C	DO 3 I=1,MX	TRANSK34
C	I=1	TRANSK35
C	CT(I,I,N)=0.0	TRANSK36
C	DT(I,I,N)=0.0	TRANSK37
C		TRANSK38
C	COMPUTE AT,RT	TRANSK39
C		TRANSK40
C	SET OFF DIAGONAL TERMS IN A TO UNITY	TRANSK41
C		TRANSK42
C	DO 50 I=1,MXM	TRANSK43
C	AT(I,I+1,N)=1.	TRANSK44
C	RT(NNX(N),I,N)=1./HS(2,I,N)	TRANSK45
C		TRANSK46
C	COMPUTE LAST ROW OF A	TRANSK47
C		TRANSK48
C	IF (IP-INT.EQ.6) CALL DEBUG(2.4-TRAN,4MSK .2,0.1W)	TRANSK49
C	DO 10 J=1,MX	TRANSK50
C	AT(NNX(N),J,N)=-HS(2,NNX(N)+2-J,N)*RT(NNX(N),1,N)	TRANSK51
C		TRANSK52
C	COMPUTE CT,DT	TRANSK53
C		TRANSK54
C	DO 20 J=1,MX	TRANSK55
C	CT(1,I,N)=HS(1,NNX(N)+2-J,N)	TRANSK56
C	IF (HS(1,1,N).EQ.0.) GO TO 40	TRANSK57
C	DO 30 J=1,MX	TRANSK58
C	CT(1,I,N)=CT(1,I,N)+AT(NNX(N),J,N)*HS(1,1,N)	TRANSK59
C	DT(1,I,N)=RT(NNX(N),1,N)*HS(1,1,N)	TRANSK60
C	CONTINUE	TRANSK61
C	IF (IP-INT.EQ.6) CALL DEBUG(3.4-TRAN,4MSK .2,0.1W)	TRANSK62
C	RETURN	TRANSK63
C	END	TRANSK64
C		TRANSK65

Figure 24. Subroutine TRANSK Program Listing

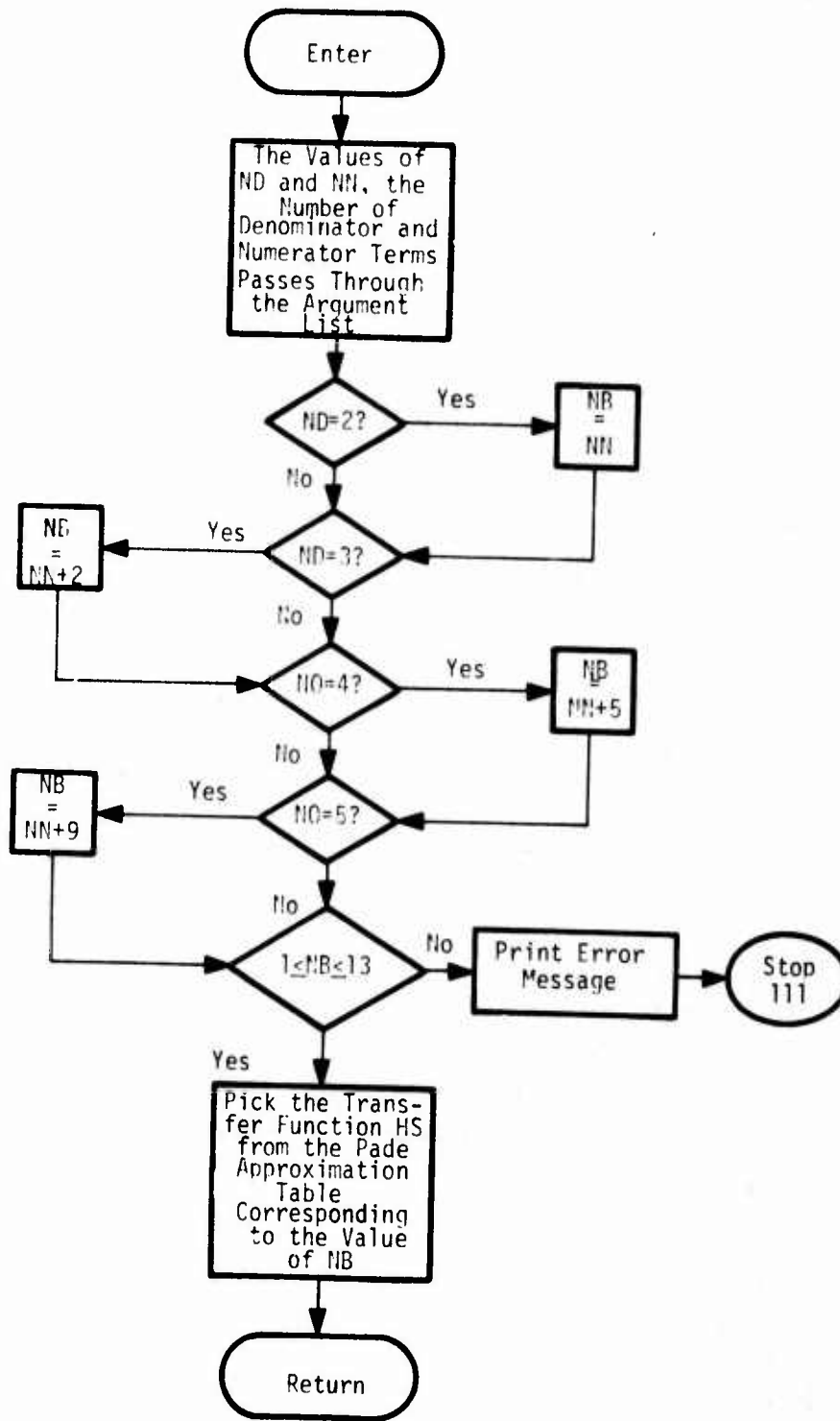


Figure 25. Subroutine DFN Flow Chart

	SUBROUTINE DFN(HS,MT,MTR,ND,NN,TD,IPRINT,IW)	DFN	2
C		DFN	3
C	PURPOSE - TO PICK A PADE APPROXIMATION TO TIME DELAY	DFN	4
C	ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC	DFN	5
C	DATE WRITTEN - 1975	DFN	6
C		DFN	7
C	ARGUMENTS LIST	DFN	8
C	ND INPUT NO OF DENOMINATOR TERMS IN THE TR FN	DFN	9
C	NN INPUT NO OF NUMERATOR TERMS IN THE TR FN	DFN	10
C	N INPUT TRANSFER FN BLOCK NO	DFN	11
C	TD INPUT TIME OR TRANSPORT DELAY	DFN	12
C	IPRINT INPUT PRINT CONTROL FLAG	DFN	13
C	IW INPUT FILE NO FOR LINE PRINTER	DFN	14
C	OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM	DFN	15
C		DFN	16
C	DIMENSION HS(2,MT,MTR)	DFN	17
C	IF (IPRINT.EQ.5) CALL DERUG(1,4HDFN,4H,2.0,IW)	DFN	18
C	IF (ND.EQ.2) NR=NN	DFN	19
C	IF (ND.EQ.3) NR=NN+2	DFN	20
C	IF (ND.EQ.4) NR=NN+5	DFN	21
C	IF (ND.EQ.5) NR=NN+9	DFN	22
C	IF ((NR.GT.13).OR.(NR.LT.1)) GO TO 660	DFN	23
C	IF (IPRINT.EQ.6) CALL DERUG(2,4HDFN,4H,2.0,IW)	DFN	24
C	GO TO (510,520,530,540,550),560,570,580,590,600,610,620,630)NB	DFN	25
C		DFN	26
C	FIRST ORDER PADE APPROXIMATIONS	DFN	27
C		DFN	28
C	510 CONTINUE	DFN	29
	HS(1,2,N)=1.0	DFN	30
	HS(2,1,N)=TD	DFN	31
	HS(2,2,N)=1.0	DFN	32
	GO TO 650	DFN	33
	520 CONTINUE	DFN	34
	HS(1,1,N)=-TD/2.0	DFN	35
	HS(1,2,N)=1.0	DFN	36
	HS(2,1,N)=TD/2.0	DFN	37
	HS(2,2,N)=1.0	DFN	38
	GO TO 650	DFN	39
C		DFN	40
C	SECOND ORDER PADE APPROXIMATIONS	DFN	41
C		DFN	42
C	530 CONTINUE	DFN	43
	HS(1,3,N)=1.0	DFN	44
	HS(2,1,N)=TD*TD/2.0	DFN	45
	HS(2,2,N)=TD	DFN	46
	HS(2,3,N)=1.0	DFN	47
	GO TO 650	DFN	48
	540 CONTINUE	DFN	49
	HS(1,2,N)=-TD/3.0	DFN	50
	HS(1,3,N)=1.0	DFN	51
	HS(2,1,N)=TD*TD/6.0	DFN	52
	HS(2,2,N)=2.0*TD/3.0	DFN	53
	HS(2,3,N)=1.0	DFN	54
	GO TO 650	DFN	55
	550 CONTINUE	DFN	56
	HS(1,1,N)=TD*TD/12.0	DFN	57
	HS(1,2,N)=-TD/2.0	DFN	58
	HS(1,3,N)=1.0	DFN	59
	HS(2,1,N)=TD*TD/12.0	DFN	60
	HS(2,2,N)=TD/2.0	DFN	61
	HS(2,3,N)=1.0	DFN	62
	GO TO 650	DFN	63
C		DFN	64

Figure 26. Subroutine DFN Program Listing

C	THIRD ORDER PADE APPROXIMATIONS	DFN	65
C		DFN	66
560	CONTINUE	DFN	67
	HS(1.1,N)=1.0	DFN	68
	HS(2.1,N)=TD**7/6.0	DFN	69
	HS(2.2,N)=TD*TD/2.0	DFN	70
	HS(2.3,N)=TD	DFN	71
	HS(2.4,N)=1.0	DFN	72
	GO TO 650	DFN	73
570	CONTINUE	DFN	74
	HS(1.1,N)=-TD/4.0	DFN	75
	HS(2.1,N)=TD**7/24.0	DFN	76
	HS(1.4,N)=1.0	DFN	77
	HS(2.2,N)=TD*TD/4.0	DFN	78
	HS(2.3,N)=TD*3.0/4.0	DFN	79
	HS(2.4,N)=1.0	DFN	80
	GO TO 650	DFN	81
580	CONTINUE	DFN	82
	HS(1.2,N)=TD*TD/20.0	DFN	83
	HS(1.3,N)=-2.0*TD/5.0	DFN	84
	HS(1.4,N)=1.0	DFN	85
	HS(2.1,N)=TD**7/60.0	DFN	86
	HS(2.2,N)=3.0*TD*TD/20.0	DFN	87
	HS(2.3,N)=TD*3.0/5.0	DFN	88
	HS(2.4,N)=1.0	DFN	89
	GO TO 650	DFN	90
590	CONTINUE	DFN	91
	HS(1.1,N)=-TD**7/120.0	DFN	92
	HS(1.2,N)=TD*TD/10.0	DFN	93
	HS(1.3,N)=-TD/2.0	DFN	94
	HS(1.4,N)=1.0	DFN	95
	HS(2.1,N)=TD**7/120.0	DFN	96
	HS(2.2,N)=TD*TD/10.0	DFN	97
	HS(2.3,N)=TD/2.0	DFN	98
	HS(2.4,N)=1.0	DFN	99
	GO TO 650	DFN	100
C		DFN	101
C	FOURTH ORDER PADE APPROXIMATIONS	DFN	102
C		DFN	103
600	CONTINUE	DFN	104
	HS(1.1,N)=1.0	DFN	105
	HS(2.1,N)=TD**4/24.0	DFN	106
	HS(2.2,N)=TD**7/6.0	DFN	107
	HS(2.3,N)=TD*TD/2.0	DFN	108
	HS(2.4,N)=TD	DFN	109
	HS(2.5,N)=1.0	DFN	110
	GO TO 650	DFN	111
610	CONTINUE	DFN	112
	HS(1.4,N)=-TD/5.0	DFN	113
	HS(1.5,N)=1.0	DFN	114
	HS(2.1,N)=TD**4/120.0	DFN	115
	HS(2.2,N)=2.0*TD**7/30.0	DFN	116
	HS(2.3,N)=TD*TD*3.0/10.0	DFN	117
	HS(2.4,N)=TD*4.0/5.0	DFN	118
	HS(2.5,N)=1.0	DFN	119
	GO TO 650	DFN	120
620	CONTINUE	DFN	121
	HS(1.2,N)=TD*TD/30.0	DFN	122
	HS(1.4,N)=-TD/7.0	DFN	123
	HS(1.5,N)=1.0	DFN	124
	HS(2.1,N)=TD**4/360.0	DFN	125
	HS(2.2,N)=TD**7/30.0	DFN	126
	HS(2.3,N)=TD*TD*2.0/10.0	DFN	127
	HS(2.4,N)=TD*2.0/3.0	DFN	128
	HS(2.5,N)=1.0	DFN	129
	GO TO 650	DFN	130

Figure 26. Subroutine DFN Program Listing (Continued)

630	CONTINUE	DFN	131
	HS(1.0,N)=-TD**3/210.0	DFN	132
	HS(1.2,N)=TD*TD/14.0	DFN	133
	HS(1.4,N)=-3.0*TD/7.0	DFN	134
	HS(1.6,N)=1.0	DFN	135
	HS(2.1,N)=TD**4/84.0	DFN	136
	HS(2.3,N)=4.0*TD**3/210.0	DFN	137
	HS(2.5,N)=TD*TD*2. /14.0	DFN	138
	HS(2.7,N)=TD*6. /7.0	DFN	139
	HS(2.9,N)=1.0	DFN	140
650	CONTINUE	DFN	141
	IF (TP-INT.EQ.6)CALL DEBUG(3.4HDFN .4H .2.0*TW)	DFN	142
	RTURN	DFN	143
C		DFN	144
C	PRINT ERROR MESSAGE	DFN	145
C		DFN	146
660	CONTINUE	DFN	147
	WRITE(IW,670)	DFN	148
670	FORMAT(1H1.//.1X.43DIMENSIONS FOR TIME DELAY EXCEEDS THE LIMIT)	DFN	149
	STOP 111	DFN	150
	END	DFN	151

Figure 26. Subroutine DFN Program Listing (Concluded)

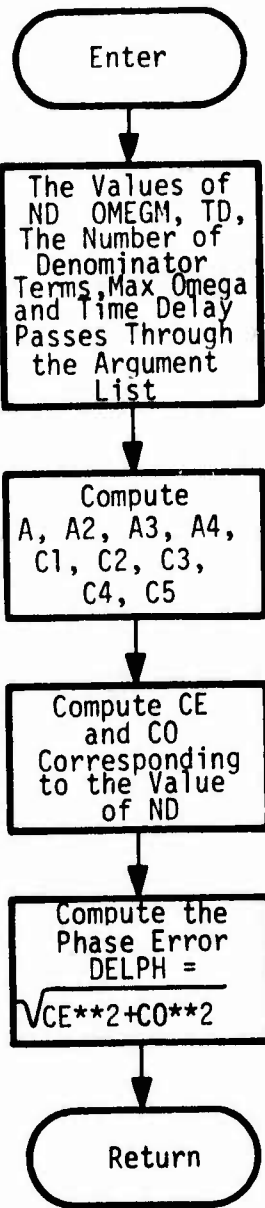


Figure 27. Subroutine PHERR Flow Chart

```

C      SUBROUTINE PHERR(HS,MT,MTR,ND,N,OMEGA,TU,DELPH,I,PRINT,IW)      PHERR  2
C      PURPOSE - TO COMPUTE PHASE ERROR OF PADE APPROXIMATION TO TIME DELPH      PHERR  3
C      ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC      PHERR  4
C      DATE WRITTEN - 1975      PHERR  5
C      PHERR  6
C      SUBPROGRAMS CALLED      PHERR  7
C      DERUG      PHERR  8
C      PHERR  9
C      ARGUMENTS LIST      PHERR 10
C      ND      INPUT      NO OF DENOMINATOR TERMS IN THE TR FN      PHERR 11
C      N      INPUT      TRANSFER FN BLOCK NO      PHERR 12
C      OMEGA      INPUT      MAXIMUM FREQUENCY FOR COMPUTING PHASE ERROR      PHERR 13
C      TD      INPUT      TIME OR TRANSPORT DELAY      PHERR 14
C      DELPH      OUTPUT      PHASE ERROR      PHERR 15
C      IPRINT      INPUT      PRINT CONTROL FLAG      PHERR 16
C      IW      INPUT      FILE NO FOR LINE PRINTER      PHERR 17
C      OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM      PHERR 18
C      PHERR 19
C      DIMENSION HS(2,MT,MTR)      PHERR 20
C      IF (IPRINT.EQ.6)CALL DERUG(1,4,PHERR,4HR      .2,0,IW)      PHERR 21
C      A=OMEGA*TD      PHERR 22
C      A2=A**2      PHERR 23
C      A3=A**3      PHERR 24
C      A4=A**4      PHERR 25
C      C1=HS(2,1,N)*A-HS(1,1,N)      PHERR 26
C      C2=HS(2,2,N)*A-HS(1,2,N)      PHERR 27
C      C3=HS(2,3,N)*A-HS(1,3,N)      PHERR 28
C      C4=HS(2,4,N)*A-HS(1,4,N)      PHERR 29
C      C5=HS(2,5,N)*A-HS(1,5,N)      PHERR 30
C      NO=ND-1      PHERR 31
C      IF (IPRINT.EQ.6)CALL DERUG(2,4,PHERR,4HR      .2,0,IW)      PHERR 32
C      GO TO(110,120,130,140)NO      PHERR 33
110  CONTINUE      PHERR 34
C      CE=C2      PHERR 35
C      CO=C1*A      PHERR 36
C      GO TO 150      PHERR 37
120  CONTINUE      PHERR 38
C      CE=C3-C1*A2      PHERR 39
C      CO=C2*A      PHERR 40
C      GO TO 150      PHERR 41
130  CONTINUE      PHERR 42
C      CE=C4-C2*A2      PHERR 43
C      CO=C3*A-C1*A3      PHERR 44
C      GO TO 150      PHERR 45
140  CONTINUE      PHERR 46
C      CE=C5-C3*A2+C1*A4      PHERR 47
C      CO=C4*A-C2*A3      PHERR 48
150  CONTINUE      PHERR 49
C      CE2=CE**2      PHERR 50
C      CO2=CO**2      PHERR 51
C      DELPH=SQR(CE2+CO2)      PHERR 52
C      IF (IPRINT.EQ.6)CALL DERUG(3,4,PHERR,4HR      .2,0,IW)      PHERR 53
C      RETURN      PHERR 54
C      END      PHERR 56

```

Figure 28. Subroutine PHERR Program Listing

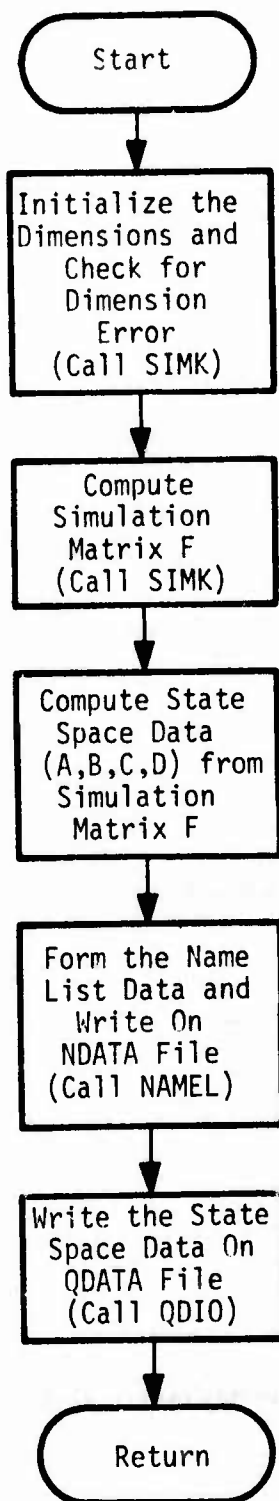


Figure 29. Subroutine STAMK3 Flow Chart


```

C      MS2      INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S2      STAMK365
C      MS3      INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S3      STAMK366
C      MS4      INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S4      STAMK367
C      NR      INPUT      MAXIMUM SYSTEM NO - IMPLICIT MODEL      STAMK368
C      NRM*MH   INPUT      MAXIMUM DIMENSION FOR RIN      STAMK369
C                                                    STAMK370
COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
COMMON /SYS/ SCODE,SDS(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)
1,PHEAD(20)
COMMON /SC1/ S1(1)
C      DIMENSION DESS(NX*10,MB),UNITSS(NX*4,MB)
C      DIMENSION DESO(NR*10,MB),UNITO(NR*4,MB)
C      DIMENSION DESII(NUM,10,MB),UNITII(NUM,4,MB)
C      DIMENSION NXX(MR),NRR(MR),NUU(MR)
COMMON /SC2/ S2(1)
C      DIMENSION ATC(NX*NX*MR),BTC(NX*NUM,MB)
C      DIMENSION CTC(NR*NX*MR),DTC(NR*NUM,MB)
C      DIMENSION PC(MN,MN),QC(MN,NJM),RC(NR,MN),SC(NM,NUM)
COMMON /SC3/ S3(1)
C      DIMENSION PP(MP,MM,MM),QQ(MQ,MM,NUM),RR(MR,NR*MM)
C      DIMENSION NSP(MP),NSQ(MQ),NSR(MR)
C      DIMENSION V(MAXN),W(MAXM),F(MAXN,MAXM)
C      DIMENSION XDOT(NX*MB),X(NX*MB),RI(NR*MR),UI(NUM,MB)
C      DIMENSION RIN(NR*MR),II(NUM),NXX(MR),NRR(MR),NNI(MR)
C      DIMENSION A(NX*NX),R(NX,NUM),C(NR,NX),D(NR,NUM)
C      DIMENSION NNS(NX),VNS(NX,2),DESS(NX,10),UNITS(NX,4)
C      DIMENSION NNO(NR),VNO(NR,2),DESO(NR,10),UNITO(NR,4)
C      DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
DATA HDINT/4HSINT/
C
C      PRINT SYSTEM DIMENSIONS IF NEEDED
C
C      IF(IPRINT.EQ.6)WRITE(IW,165)MS1,MS2,MS3,MS4,MAXN,MAXM
1,NX,NR,NUM,NYM,MM,MP,MQ,MR,NA,NS,MN
165 FORMAT(1X,15(15,1X))
C
C      COMPUTE ARRAY START INDEXES
C
C
C      FOR DESS,UNITSS,DESO,UNITO,DESII,UNITII,NXX,NRR,NUU
C
C      L1=1 $ L2=L1+NX*MB*10 $ L3=L2+NX*MB*4 $ L4=L3+NR*MB*10
L5=L4+NR*MB*4 $ L6=L5+NUM*MB*10 $ L7=L6+NUM*MB*4
L8=L7+MB $ L9=L8+MR $ L10=L9+MR
C
C      FOR ATC,BTC,CTC,DTC,PC,QC,RC,SC
C
C      M1=1 $ M2=M1+NX*NX*MR $ M3=M2+NX*NUM*MR $ M4=M3+NR*NX*MB
M5=M4+NR*NUM*MB $ M6=M5+MN*MN $ M7=M6+MN*NUM $ M8=M7+NR*MN
M9=M8+NR*NUM
C
C      FOR PP,QQ,RR,NSP,NSQ,NSR
C
C      K1=1 $ K2=K1+MP*MM*MM $ K3=K2+MQ*MM*NUM $ K4=K3+MR*NR*MM
K5=K4+MP $ K6=K5+MQ $ K7=K6+MR
IF(IPRINT.EQ.6)WRITE(IW,165)L1,L2,L3,L4,L5,L6,L7,L8,L9,L10
IF(IPRINT.EQ.6)WRITE(IW,165)M1,M2,M3,M4,M5,M6,M7,M8,M9
IF(IPRINT.EQ.6)WRITE(IW,165)K1,K2,K3,K4,K5,K6,K7
C
C      CHECK IF SCRATCH ARRAY SIZES ARE SUFFICIENT
C
C      IF((L10.GT.MS1).OR.(M9.GT.MS2).OR.(K7.GT.MS3))
ICALL DERRM(L10,M9,K7,MS4,MS1,MS2,MS3,MS4,3,0,4HSTAM,4HK3 ,IW)
IF(SCODE.EQ.HDINT)GO TO 5
CALL QUADK(A,B,C,D,NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,
INNT,VNT,DESI,UNITI,S1(L1),S1(L2),S1(L3),S1(L4),S1(L5),S1(L6),
STAMK126
STAMK127
STAMK128
STAMK129
STAMK130

```

Figure 30. Subroutine STAMK3 Program Listing (Continued)

	2S1(L7),S1(L8),S1(L9),NX4,NRM,NUM,4B,NB)	STAMK131
	RETURN	STAMK132
5	CONTINUE	STAMK133
	NR1=0 \$ NR2=0 \$ NR3=0 \$ NU1=0 \$ NU2=0 \$ NU3=0	STAMK134
	NXA=0 \$ NRA=0 \$ NUA=0	STAMK135
	EPSF=1.0E-30 \$ T=0.0 \$ INIT=0 \$ NFLAG=1	STAMK136
	IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))CALL HPR(HEAD,1W)	STAMK137
C		STAMK138
C	INITIALIZING CALL TO SUBROUTINE SIMK	STAMK139
C		STAMK140
	NX=0 \$ NR=0 \$ NU=0 \$ NY=0	STAMK141
	N1=1 \$ N2=N1+NX \$ N3=N2+NY	STAMK142
	CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,!,RIN,NNX,NNR,NNU,	STAMK143
	S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),	STAMK144
	2S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,	STAMK145
	3MB,MM,MP,MO,MR,NXM,NUM,NRM,4N,INIT,T,NXA,NRA,NUA,NB,NRMMB)	STAMK146
C		STAMK147
C	CHECK FOR DIMENSION ERROR	STAMK148
C		STAMK149
	INIT = 1	STAMK150
	M=2*NX+NY+NU	STAMK151
	N=NX+NY+NR	STAMK152
	IF((NX.GT.NXM).OR.(NR.GT.NRM).OR.(NU.GT.NUM).OR.(NY.GT.NYM))	STAMK153
	1CALL DERRMS(NX,NR,NU,NY,NXM,NRM,NUM,NYM,3.0,4HSTAM,4MK3 ,1W)	STAMK154
	N1=1 \$ N2=N1+NX \$ N3=N2+NY	STAMK155
C		STAMK156
C	ZERO OUT XDOT,RI,UI,X,U	STAMK157
C		STAMK158
	DO 10 NN=1,NMAX	STAMK159
	MX=NNX(NN)	STAMK160
	DO 10 J=1,MX	STAMK161
	XDOT(J,NN)=0.0	STAMK162
10	X(J,NN)=0.0	STAMK163
	DO 11 NN=1,NMAX	STAMK164
	MX=NNR(NN)	STAMK165
	DO 12 J=1,MX	STAMK166
12	RI(J,NN)=0.0	STAMK167
	MX=NNI(NN)	STAMK168
	DO 13 J=1,MX	STAMK169
13	UI(J,NN)=0.0	STAMK170
11	CONTINUE	STAMK171
	DO 14 I=1,NU	STAMK172
14	U(I)=0.0	STAMK173
C		STAMK174
C	COMPUTE PARTIALS WRT STATE DERIVATIVES	STAMK175
C		STAMK176
45	CONTINUE	STAMK177
	JJ=0	STAMK178
	DO 50 NN=1,NMAX	STAMK179
	MX=NNX(NN)	STAMK180
	DO 50 J=1,MX	STAMK181
	JJ=JJ+1	STAMK182
	XDOT(J,NN)=1.	STAMK183
	CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,	STAMK184
	S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),	STAMK185
	2S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,	STAMK186
	3MB,MM,MP,MO,MR,NXM,NUM,NRM,4N,INIT,T,NXA,NRA,NUA,NB,NRMMB)	STAMK187
	XDOT(J,NN)=0.	STAMK188
	DO 50 I=1,N	STAMK189
50	F(I,JJ)=V(I)	STAMK190
C		STAMK191
C	COMPUTE PARTIALS WRT INTERNAL OUTPUTS	STAMK192
C		STAMK193
	DO 100 NN=1,NMAX	STAMK194
	MX=NNR(NN)	STAMK195
	.DO 100 J=1,MX	STAMK196

Figure 30. Subroutine STAMK3 Program Listing (Continued)

	JJ=JJ+1	STAMK197
	RI(J,NN)=1.	STAMK198
	CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,	STAMK199
	IS2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),	STAMK200
	S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,	STAMK201
	3MR,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,NXA,NRA,NUA,NB,NRMMB)	STAMK202
	RI(J,NN)=0.	STAMK203
	DO 10 I=1,N	STAMK204
100	F(I,J)=V(I)	STAMK205
C		STAMK206
C	COMPUTE PARTIALS WPT INTERNAL INPUTS	STAMK207
C		STAMK208
	DO 15 NN=1,NMAX	STAMK209
	MX=NN*(NN)	STAMK210
	DO 15 J=1,MX	STAMK211
	JJ=JJ+1	STAMK212
	UI(J,NN)=1.	STAMK213
	CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,	STAMK214
	IS2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),	STAMK215
	S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,	STAMK216
	3MR,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,NXA,NRA,NUA,NB,NRMMB)	STAMK217
	UI(J,NN)=0.	STAMK218
	DO 15 I=1,N	STAMK219
150	F(I,J)=V(I)	STAMK220
C		STAMK221
C	COMPUTE PARTIALS WPT STATES	STAMK222
C		STAMK223
	DO 201 NN=1,NMAX	STAMK224
	MX=NN*(NN)	STAMK225
	DO 201 J=1,MX	STAMK226
	JJ=JJ+1	STAMK227
	X(J,NN)=1.	STAMK228
	CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,	STAMK229
	IS2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),	STAMK230
	S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,	STAMK231
	3MR,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,NXA,NRA,NUA,NB,NRMMB)	STAMK232
	X(J,NN)=0.	STAMK233
	DO 201 I=1,N	STAMK234
201	F(I,J)=V(I)	STAMK235
C		STAMK236
C	COMPUTE PARTIALS WPT EXTERNAL INPUTS	STAMK237
C		STAMK238
	DO 251 J=1,NU	STAMK239
	JJ=JJ+1	STAMK240
	U(J)=.	STAMK241
	CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,	STAMK242
	IS2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),	STAMK243
	S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,	STAMK244
	3MR,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,NXA,NRA,NUA,NB,NRMMB)	STAMK245
	U(J)=.	STAMK246
	DO 251 I=1,N	STAMK247
251	F(I,J)=V(I)	STAMK248
8002	CONTINUE	STAMK249
C		STAMK250
C	COMPUTE THE SIMULATION MATRIX	STAMK251
C		STAMK252
	NV=NX*NY	STAMK253
	IF(IP=INT.EQ.6)CALL MPRS(F,MAXN,MAXM,N,M,T,4HSTM)	STAMK254
	DO 51 I=1,NV	STAMK255
	DO 52 J=1,NV	STAMK256
52	F(I,J)=-F(I,J)	STAMK257
51	F(I,I)=F(I,I)+1.	STAMK258
C		STAMK259
C	XDOT ARRAY IS BEING USED AS A SCRATCH ARRAY IN TDINVR	STAMK260
C		STAMK261
	CALL TDINVR(ISOL,INSOL,NV,-M,F,MAXN,XDOT,DET)	STAMK262

Figure 30. Subroutine STAMK3 Program Listing (Continued)

IR=NV+1	STAMK263
IF=NV+NR	STAMK264
JR=JP	STAMK265
JE=M	STAMK266
DO 53 I=IR,IF	STAMK267
DO 53 J=JR,JE	STAMK268
DO 53 K=1,NV	STAMK269
53 F(I,J)=F(I,J)*F(I,K)*F(K,J)	STAMK270
DO 53 I=1,IE	STAMK271
DO 53 J=1,JE	STAMK272
IF(ABS(F(I,J)),LE,FPSP) F(I,J) = 0.0	STAMK273
530 CONTINUE	STAMK274
IF(IPRINT.EQ.6)CALL MPPS(F,MAXN,MAXM,N,M,T,4HSIM1)	STAMK275
C	STAMK276
C FORM A,B,C,D MATRICES	STAMK277
C	STAMK278
J1=NV+1	STAMK279
J2=NV+NX	STAMK280
J3=J1+NX	STAMK281
J4=J2+NU	STAMK282
I1=NV+1	STAMK283
I2=NV+NR	STAMK284
DO 60 1 I=1,NX	STAMK285
DO 60 1 J=J1,J2	STAMK286
JJ=J-1+1	STAMK287
6001 A(I,J)=F(I,J)	STAMK288
DO 60 2 I=1,NX	STAMK289
DO 60 2 J=J3,J4	STAMK290
JJ=J-1+1	STAMK291
6002 R(I,J)=F(I,J)	STAMK292
DO 60 3 I=I1,I2	STAMK293
II=I-1+1	STAMK294
DO 60 3 J=J1,J2	STAMK295
JJ=J-1+1	STAMK296
6003 C(II,JJ)=F(I,J)	STAMK297
DO 60 4 I=I1,I2	STAMK298
II=I-1+1	STAMK299
DO 60 4 J=J3,J4	STAMK300
JJ=J-1+1	STAMK301
6004 D(II,IJ)=F(I,J)	STAMK302
C	STAMK303
C UPDATE NAME LIST DATA	STAMK304
C	STAMK305
KR=NMAX	STAMK306
CALL NAMEL(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,	STAMK307
IOESI,INITI,S1(L1),S1(L2),S1(L3),S1(L4),S1(L5),S1(L6),	STAMK308
2S1(L7),S1(L8),S1(L9),NXM,NRM,NUM,NX,NR,NU,NFLAG,MR,KB,NB)	STAMK309
C	STAMK310
C WRITE QUADRUPLE DATA ON FILE QDATA	STAMK311
C	STAMK312
IQ=0	STAMK313
MFLAG=2	STAMK314
CALL QDIO(A,B,C,D,S1,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA,	STAMK315
INR1,NP2,NR3,NU1,NU2,NU3,T,IQ,IPRINT,IW,JU,HEAD,MARK,	STAMK316
2LOCATE=NULL,INSERT,MFLAG)	STAMK317
RETURN	STAMK318
END	STAMK319

Figure 30. Subroutine STAMK3 Program Listing (Concluded)

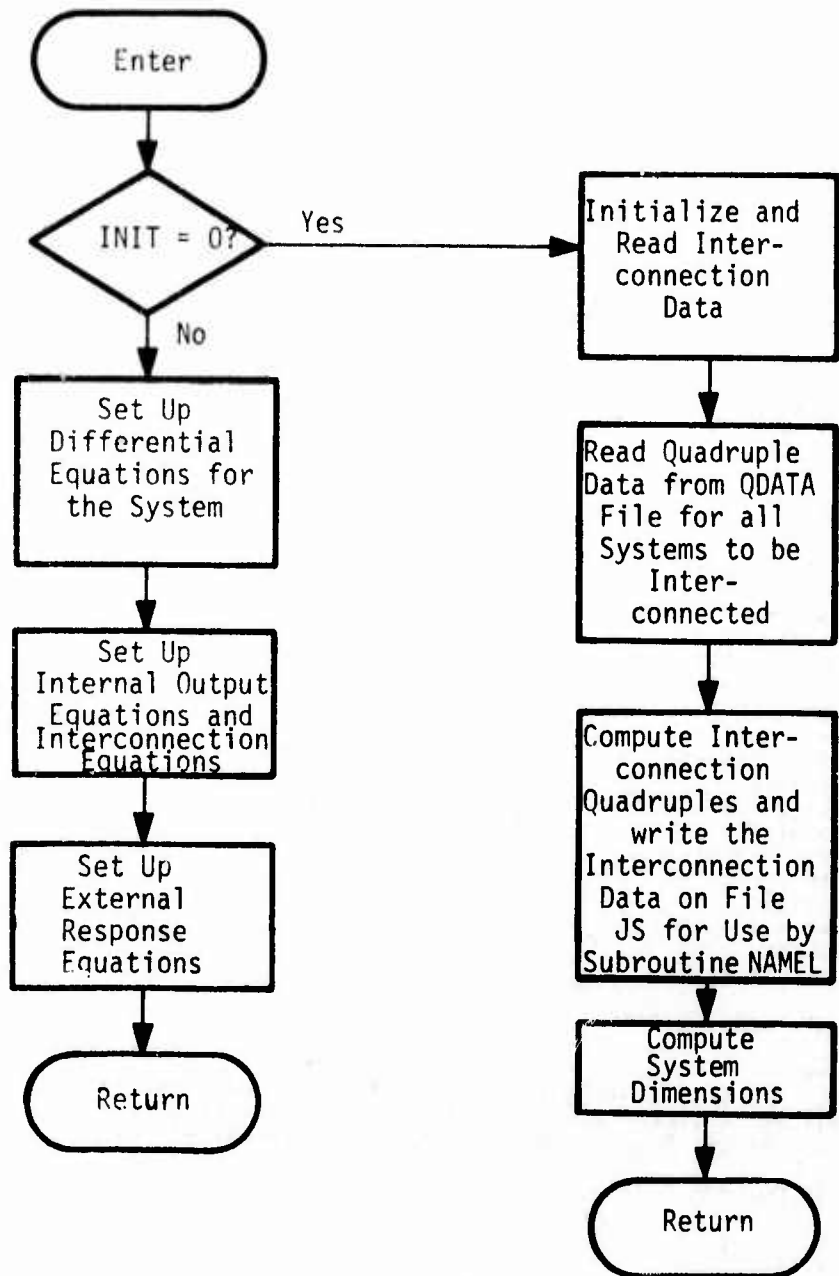


Figure 31. Subroutine SIMK Flow Chart

T=0.C	SIMK 65
NX1=0 & NR1=0 & NU1=0	SIMK 66
IF(IP=INT.EQ.6)WRITE(1W,900)KP,MB,MM,MP,MQ,MR,MS,NUM,NRM,MN,NB	SIMK 67
900 FORMAT(15(1X,I2))	SIMK 68
DO 96 I=1,NR	SIMK 69
960 NSYS(I)=-1	SIMK 70
DO 97 I=1,MP	SIMK 71
970 NSP(I)=0	SIMK 72
DO 98 I=1,MQ	SIMK 73
980 NSQ(I)=0	SIMK 74
DO 99 I=1,MR	SIMK 75
990 NSR(I)=0	SIMK 76
CALL ZERO(P,MN,MN)	SIMK 77
CALL ZERO(Q,MN,NUM)	SIMK 78
CALL ZERO(R,NRM,MN)	SIMK 79
CALL ZERO(S,NRM,NUM)	SIMK 80
DO 1010 J=1,MM	SIMK 81
DO 1010 K=1,MM	SIMK 82
DO 1010 I=1,MP	SIMK 83
1000 PP(I,J,K)=P(J,K)	SIMK 84
DO 1010 K=1,NUM	SIMK 85
DO 1010 I=1,MQ	SIMK 86
1010 QQ(I,J,K)=Q(J,K)	SIMK 87
DO 1020 J=1,NRM	SIMK 88
DO 1020 K=1,MM	SIMK 89
DO 1020 I=1,MR	SIMK 90
1020 RR(I,J,K)=R(J,K)	SIMK 91
C	SIMK 92
C READ INTERCONNECTION DATA	SIMK 93
C	SIMK 94
1040 CONTINUE	SIMK 95
READ(IP,1060)CARD	SIMK 96
1060 FORMAT(20A4)	SIMK 97
DECODE(4,1100,CARD(1))CC,DUMMY	SIMK 98
1100 FORMAT(A1,A3)	SIMK 99
1120 FORMAT(A2,11,A1)	SIMK 100
1140 FORMAT(A2,A2)	SIMK 101
IF(CC.EQ.HC)GO TO 1040	SIMK 102
IF(CARD(1).EQ.HEND)GO TO 1400	SIMK 103
DECODE(4,1140,CARD(1))CODE1.CODE2	SIMK 104
IF(CODE1.EQ.HRS)GO TO 1200	SIMK 105
DECODE(4,1120,CARD(1))CODE1,NSY1,DUMMY	SIMK 106
IF(CODE1.EQ.HU)GO TO 1160	SIMK 107
GO TO 1400	SIMK 108
C	SIMK 109
C UPDATE THE SYSTEM NUMBER COUNTER NSYS	SIMK 110
C	SIMK 111
1160 CONTINUE	SIMK 112
IF(KB.EQ.0)GO TO 1175	SIMK 113
DO 1170 N=1,KN	SIMK 114
IF(NSY1.EQ.NSYS(N))GO TO 1180	SIMK 115
1170 CONTINUE	SIMK 116
1175 CONTINUE	SIMK 117
KB=KB+1	SIMK 118
IF(KB.GT.MB)GO TO 1440	SIMK 119
NSYS(KB)=NSY1	SIMK 120
1180 CONTINUE	SIMK 121
DECODE(4,1120,CARD(2))CODE2,NSY2,DUMMY	SIMK 122
IF(CODE2.EQ.HR)GO TO 1320	SIMK 123
DECODE(4,1100,CARD(2))CODE2,DUMMY	SIMK 124
IF(CODE2.EQ.HU)GO TO 1360	SIMK 125
GO TO 1400	SIMK 126
1200 CONTINUE	SIMK 127
IF(CODE2.EQ.HR)GO TO 1220	SIMK 128
IF(CODE2.EQ.HU)GO TO 1280	SIMK 129
GO TO 1400	SIMK 130

Figure 32. Subroutine SIMK Program Listing (Continued)

C		SIMK 131
C	READ R MATRIX (R/RIM) INTO PROPER AREA OF RR MATRIX	SIMK 132
C		SIMK 133
1220	CONTINUE	SIMK 134
	DECODE(4,1240,CARD(2),NSY2,DUMMY	SIMK 135
1240	FORMAT(I1,A3)	SIMK 136
	KR=KR+1	SIMK 137
	IF(KR.GT.MR)GO TO 1470	SIMK 138
	NSR(KR)=NSY2	SIMK 139
	CALL ZERO(R,NRM,MN)	SIMK 140
	CALL INPT(R,NRM,MN)	SIMK 141
	DO 1240 I=1,NRM	SIMK 142
	DO 1240 J=1,MN	SIMK 143
1260	RR(KR,I,J)=R(I,J)	SIMK 144
	GO TO 1040	SIMK 145
C		SIMK 146
C	READ S MATRIX (R/U)	SIMK 147
C		SIMK 148
1290	CONTINUE	SIMK 149
	CALL ZERO(S,NRM,NUM)	SIMK 150
	CALL INPT(S,NRM,NUM)	SIMK 151
	GO TO 1040	SIMK 152
C		SIMK 153
C	READ P MATRIX (UIN/RIM) INTO PROPER AREA OF PP MATRIX	SIMK 154
C		SIMK 155
1320	CONTINUE	SIMK 156
	NSY=NR*(NSY1-1)+NSY2	SIMK 157
	KP=KP+1	SIMK 158
	IF(KP.GT.MP)GO TO 1470	SIMK 159
	NSP(KP)=NSY	SIMK 160
	CALL ZERO(P,MN,MN)	SIMK 161
	CALL INPT(P,MN,MN)	SIMK 162
	DO 1340 I=1,MN	SIMK 163
	DO 1340 J=1,MN	SIMK 164
1340	PP(KP,I,J)=P(I,J)	SIMK 165
	IF(IPRINT.LT.6)GO TO 1040	SIMK 166
	WRITE(IW,900)KP,NSY,NSP	SIMK 167
	CALL MPRS(P,MN,MN,MM,MM,0,0,4HPP)	SIMK 168
	GO TO 1040	SIMK 169
C		SIMK 170
C	READ Q MATRIX (UIN/U) INTO PROPER AREA OF QQ MATRIX	SIMK 171
C		SIMK 172
1360	CONTINUE	SIMK 173
	KQ=KQ+1	SIMK 174
	IF(KQ.GT.MQ)GO TO 1470	SIMK 175
	NSQ(KQ)=NSY1	SIMK 176
	CALL ZERO(Q,MN,NUM)	SIMK 177
	CALL INPT(Q,MN,NUM)	SIMK 178
	DO 1380 I=1,MN	SIMK 179
	DO 1380 J=1,NUM	SIMK 180
1380	QQ(KQ,I,J)=Q(I,J)	SIMK 181
	GO TO 1040	SIMK 182
1400	CONTINUE	SIMK 183
C		SIMK 184
C	PRINT ERROR MESSEGE	SIMK 185
C		SIMK 186
	WRITE(IW,1420)	SIMK 187
1420	FORMAT(IH1,///,1X,37HDATA CONTROL CARD SPECIFICATION ERROR)	SIMK 188
	STOP 111	SIMK 189
1440	CONTINUE	SIMK 190
	WRITE(IW,1460)KR,MR	SIMK 191
1460	FORMAT(IH1,///,1X,30H100 MANY SYSTEMS FOR COMBINING,	SIMK 192
	1//,1X,5HKB = ,12,5X,5HMR = ,1?)	SIMK 193
	STOP 111	SIMK 194
1470	CONTINUE	SIMK 195
	WRITE(IW,1475)	SIMK 196

Figure 32. Subroutine SIMK Program Listing (Continued)

1475	FORMAT(IH1,/,/,IX,39H) TOO MANY INTERCONNECTIONS FOR COMBINING)	SIMK 197
	STOP III	SIMK 198
C		SIMK 199
C	OBTAIN QUADRUPLE DATA FOR SUBSYSTEMS FROM Q DATA FILE	SIMK 200
C		SIMK 201
1480	CONTINUE	SIMK 202
	DO 1490 I=1,20	SIMK 203
1490	CARD(I)=HEAD(I)	SIMK 204
	DO 1520 N=1,KB	SIMK 205
	NSY=NSYS(N)	SIMK 206
	DO 1500 I=1,20	SIMK 207
	HEAD(I)=SHEAD(NSY,I)	SIMK 208
1500	CONTINUE	SIMK 209
	CALL FILE(JQ,LOCATF,HEAD)	SIMK 210
	READ(.IO)T,NNXN,NNRN,NNUN,	SIMK 211
	1 ((AT(I,J,N),I=1,NNXN),J=1,NNXN),	SIMK 212
	2 ((BT(I,J,N),I=1,NNXN),J=1,NNUN),	SIMK 213
	3 ((CT(I,J,N),I=1,NNRN),J=1,NNXN),	SIMK 214
	4 ((DT(I,J,N),I=1,NNRN),J=1,NNUN)	SIMK 215
	NNX(N)=NNXN	SIMK 216
	NNR(N)=NNRN	SIMK 217
	NNU(N)=NNUN	SIMK 218
C		SIMK 219
C	STORE THE IMPLICIT MODEL SYSTEM DIMENSIONS SEPARATELY	SIMK 220
C		SIMK 221
	IF(NSY,NE,NR)GO TO 1510	SIMK 222
	NX1=NNXN	SIMK 223
	NR1=NNRN	SIMK 224
	NU1=NNUN	SIMK 225
1510	CONTINUE	SIMK 226
1520	CONTINUE	SIMK 227
	DO 1530 I=1,20	SIMK 228
1530	HEAD(I)=CARD(I)	SIMK 229
C		SIMK 230
C	FORM INTERCONNECTION QUADRUPLES	SIMK 231
C		SIMK 232
	CALL ZERO(P,MN,MN)	SIMK 233
	CALL ZERO(Q,MN,NUM)	SIMK 234
	CALL ZERO(R,NRM,MN)	SIMK 235
C		SIMK 236
C	FORM P MATRIX (R/R1)	SIMK 237
C		SIMK 238
	KYOUT=0	SIMK 239
	NM1=1	SIMK 240
	NM2=0	SIMK 241
	DO 1565 M=1,KB	SIMK 242
	KYOUT=KYOUT+NNR(M)	SIMK 243
	IF(M.GT.1)NM1=NM1+NNR(M-1)	SIMK 244
	NM2=NM2+NNR(M)	SIMK 245
	NSY2=NSYS(M)	SIMK 246
	DO 1533 KR=1,MR	SIMK 247
	IF(NSR(KR).EQ.NSY2)GO TO 1536	SIMK 248
1533	CONTINUE	SIMK 249
	GO TO 1545	SIMK 250
1536	CONTINUE	SIMK 251
	DO 1540 I=1,NRM	SIMK 252
	DO 1540 J=NM1,NM2	SIMK 253
	JJ=J-NM1+1	SIMK 254
1540	R(I,J)=RR(KR,I,JJ)	SIMK 255
1545	CONTINUE	SIMK 256
C		SIMK 257
C	FORM P MATRIX (UI/R1)	SIMK 258
C		SIMK 259
	NN1=1	SIMK 260
	NN2=0	SIMK 261
	DO 1562 N=1,KB	SIMK 262

Figure 32. Subroutine SIMK Program Listing (Continued)

IF (N.GT.1)NN1=NN1+NNU(N-1)	SIMK 263
NN2=NN2+NNU(N)	SIMK 264
NSY1=NSYS(N)	SIMK 265
NSY=NR*(NSY1-1)+NSY2	SIMK 266
DO 1550 KP=1,MP	SIMK 267
IF (NSP(KP).EQ.NSY)GO TO 1555	SIMK 268
1550 CONTINUE	SIMK 269
GO TO 1567	SIMK 270
1555 CONTINUE	SIMK 271
DO 1540 I=NN1,NN2	SIMK 272
II=I-NN1+1	SIMK 273
DO 1540 J=NM1,NN2	SIMK 274
JJ=J-NM1+1	SIMK 275
1560 P(II,J)=PP(KP,II,JJ)	SIMK 276
1562 CONTINUE	SIMK 277
IF (IPRINT.LT.6)GO TO 1565	SIMK 278
WRITE (I,900)KP,NSY,NSP	SIMK 279
NNP=NN2-NN1+1	SIMK 280
NMP=NN2-NM1+1	SIMK 281
CALL MPRS(P,MN,MN,NNP,NMP,0.0,4HPP)	SIMK 282
1565 CONTINUE	SIMK 283
C	SIMK 284
C FORM Q MATRIX (UI/I)	SIMK 285
C	SIMK 286
KYIN=0	SIMK 287
NN1=1	SIMK 288
NN2=0	SIMK 289
DO 1600 N=1,KN	SIMK 290
KYIN=KYIN+NNU(N)	SIMK 291
IF (N.GT.1)NN1=NN1+NNU(N-1)	SIMK 292
NN2=NN2+NNU(N)	SIMK 293
NSY1=NSYS(N)	SIMK 294
DO 1570 KQ=1,MQ	SIMK 295
IF (NSQ(KQ).EQ.NSY1)GO TO 1575	SIMK 296
1570 CONTINUE	SIMK 297
GO TO 1600	SIMK 298
1575 CONTINUE	SIMK 299
DO 1580 I=NN1,NN2	SIMK 300
II=I-NN1+1	SIMK 301
DO 1580 J=1,NUM	SIMK 302
1580 Q(II,J)=QQ(KQ,II,J)	SIMK 303
1600 CONTINUE	SIMK 304
IF (IPRINT.NE.6)GO TO 1610	SIMK 305
CALL MPRS(P,MN,MN,KYIN,KYOUT,T,4HPR)	SIMK 306
CALL MPRS(Q,MN,NUM,KYIN,NUM,T,4HOR)	SIMK 307
CALL MPRS(R,NRM,MN,NRM,KYOUT,T,4HRR)	SIMK 308
CALL MPRS(S,NRM,NUM,NRM,NUM,T,4HSR)	SIMK 309
1610 CONTINUE	SIMK 310
C	SIMK 311
C CALCULATE NR AND NI BY USING Q, R AND S MATRICES	SIMK 312
C	SIMK 313
DO 1640 J=1,NUM	SIMK 314
DO 1620 I=1,KYIN	SIMK 315
IF (Q(I,J).NE.0.0)GO TO 1660	SIMK 316
1620 CONTINUE	SIMK 317
DO 1640 I=1,NRM	SIMK 318
IF (S(I,J).NE.0.0)GO TO 1660	SIMK 319
1640 CONTINUE	SIMK 320
NU=J-1	SIMK 321
GO TO 1680	SIMK 322
1660 CONTINUE	SIMK 323
NU=NUM	SIMK 324
1680 CONTINUE	SIMK 325
IF (NU.EQ.0)GO TO 1780	SIMK 326
DO 1740 I=1,NRM	SIMK 327
DO 1710 J=1,KYOUT	SIMK 328

Figure 32. Subroutine SIMK Program Listing (Continued)

	IF (R(I,J).NE.0.0)GO TO 1740	SIMK 329
1700	CONTINUE	SIMK 330
	DO 1720 J=1,NU	SIMK 331
	IF (S(I,J).NE.0.0)GO TO 1740	SIMK 332
1720	CONTINUE	SIMK 333
	NR=I-1	SIMK 334
	GO TO 1760	SIMK 335
1740	CONTINUE	SIMK 336
	NR=NR+1	SIMK 337
1760	CONTINUE	SIMK 338
	IF (NR.GT.0)GO TO 1820	SIMK 339
C		SIMK 340
C	PRINT ERROR MESSAGE	SIMK 341
C		SIMK 342
1780	CONTINUE	SIMK 343
	WRITE(IW,1800)	SIMK 344
1800	FORMAT(1H1,/,/,1X,35HINTERCONNECTION SPECIFICATION ERROR)	SIMK 345
	STOP 111	SIMK 346
C		SIMK 347
C	CALCULATE NX AND NY	SIMK 348
		SIMK 349
1820	CONTINUE	SIMK 350
	NX=0	SIMK 351
	DO 1840 N=1,KR	SIMK 352
	NX=NX+NNX(N)	SIMK 353
1840	CONTINUE	SIMK 354
	NY=KYIN+KYOUT	SIMK 355
	IF ((IPRINT.NE.7).AND.(IPRINT.LT.5))GO TO 1880	SIMK 356
	WRITE(IW,1860)	SIMK 357
1860	FORMAT(//,20X,28H*** INTERCONNECTION DATA **/,//)	SIMK 358
	CALL MPRS(P,MN,MN,KYIN,KYOUT,T,4HP)	SIMK 359
	CALL MPRS(Q,MN,NUM,KYIN,NU,T,4HQ)	SIMK 360
	CALL MPRS(R,NRM,MN,NR,KYOUT,T,4HR)	SIMK 361
	CALL MPRS(S,NRM,NUM,NR,NU,T,4HS)	SIMK 362
C		SIMK 363
C	CALCULATE NSA,NRA AND NUA	SIMK 364
C		SIMK 365
1880	CONTINUE	SIMK 366
	NXA=NX-NX1	SIMK 367
	NRA=NR-NR1	SIMK 368
	NUA=NU-NU1	SIMK 369
C		SIMK 370
C	WRITE INTERCONNECTION DATA ON SCRATCH FILE FOR NAMEL	SIMK 371
C	TO FORM NAME LIST DATA	SIMK 372
C		SIMK 373
	REWIND JS	SIMK 374
	IF (IPRINT.EQ.6)WRITE(IW,1890)	SIMK 375
1890	FORMAT(/,1X,30HDATA ON SCRATCH FILE FOR NAMEL,/)	SIMK 376
C		SIMK 377
C	CALCULATE AND WRITE DATA TO FORM NAME LIST FOR OUTPUTS	SIMK 378
C		SIMK 379
	CARD(1)=HOUTP	SIMK 380
	WRITE(JS,1060)CARD	SIMK 381
	IF (IPRINT.EQ.6)WRITE(IW,2000)CARD	SIMK 382
2000	FORMAT(1X,20A4)	SIMK 383
	NNRK=	SIMK 384
	NNRKP=1	SIMK 385
	DO 2100 K=1,KR	SIMK 386
	NNRK=NNRK+NNR(K)	SIMK 387
	IF (K.GT.1)NNRKP=NNRKP+NNR(K-1)	SIMK 388
	DO 2100 I=1,NR	SIMK 389
	DO 2100 J=NNRKP,NNRK	SIMK 390
	IF (R(I,J).EQ.0.0)GO TO 2100	SIMK 391
C	DO 2020 II=1,NR	SIMK 392
C	IF (II.EQ.1)GO TO 2020	SIMK 393
C	IF (R(II,J).NE.0.0)GO TO 2100	SIMK 394

Figure 32. Subroutine SIMK Program Listing (Continued)

C2020	CONTINUE	SIMK 395
	DO 2040 JJ=NNRKP,NNRK	SIMK 396
	IF(JJ,EQ,J)GO TO 2040	SIMK 397
	IF(R(I,J),NE,0.0)GO TO 2100	SIMK 398
2040	CONTINUE	SIMK 399
	NNRKK=0	SIMK 400
	NNRKKD=1	SIMK 401
	DO 2070 KK=1,KB	SIMK 402
	NNRKK=NNRKK+NNR(KK)	SIMK 403
	IF(KK,GT,1)NNRKKP=NNRKKP+NNR(KK-1)	SIMK 404
	IF(KK,EQ,K)GO TO 2070	SIMK 405
	DO 2040 JJ=NNRKKP,NNRKK	SIMK 406
	IF(R(I,J),NE,0.0)GO TO 2100	SIMK 407
2060	CONTINUE	SIMK 408
2070	CONTINUE	SIMK 409
	JJJ=J-NNRKP+1	SIMK 410
	WRITE(JS,2080)I,K,JJJ	SIMK 411
2080	FORMAT(3I2)	SIMK 412
	IF(IPRINT,EQ,6)WRITE(IW,2090)I,K,JJJ	SIMK 413
2090	FORMAT(1X,3(I2,1X))	SIMK 414
2100	CONTINUE	SIMK 415
	I=-1	SIMK 416
	WRITE(JS,2080)I	SIMK 417
	IF(IPRINT,EQ,6)WRITE(IW,2090)I	SIMK 418
C		SIMK 419
C	CALCULATE AND WRITE DATA TO FORM NAME LIST FOR INPUTS	SIMK 420
C		SIMK 421
	CARD(1)=HINPU	SIMK 422
	WRITE(JS,1060)CARD	SIMK 423
	IF(IPRINT,EQ,6)WRITE(IW,2000)CARD	SIMK 424
	NNUK=)	SIMK 425
	NNUKP=1	SIMK 426
	DO 2200 K=1,KB	SIMK 427
	NNUK=NNUK+NNU(K)	SIMK 428
	IF(K,GT,1)NNUKP=NNUKP+NNU(K-1)	SIMK 429
	DO 2200 I=NNUKP,NNUK	SIMK 430
	DO 2200 J=1,NU	SIMK 431
	IF(Q(I,J),EQ,0.0)GO TO 2200	SIMK 432
	DO 2120 II=NNUKP,NNUK	SIMK 433
	IF(II,EQ,I)GO TO 2120	SIMK 434
	IF(Q(II,J),NE,0.0)GO TO 2200	SIMK 435
2120	CONTINUE	SIMK 436
	DO 2140 JJ=1,NU	SIMK 437
	IF(JJ,EQ,J)GO TO 2140	SIMK 438
	IF(Q(I,JJ),NE,0.0)GO TO 2200	SIMK 439
2140	CONTINUE	SIMK 440
	NNUKK=0	SIMK 441
	NNUKKD=1	SIMK 442
	DO 2170 KK=1,KB	SIMK 443
	NNUKK=NNUKK+NNU(KK)	SIMK 444
	IF(KK,GT,1)NNUKKP=NNUKKP+NNU(KK-1)	SIMK 445
	IF(KK,EQ,K)GO TO 2170	SIMK 446
	DO 2140 II=NNUKKP,NNUKK	SIMK 447
	IF(Q(II,J),NE,0.0)GO TO 2200	SIMK 448
2160	CONTINUE	SIMK 449
2170	CONTINUE	SIMK 450
	III=I-NNUKP+1	SIMK 451
	WRITE(JS,2080)J,K,III	SIMK 452
	IF(IPRINT,EQ,6)WRITE(IW,2090)J,K,III	SIMK 453
2200	CONTINUE	SIMK 454
	J=-1	SIMK 455
	WRITE(JS,2080)J	SIMK 456
	IF(IPRINT,EQ,6)WRITE(IW,2090)J	SIMK 457
	CARD(1)=HEND	SIMK 458
	WRITE(JS,1060)CARD	SIMK 459
	IF(IPRINT,EQ,6)WRITE(IW,2000)CARD	SIMK 460

Figure 32. Subroutine SIMK Program Listing (Continued)

RETURN	SIMK 461
C	SIMK 462
C COMPUTE SUBSYSTEM STATES XDOT(N)=AN*XN +RN*UN	SIMK 463
C	SIMK 464
100 CONTINUE	SIMK 465
II=0	SIMK 466
DO 251 N=1,KR	SIMK 467
NNXN=NX(N)	SIMK 468
DO 201 I=1,NNXN	SIMK 469
II=II+1	SIMK 470
XDOTL(II)=0.0	SIMK 471
NNUN=NU(N)	SIMK 472
DO 201 J=1,NNUN	SIMK 473
201 XDOTL(II)=XDOTL(II)+BT(I,J,N)*UI(J,N)	SIMK 474
DO 201 J=1,NNXN	SIMK 475
200 XDOTL(II)=XDOTL(II)+AT(I,J,N)*X(J,N)	SIMK 476
251 CONTINUE	SIMK 477
C	SIMK 478
C INTERCONNECTION EQUATIONS	SIMK 479
C	SIMK 480
C INTERNAL OUTPUTS RIN=CN*XN+DN*UN	SIMK 481
C	SIMK 482
II=0	SIMK 483
DO 35 N=1,KR	SIMK 484
NNRN=NR(N)	SIMK 485
DO 30 I=1,NNRN	SIMK 486
II=II+1	SIMK 487
YL(II)=0.0	SIMK 488
NNXN=NX(N)	SIMK 489
DO 301 J=1,NNXN	SIMK 490
301 YL(II)=YL(II)+CT(I,J,N)*X(J,N)	SIMK 491
NNUN=NU(N)	SIMK 492
DO 301 J=1,NNUN	SIMK 493
300 YL(II)=YL(II)+DT(I,J,N)*UI(J,N)	SIMK 494
350 CONTINUE	SIMK 495
C	SIMK 496
C INTERNAL INPUTS	SIMK 497
C	SIMK 498
J=0	SIMK 499
DO 22 N=1,KR	SIMK 500
NNRN=NR(N)	SIMK 501
DO 22 I=1,NNRN	SIMK 502
J=J+1	SIMK 503
220 RIN(J)=RI(I,N)	SIMK 504
DO 24 I=1,KYIN	SIMK 505
II=II+1	SIMK 506
YL(II)=0.0	SIMK 507
DO 23 J=1,KYOUT	SIMK 508
230 YL(II)=YL(II)+P(I,J)*RIN(J)	SIMK 509
DO 24 J=1,NU	SIMK 510
240 YL(II)=YL(II)+Q(I,J)*U(J)	SIMK 511
C	SIMK 512
C EXTERNAL RESPONSE EQUATIONS	SIMK 513
C	SIMK 514
II=0	SIMK 515
DO 28 I=1,NR	SIMK 516
II=II+1	SIMK 517
RL(II)=0.0	SIMK 518
DO 27 J=1,KYOUT	SIMK 519
270 RL(II)=RL(II)+P(I,J)*RIN(J)	SIMK 520
DO 28 J=1,NU	SIMK 521
280 RL(II)=RL(II)+S(I,J)*U(J)	SIMK 522
RETURN	SIMK 523
END	SIMK 524

Figure 32. Subroutine SIMK Program Listing (Concluded)

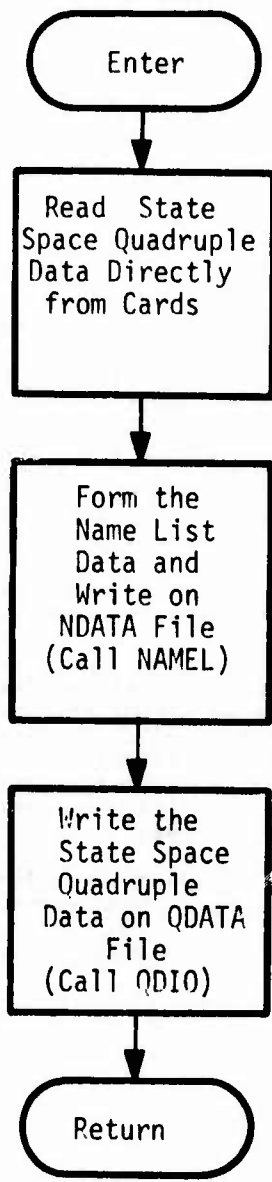


Figure 33. Subroutine QUADK Flow Chart

	IF ((CARD(1).NE.HXDOT).OR.(CARD(2).NE.HSX))GO TO 220	QUADK 65
C		QUADK 66
C	READ A MATRIX (XDOT/X)	QUADK 67
C		QUADK 68
	DECODE (4,210,CARD(4))NX,DUMMY	QUADK 69
210	FORMAT (I3,A1)	QUADK 70
	CALL INPT(A,NXM,NXM)	QUADK 71
	GO TO 100	QUADK 72
220	CONTINUE	QUADK 73
	IF ((CARD(1).NE.HXDOT).OR.(CARD(2).NE.HSU))GO TO 240	QUADK 74
C		QUADK 75
C	READ B MATRIX (XDOT/U)	QUADK 76
C		QUADK 77
	DECODE (4,210,CARD(4))NX,DUMMY	QUADK 78
	DECODE (4,230,CARD(5))DUMMY,NU	QUADK 79
230	FORMAT (A1,I3)	QUADK 80
	CALL INPT(B,NXM,NUM)	QUADK 81
	GO TO 100	QUADK 82
240	CONTINUE	QUADK 83
	IF (CARD(1).NE.HRSX)GO TO 260	QUADK 84
C		QUADK 85
C	READ C MATRIX (R/X)	QUADK 86
C		QUADK 87
	DECODE (4,210,CARD(4))NR,DUMMY	QUADK 88
	DECODE (4,230,CARD(5))DUMMY,NX	QUADK 89
230	FORMAT (A1,I3)	QUADK 90
	CALL INPT(C,NRM,NXM)	QUADK 91
	GO TO 100	QUADK 92
260	CONTINUE	QUADK 93
	IF (CARD(1).NE.HRSU)GO TO 280	QUADK 94
C		QUADK 95
C	READ D MATRIX (R/U)	QUADK 96
C		QUADK 97
	DECODE (4,210,CARD(4))NR,DUMMY	QUADK 98
	DECODE (4,230,CARD(5))DUMMY,NU	QUADK 99
	CALL INPT(D,NRM,NUM)	QUADK 100
	GO TO 100	QUADK 101
280	CONTINUE	QUADK 102
	IF (CARD(1).NE.HEND)GO TO 320	QUADK 103
C		QUADK 104
C	READ AND UPDATE NAME LIST DATA	QUADK 105
C		QUADK 106
	NFLAG=0	QUADK 107
	CALL NAMEL (NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,	QUADK 108
	IDESI,INITI,DESS,UNITSS,DESO,UNITOO,DESI,UNITII,NXX,NRR,NUU,	QUADK 109
	2NXM,NRM,NUM,NX,NR,NU,NFLAG,MB,KB,NB)	QUADK 110
C		QUADK 111
C	WRITE QUADRUPLE DATA ON FILE QDATA	QUADK 112
C		QUADK 113
	IQ=0	QUADK 114
	MFLAG=2	QUADK 115
	NXA=NX \$ NRA=NR \$ NUA=NU	QUADK 116
	CALL QDIO (A,B,C,D,DESS,NX,NR,NU,NXM,NR4,NUM,NXA,NRA,NUA,	QUADK 117
	INR1,NR2,NR3,NU1,NU2,NU3,T,IO,IPRINT,IW,JO,HEAD,MARK,	QUADK 118
	2LOCATE,NULL,INSERT,MFLAG)	QUADK 119
	RETURN	QUADK 120
C		QUADK 121
C	PRINT ERROR MESSAGE	QUADK 122
C		QUADK 123
320	CONTINUE	QUADK 124
	WRITE (IW,340)	QUADK 125
340	FORMAT (IH1,/,/,IX,37HDATA CONTROL CARD SPECIFICATION ERROR)	QUADK 126
	STOP 111	QUADK 127
	END	

Figure 34. Subroutine QUADK Program Listing (Concluded)

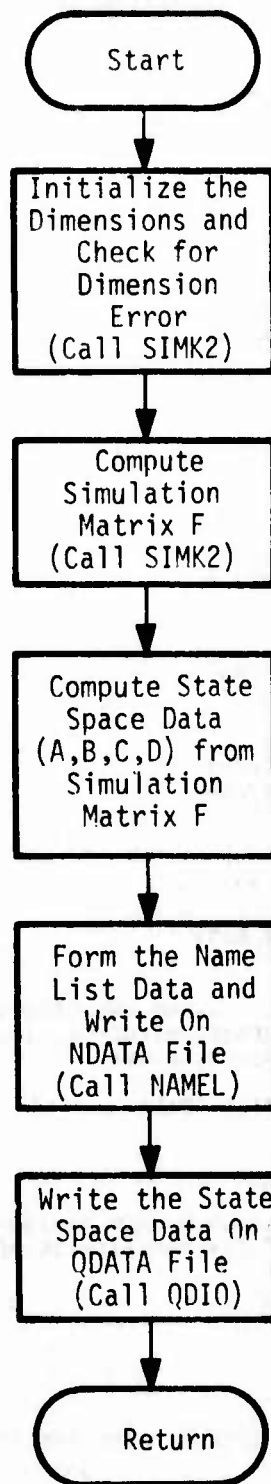


Figure 35. Subroutine STAMK4 Flow Chart


```

SUBROUTINE STAMK4 (V,W,F,U,A,R,C,D,NNS,VNS,DESS,UNITS,
INNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI,MAXN,MAXM,
2NXM,NUM,NUM,NYM,MB,MS1,MS2,MS3,MS4,NB)

C
C
C   PURPOSE - TO OBTAIN STATE SPACE MODEL FROM USER WRITTEN
C   SIMULATION EQUATION SUBROUTINE SIMK2
C   ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC
C   DATE WRITTEN - 1975

C   SUBPROGRAMS CALLED
C     DERRM
C     MPDS
C     QD10
C     TDINVR
C     DERRMS
C     NAMEL
C     SIMK2

C   ARGUMENTS LIST
C     V           V ARRAY FOR COMPUTING SIMULATION MATRIX
C     W           W ARRAY FOR COMPUTING SIMULATION MATRIX
C     F           SIMULATION MATRIX
C     U           ARRAY FOR EXTERNAL INPUTS
C     A           IN/OUT STATE TRANSITION MATRIX
C     R           IN/OUT CONTROL INPUT MATRIX
C     C           IN/OUT STATE OUTPUT MATRIX
C     D           IN/OUT CONTROL OUTPUT MATRIX
C     NNS         IN/OUT NUMBER ARRAY FOR STATE
C     VNS         IN/OUT VARIABLE NAME ARRAY FOR STATE
C     DESS        IN/OUT DESCRIPTION ARRAY FOR STATE
C     UNITS       IN/OUT UNIT ARRAY FOR STATE
C     NNO         IN/OUT NUMBER ARRAY FOR OUTPUT
C     VNO         IN/OUT VARIABLE NAME ARRAY FOR OUTPUT
C     DESO        IN/OUT DESCRIPTION ARRAY FOR OUTPUT
C     UNITO       IN/OUT UNIT ARRAY FOR OUTPUT
C     NNI         IN/OUT NUMBER ARRAY FOR INPUT
C     VNI         IN/OUT VARIABLE NAME ARRAY FOR INPUT
C     DESI        IN/OUT DESCRIPTION ARRAY FOR INPUT
C     UNITI       IN/OUT UNIT ARRAY FOR INPUT
C     MAXN        INPUT  MAXIMUM ROW DIMENSION FOR SIMULA MATRIX F
C     MAXM        INPUT  MAXIMUM COLUMN DIMENSION FOR SIMU MATRIX F
C     NXM         INPUT  MAXIMUM NUMBER OF STATES
C     NRM         INPUT  MAXIMUM NUMBER OF OUTPUTS
C     NUM         INPUT  MAXIMUM NUMBER OF INPUTS
C     NYM         INPUT  MAXIMUM DIMENSION FOR INTERCONN EQUATIONS
C     MB          INPUT  MAXIMUM NO OF SUBSYSTEMS FOR COMBINING
C     MS1         INPUT  MAXIMUM DIMENSION FOR SCRATCH ARRAY S1
C     MS2         INPUT  MAXIMUM DIMENSION FOR SCRATCH ARRAY S2
C     MS3         INPUT  MAXIMUM DIMENSION FOR SCRATCH ARRAY S3
C     MS4         INPUT  MAXIMUM DIMENSION FOR SCRATCH ARRAY S4
C     NB          INPUT  MAXIMUM SYSTEM NO - IMPLICIT MODEL

C     COMMON /INOUT/ IR, IW, IPRINT, INSERT, LOCATE, NULL, MARK(20), JN, JQ, JS
C     COMMON /SYS/  SCODE, SDES(5), MSYS, HEAD(20), NSYS(9), SHEAD(9,20)
C     I, PHEAD(20)
C     DIMENSION V (MAXN), W (MAXM), F (MAXN, MAXM)
C     DIMENSION U (NUM)
C     DIMENSION A (NXM, NXM), R (NXM, NUM), C (NRM, NXM), D (NRM, NUM)
C     DIMENSION NNS (NXM), VNS (NXM, ?), DESS (NXM, 10), UNITS (NXM, 4)
C     DIMENSION NNO (NRM), VNO (NRM, ?), DESO (NRM, 10), UNITO (NRM, 4)
C     DIMENSION NNI (NUM), VNI (NUM, ?), DESI (NUM, 10), UNITI (NUM, 4)
C     COMMON /SC1/  S1(1)
C     DIMENSION DESS (NXM, 10, MB), UNITS (NXM, 4, MB)

STAMK4 2
STAMK4 3
STAMK4 4
STAMK4 5
STAMK4 6
STAMK4 7
STAMK4 8
STAMK4 9
STAMK410
STAMK411
STAMK412
STAMK413
STAMK414
STAMK415
STAMK416
STAMK417
STAMK418
STAMK419
STAMK420
STAMK421
STAMK422
STAMK423
STAMK424
STAMK425
STAMK426
STAMK427
STAMK428
STAMK429
STAMK430
STAMK431
STAMK432
STAMK433
STAMK434
STAMK435
STAMK436
STAMK437
STAMK438
STAMK439
STAMK440
STAMK441
STAMK442
STAMK443
STAMK444
STAMK445
STAMK446
STAMK447
STAMK448
STAMK449
STAMK450
STAMK451
STAMK452
STAMK453
STAMK454
STAMK455
STAMK456
STAMK457
STAMK458
STAMK459
STAMK460
STAMK461
STAMK462
STAMK463
STAMK464

```

Figure 36. Subroutine STAMK4 Program Listing

```

C      DIMENSION DES00(NRM,1),MR1,INIT00(NRM,4,MR)          STAMK465
C      DIMENSION DES11(NUM,10,MR),INIT11(NUM,4,MR)          STAMK466
C      DIMENSION NXX(MR),NRR(MR),NUU(MR)                    STAMK467
      L1=1 $ L2=L1+NXM*MR*10 $ L3=L2+NXM*MR*4 $ L4=L3+NRM*MR*10 STAMK468
      L5=L4+NRM*MR*4 $ L6=L5+NUM*MR*10 $ L7=L6+NUM*MR*4     STAMK469
      L8=L7+MR $ L9=L8+MR $ L10=L9+MR                       STAMK470
      IF(L1.GT.MS1)                                          STAMK471
1CALL DERRM(L10,MS2,MS3,MS4,MS1,MS2,MS3,MS4,0,0,4HSTAM,4HK4  .IW) STAMK472
      NR1=0 $ NR2=0 $ NR3=0 $ NU1=0 $ NU2=0 $ NU3=0        STAMK473
      NXA=0 $ NMA=0 $ NUA=0                                STAMK474
      EPSF=1.0E-30 $ T=0.0 $ NFLAG=0                      STAMK475
C      STAMK476
C      INITIALIZING CALL TO SUBROUTINE SIMK2                 STAMK477
C      STAMK478
      INIT=                                                  STAMK479
      NX=0 $ NY=0 $ NR=0 $ NU=0                             STAMK480
      N1=1 $ N2=N1+NX $ N3=N2+NY $ N4=N3+NX                STAMK481
      CALL SIMK2(W(N1),W(N2),W(N3),W(N4),V(N1),V(N2),V(N3), STAMK482
      INX,NY,NR,NU,INIT,T)                                  STAMK483
C      STAMK484
C      CHECK FOR DIMENSION ERROR                             STAMK485
C      STAMK486
      INIT = 1                                              STAMK487
      M=2+NX+NY+NU                                          STAMK488
      N=NX+NY+NR                                             STAMK489
      IF((NX.GT.NXM).OR.(NR.GT.NRM).OR.(NU.GT.NUM).OR.(NY.GT.NYM) STAMK490
1CALL DERRMS(INX,NR,N1,NY,NXM,NRM,NUM,NYM,1,0,4HSTAM,4HK1  .IW) STAMK491
      N1=1 $ N2=N1+NX $ N3=N2+NY $ N4=N3+NX                STAMK492
      DO 101 J=1,M                                          STAMK493
191 W(J)=.0.                                                STAMK494
      DO 501 J=1,M                                          STAMK495
      W(J)=1.                                               STAMK496
      CALL SIMK2(W(N1),W(N2),W(N3),W(N4),V(N1),V(N2),V(N3), STAMK497
      INX,NY,NR,NU,INIT,T)                                  STAMK498
      W(J)=.0.                                              STAMK499
      DO 501 I=1,N                                          STAMK100
501 F(I,J)=V(I)                                             STAMK101
C      STAMK102
C      COMPUTE THE SIMULATION MATRIX                         STAMK103
C      STAMK104
      NV=NX+NY                                              STAMK105
      IF(IP=INT.EQ.6)CALL MPRS(F,MAXN,MAXM,V,M,T,4HSTM 1)  STAMK106
      DO 51 I=1,NV                                          STAMK107
      DO 52 J=1,NV                                          STAMK108
52 F(I,J)=-F(I,J)                                          STAMK109
51 F(I,I)=F(I,I)+1.                                         STAMK110
      CALL TDINVR(ISOL,INSOL,NV,4,F,MAXN,KDUM,DET)         STAMK111
      IB=NV+1                                               STAMK112
      IE=NV+NR                                              STAMK113
      JB=IB                                                  STAMK114
      JE=M                                                  STAMK115
      DO 53 I=IB,IE                                         STAMK116
      DO 53 J=JB,JE                                         STAMK117
      DO 53 K=1,NV                                          STAMK118
53 F(I,J)=F(I,J)+F(I,K)*F(K,J)                            STAMK119
      DO 53 I=IB,IE                                         STAMK120
      DO 53 J=1,JE                                         STAMK121
      IF(ABS(F(I,J)).LE.EPSF) F(I,J) = 0.0                STAMK122
530 CONTINUE                                               STAMK123
      IF(IP=INT.EQ.6)CALL MPRS(F,MAXN,MAXM,V,4,T,4HSTM1)  STAMK124
C      STAMK125
C      FORM A,B,C,D MATRICES                                STAMK126
C      STAMK127
      J1=NV+1                                               STAMK128
      J2=NV+NX                                              STAMK129
      J3=J1+NX                                              STAMK130

```

Figure 36. Subroutine STAMK4 Program Listing (Continued)

	J4=J2*NU	STAMK131
	I1=NV*J	STAMK132
	I2=NV*NR	STAMK133
	DO 6001 I=1,NX	STAMK134
	DO 6001 J=J1,J2	STAMK135
	JJ=J-J1*J	STAMK136
6001	A(I,J)=F(I,J)	STAMK137
	DO 6002 I=1,NX	STAMK138
	DO 6002 J=J3,J4	STAMK139
	JJ=J-J3*J	STAMK140
6002	B(I,J)=F(I,J)	STAMK141
	DO 6003 I=1,I2	STAMK142
	II=I-II*J	STAMK143
	DO 6003 J=J1,J2	STAMK144
	JJ=J-J1*J	STAMK145
6003	C(II,JJ)=F(I,J)	STAMK146
	DO 6004 I=1,I2	STAMK147
	II=I-II*J	STAMK148
	DO 6004 J=J3,J4	STAMK149
	JJ=J-J3*J	STAMK150
6004	D(II,JJ)=F(I,J)	STAMK151
C		STAMK152
C	READ AND UPDATE NAME LIST DATA	STAMK153
C		STAMK154
	KB=NMAX	STAMK155
	CALL NAMEL(INNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,VNI,VNI,	STAMK156
	1DESI,INITI,S1(L1),S1(L2),S1(L3),S1(L4),S1(L5),S1(L6),	STAMK157
	2S1(L7),S1(L8),S1(L9),NXM,NRM,NUM,NX,NR,NU,NFLAG,MB,KB,NB)	STAMK158
C		STAMK159
C	WRITE QUADRUPLE DATA ON FILE QDATA	STAMK160
C		STAMK161
	IQ=0	STAMK162
	MFLAG=?	STAMK163
	NXA=NX \$ NRA=NR \$ NUA=NU	STAMK164
	CALL QDIO(A,B,C,D,A,NX,NR,NI,NXM,NRM,NUM,NXA,NPA,NUA,	STAMK165
	INR1,NR2,NR3,NU),NU2,NU3,T,IQ,IPRINT,IW,JG,HEAD,MARK,	STAMK166
	2LOCATE,NULL,INSERT,MFLAG)	STAMK167
	RETURN	STAMK168
	END	STAMK169

Figure 36. Subroutine STAMK4 Program Listing (Concluded)

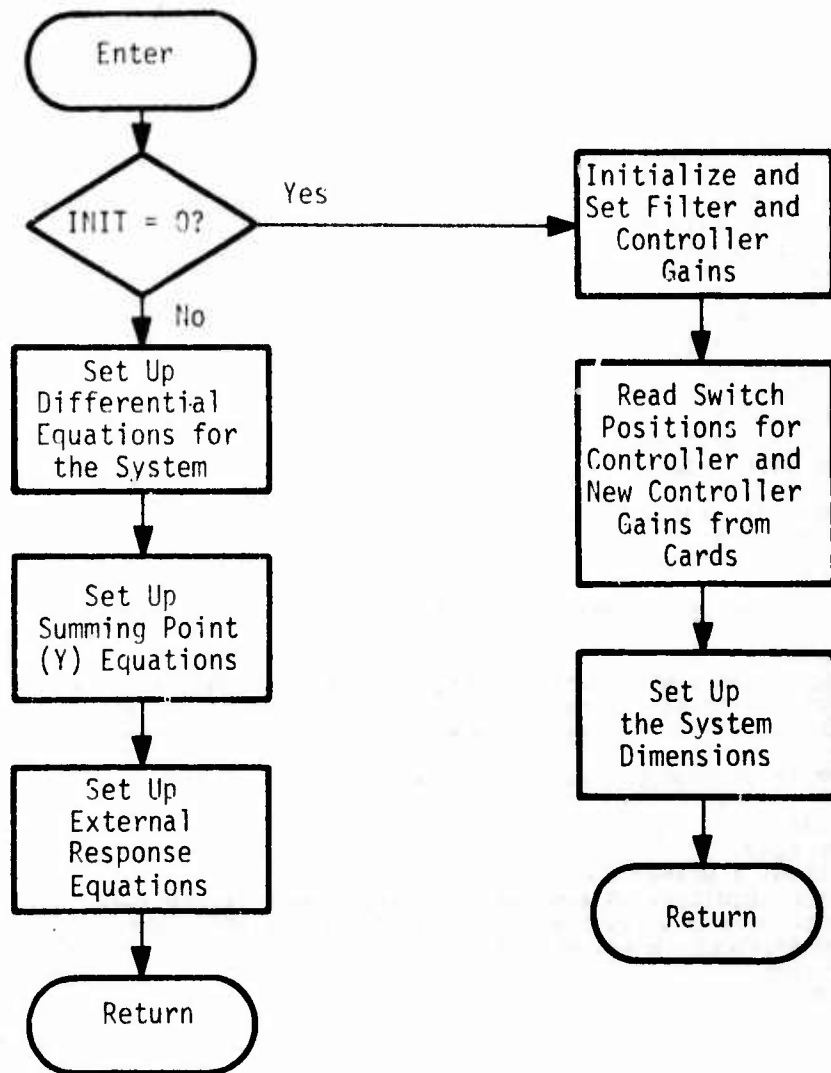


Figure 37. Subroutine SIMK2 Flow Chart

```

SUBROUTINE SIMK2(XDOT,Y,X,U,XDOTL,YL,RL,NX,NY,NR,NU,INIT,T)
PURPOSE - TO IMPLEMENT SIMULATION EQUATIONS FOR CSA CONTROLLER
ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
DATE WRITTEN - 1975

ARGUMENTS LIST
XDOT      ARRAY FOR STATE DERIVATIVES
Y         ARRAY FOR Y EQUATIONS
X         ARRAY FOR STATES
U         ARRAY FOR EXTERNAL INPUTS
XDOTL     OUTPUT    ARRAY FOR DERIVATIVE OF STATE
YL        OUTPUT    ARRAY FOR Y EQUATION VARIABLES
RL        OUTPUT    ARRAY FOR EXTERNAL RESPONSE VARIABLES
NX        OUTPUT    NUMBER OF STATES
NY        OUTPUT    NUMBER OF Y EQUATIONS
NR        OUTPUT    NUMBER OF OUTPTS
NU        OUTPUT    NUMBER OF INPUTS
INIT      INPUT     INITIAL MODE FLAG
T         OUTPUT    SAMPLE TIME

DIMENSION XDOT(NX),Y(NY),X(NX),U(NU),XDOTL(NX),YL(NY),RL(NR)
COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
DIMENSION CARD(20)
REAL KM1,KM2,KAF,KQ,KP,KNF,MLC1,MLC2
DATA HENDB,HWITC,HAINB/4HEND ,4HWITC,4HAIN /
DATA HMLC1,HMLC2,HSASB,HALDC/4HMLC1,4HMLC2,4HSAS ,4HALDC/
DATA HKM1B,HKM2B,HKAFB,HKQBR/4HKM1 ,4HKM2 ,4HKAF ,4HKQ /
DATA HKPBB,HKNFB/4HKP ,4HKNF /

CHECK IF INITIALIZATION MODE
IF(INIT.NE.0) GO TO 100

SET FILTER GAINS
AP=-.1 $ BP=.22361E-03
ANF=-4.0
AF=-.02
AM1=-.01
AM2=-.01
AMF=-1. $ BHF=-1.
ATF=-4.0 $ BTF=4.0

SET CONTROLLER SWITCHES
SAS=0.0 $ ALDCS=0.0 $ MLC1=0.0 $ MLC2=0.0

SET CONTROLLER GAINS
KM1=1.0/0.26
KM2=1.0/0.05591
KAF=36.0*0.26
KQ=0.5
KP=0.7062
KNF=-0.09

READ CONTROLLER SWITCHES ON AND CONTROLLER GAIN VALUES
10 CONTINUE
READ(IR,20)CARD
20 FORMAT(20A4)
IF(CARD(1).EQ.HENDR)GO TO 80

```

```

SIMK2 2
SIMK2 3
SIMK2 4
SIMK2 5
SIMK2 6
SIMK2 7
SIMK2 8
SIMK2 9
SIMK2 10
SIMK2 11
SIMK2 12
SIMK2 13
SIMK2 14
SIMK2 15
SIMK2 16
SIMK2 17
SIMK2 18
SIMK2 19
SIMK2 20
SIMK2 21
SIMK2 22
SIMK2 23
SIMK2 24
SIMK2 25
SIMK2 26
SIMK2 27
SIMK2 28
SIMK2 29
SIMK2 30
SIMK2 31
SIMK2 32
SIMK2 33
SIMK2 34
SIMK2 35
SIMK2 36
SIMK2 37
SIMK2 38
SIMK2 39
SIMK2 40
SIMK2 41
SIMK2 42
SIMK2 43
SIMK2 44
SIMK2 45
SIMK2 46
SIMK2 47
SIMK2 48
SIMK2 49
SIMK2 50
SIMK2 51
SIMK2 52
SIMK2 53
SIMK2 54
SIMK2 55
SIMK2 56
SIMK2 57
SIMK2 58
SIMK2 59
SIMK2 60
SIMK2 61
SIMK2 62
SIMK2 63
SIMK2 64

```

Figure 38. Subroutine SIMK2 Program Listing

	IF(CARD(4).NE.HWITC)GO TO 40	SIMK2 65
C		SIMK2 66
C	READ CONTROLLER SWITCHES ON	SIMK2 67
C		SIMK2 68
30	CONTINUE	SIMK2 69
	READ(IR,20)CARD	SIMK2 70
	IF(CARD(1).EQ.HENDR)GO TO 10	SIMK2 71
	IF(CARD(1).EQ.HMLC1)MLC1=1.0	SIMK2 72
	IF(CARD(1).EQ.HMLC1)GO TO 30	SIMK2 73
	IF(CARD(1).EQ.HMLC2)MLC2=1.0	SIMK2 74
	IF(CARD(1).EQ.HMLC2)GO TO 30	SIMK2 75
	IF(CARD(1).EQ.HSASR)SAS=1.0	SIMK2 76
	IF(CARD(1).EQ.HSASR)GO TO 30	SIMK2 77
	IF(CARD(1).EQ.HALDC)ALDCS=1.0	SIMK2 78
	IF(CARD(1).EQ.HALDC)GO TO 30	SIMK2 79
	STOP 111	SIMK2 80
C		SIMK2 81
C	READ CONTROLLER GAIN VALUES	SIMK2 82
C		SIMK2 83
40	CONTINUE	SIMK2 84
	IF(CARD(4).NE.HAINR)STOP 111	SIMK2 85
50	CONTINUE	SIMK2 86
	READ(IR,20)CARD	SIMK2 87
	IF(CARD(1).EQ.HENDR)GO TO 10	SIMK2 88
	IF(CARD(1).EQ.HKM1R)READ(IR,60)KM1	SIMK2 89
60	FORMAT(E12.6)	SIMK2 90
	IF(CARD(1).EQ.HKM1R)GO TO 50	SIMK2 91
	IF(CARD(1).EQ.HKM2R)READ(IR,60)KM2	SIMK2 92
	IF(CARD(1).EQ.HKM2R)GO TO 50	SIMK2 93
	IF(CARD(1).EQ.HKAFR)READ(IR,60)KAF	SIMK2 94
	IF(CARD(1).EQ.HKAFR)GO TO 50	SIMK2 95
	IF(CARD(1).EQ.HKQBR)READ(IR,60)KQ	SIMK2 96
	IF(CARD(1).EQ.HKQBR)GO TO 50	SIMK2 97
	IF(CARD(1).EQ.HKPBFR)READ(IR,60)KPF	SIMK2 98
	IF(CARD(1).EQ.HKPBFR)GO TO 50	SIMK2 99
	IF(CARD(1).EQ.HKNFR)READ(IR,60)KNF	SIMK2100
	IF(CARD(1).EQ.HKNFR)GO TO 50	SIMK2101
	STOP 111	SIMK2102
80	CONTINUE	SIMK2103
C		SIMK2104
C	SET DIMENSIONS OF SYSTEM	SIMK2105
C		SIMK2106
C	NX=7 \$ NR=3 \$ NU=9 \$ NY=5	SIMK2107
C		SIMK2108
C	RETURN	SIMK2109
C		SIMK2110
C	RETURN	SIMK2111
C		SIMK2112
C	SIMULATION EQUATIONS	SIMK2113
C		SIMK2114
100	CONTINUE	SIMK2115
C		SIMK2116
C	DIFFERENTIAL EQUATIONS	SIMK2117
C		SIMK2118
	XDOTL(1)=AP*X(1)+BP*U(3)	SIMK2119
	XDOTL(2)=ANF*X(2)+ALDCS*U(6)	SIMK2120
	XDOTL(3)=AM1*X(3)+MLC1*Y(2)	SIMK2121
	XDOTL(4)=AF*X(4)+ALDCS*Y(1)	SIMK2122
	XDOTL(5)=ATF*X(5)+RTF*Y(3)	SIMK2123
	XDOTL(6)=AMF*X(6)+RMF*U(2)	SIMK2124
	XDOTL(7)=AM2*X(7)+MLC2*Y(4)	SIMK2125
C		SIMK2126
C	SUMMING POINT EQUATIONS	SIMK2127
C		SIMK2128
	YL(1)=KAF*X(2)+ANF*U(4)	SIMK2129
	YL(2)=KN1*U(4)-U(9)	SIMK2130

Figure 38. Subroutine SIMK2 Program Listing (Continued)

	YL (3)=KP*X (1)*X (6)+U (2)*KNF*U (7)	SIMK2131
	YL (4)=KN2*U (5)-U (9)	SIMK2132
	YL (5)=ALDCS*X (5)+X (1)*SAS*KQ*U (8)	SIMK2133
C		SIMK2134
C	RESPONSE EQUATIONS	SIMK2135
C		SIMK2136
	RL (1)=U (1)	SIMK2137
	RL (2)=Y (5)	SIMK2138
	RL (3)=X (1)	SIMK2139
C		SIMK2140
C	RETURN	SIMK2141
C		SIMK2142
	RETURN	SIMK2143
	END	SIMK2144

Figure 38. Subroutine SIMK2 Program Listing (Concluded)

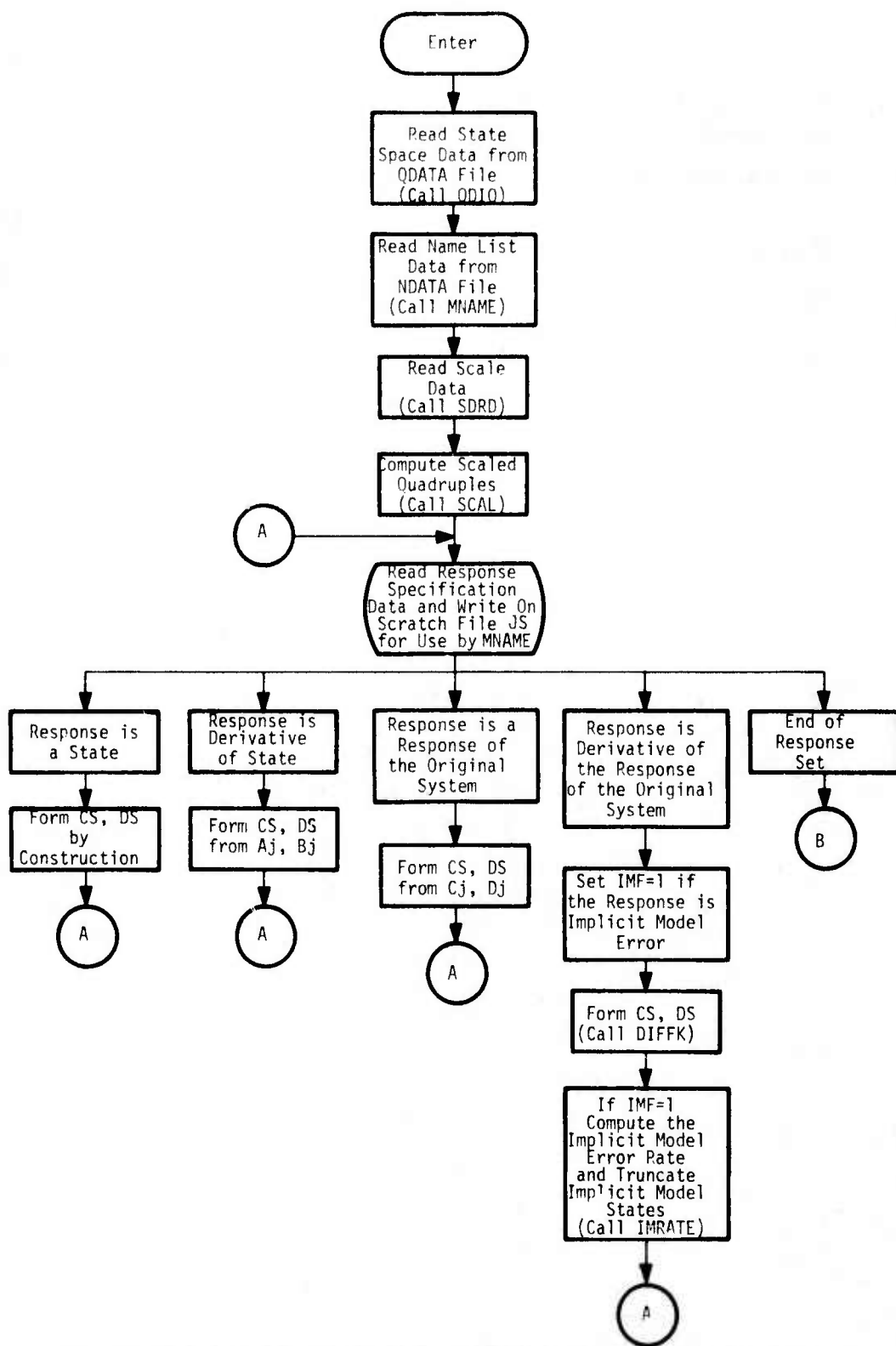


Figure 39. Subroutine CONDK Flow Chart

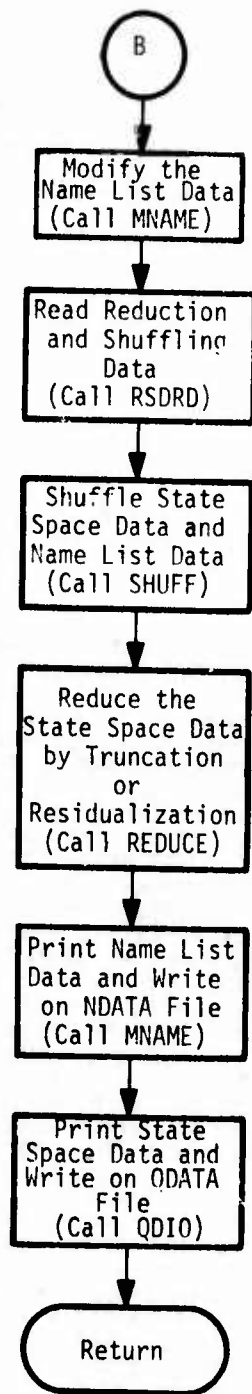


Figure 39. Subroutine CONDK Flow Chart (Concluded)


```

C      NSHUFFS   INPUT   SHUFFLING ARRAY FOR STATE           CONDK 65
C      NSHUFFO   INPUT   SHUFFLING ARRAY FOR OUTPUT          CONDK 66
C      NSHUFFI   INPUT   SHUFFLING ARRAY FOR INPUT           CONDK 67
C      CS        IN/OUT   SPECIFIED STATE OUTPUT MATRIX       CONDK 68
C      DS        IN/OUT   SPECIFIED CONTROL OUTPUT MATRIX     CONDK 69
C      CW        IN/OUT   IMPLICIT MODEL STATE OUTPUT MATRIX  CONDK 70
C      DW        IN/OUT   IMPLICIT MODEL CONTROL OUTPUT MATRIX CONDK 71
C      IRS       IN/OUT   ARRAY FOR DERIVATIVES OF RESPONSES  CONDK 72
C      Q         IN/OUT   QUADRATIC WEIGHT MATRIX             CONDK 73
C      NXM       INPUT   MAXIMUM NUMBER OF STATES             CONDK 74
C      NRM       INPUT   MAXIMUM NUMBER OF OUTPUTS            CONDK 75
C      NU        INPUT   MAXIMUM NUMBER OF INPUTS             CONDK 76
C      NDM11     INPUT   MAX ROW DIMENSION FOR SCRATCH ARRAY DUMMY1 CONDK 77
C      NDM12     INPUT   MAX COL DIMENSION FOR SCRATCH ARRAY DUMMY1 CONDK 78
C      NDM21     INPUT   MAX ROW DIMENSION FOR SCRATCH ARRAY DUMMY2 CONDK 79
C      NDM22     INPUT   MAX COL DIMENSION FOR SCRATCH ARRAY DUMMY2 CONDK 80
C      ND122     INPUT   MAX COL DIMENSION FOR SCRATCH ARRAY DUMMY2 CONDK 81
C      COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)
C      1,PHEAD(20)
C      DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
C      DIMENSION CM(NRM,NXM),DM(NRM,NUM)
C      DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
C      DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
C      DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
C      DIMENSION NNNS(NXM),VNNS(NXM,2),DESNS(NXM,10),UNITNS(NXM,4)
C      DIMENSION NNNO(NRM),VNNO(NRM,2),DESN0(NRM,10),UNITNO(NRM,4)
C      DIMENSION NNNI(NUM),VNNI(NUM,2),DESNI(NUM,10),UNITNI(NUM,4)
C      DIMENSION DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22),DUMMY3(NUM)
C      DIMENSION ES(NXM,NUM),EP(NRM,NUM)
C      DIMENSION NSHUFFS(NXM),NSHUFFO(NRM),NSHUFFI(NUM)
C      DIMENSION CS(NRM,NXM),DS(NRM,NUM),CW(NRM,NXM),DW(NRM,NUM)
C      DIMENSION IRS(NRM),Q(NRM,NRM)
C      DIMENSION CARD(20)
C      EQUIVALENCE (NU1,NCL),(NU2,NGT),(NU3,VCD),(NR1,NDR),(NR2,NPR),
C      1(NR3,ISR)
C      DATA HENDB,HRBBB,HRDOT,HXDOT/4HEND,4HR,4HRDOT,4HXDOT/
C      DATA HXBBB,HSCAL/4HX,4HSCAL/
C      DATA HCONS,HSELE/4HCONS,4HSFLE/
C      DATA HSIGN,HRFOR,HFNSO/4HSIGN,4HRFOR,4HFNSO/
C      DATA HONTR,HUSTB,HOMMA/4HONTR,4HUST,4HOMMA/
C      READ QUADRUPLE DATA
C      IQ=0
C      NFLAG=1
C      CALL QDIO(A,B,C,D,Q,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA,
C      1NR1,NR2,NR3,NU1,NU2,NU3,T,IQ,IPRINT,IW,JQ,PHEAD,MARK,
C      2LOCATE,NULL,INSERT,NFLAG)
C      IF(IPRINT.EQ.6)CALL DERUG(1,4HRESP,4HK,5.0,IW)
C      READ NAME LIST DATA
C      MFLAG=1
C      CALL MNAME(INNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,
C      1NNI,VNI,DESI,UNITI,NNNS,VNNS,DESNS,UNITNS,NNNO,VNNO,DESN0,
C      2UNITNO,NNNI,VNNI,DESNI,UNITNI,NX,NR,NU,NXM,NRM,NUM,
C      3NU1,NU2,NU3,NR1,NR2,NR3,MFLAG)
C      IF(IPRINT.EQ.6)CALL DERUG(2,4HRESP,4HK,5.0,IW)
C      REWIND JS
C      READ(IR,140)CARD
C      IF(CARD(1).EQ.HENDR)GO TO 110
C      IF(CARD(1).NE.HSCAL)CALL ERDM(1,4HRESP,4HK,5.0,IW)
C      READ SCALE AND NEW UNIT DATA
C      CONDK130

```

Figure 40. Subroutine CONDK Program Listing (Continued)

	CALL SDRD(DUMMY1,UNITNS,UNITS,DUMMY2,UNITNO,UNITO,DUMMY3,UNITNI, UNITI,NX,NR,NU,NXM,NRM,NUM,[R,[W,[PRINT)	CONDK131
C		CONDK132
C	COMPUTE SCALED QUADRUPLES	CONDK133
C		CONDK134
	CALL SCAL(A,B,C,D,DUMMY1,DUMMY2,DUMMY3,NX,NR,NU,NXM,NRM,NUM)	CONDK135
C		CONDK136
C	READ RESPONSE SPECIFICATION DATA AND WRITE IT	CONDK137
C	ON A SCRATCH FILE JS FOR USE BY SUBROUTINE MNAME	CONDK138
C		CONDK139
C		CONDK140
110	CONTINUE	CONDK141
	NXO=NX \$ NRO=NR	CONDK142
	IFLAG=0	CONDK143
	IRR=0 \$ K=0 \$ NDR=0 \$ NPR=0 \$ NSR=0	CONDK144
	JU=0 \$ NCL=0 \$ NGT=0 \$ NCD=0	CONDK145
	IRES=0	CONDK146
120	CONTINUE	CONDK147
	READ([R,140)CARD	CONDK148
140	FORMAT(20A4)	CONDK149
	WRITE(JS,140)CARD	CONDK150
	IF(CARD(1).EQ.HENDR)GO TO 570	CONDK151
	IRES=1	CONDK152
	IF((CARD(1).NE.HCONS).AND.(CARD(1).NE.HSELE))	CONDK153
	CALL FERRM(2,4HRESP,4HX ,5,0,IW)	CONDK154
	IF((CARD(3).NE.HONTR).AND.(CARD(3).NE.HUSTB).AND.(CARD(3).NE. HOMMA))GO TO 240	CONDK155
C		CONDK156
C	READ INPUT SPECIFICATION AND MODIFY R AND D MATRICES	CONDK157
C		CONDK158
	IF(CARD(3).EQ.HONTR)IUU=1	CONDK159
	IF(CARD(3).EQ.HUSTB)IUU=2	CONDK160
	IF(CARD(3).EQ.HOMMA)IUU=3	CONDK161
160	CONTINUE	CONDK162
	READ([R,140)CARD	CONDK163
	WRITE(JS,140)CARD	CONDK164
	IF(CARD(1).EQ.HENDR)GO TO 220	CONDK165
	JU=JU+1	CONDK166
	DECODE(4,340,CARD(2))D1,K,D2	CONDK167
	DO 180 I=1,NX	CONDK168
180	DUMMY1(I,JU)=B(I,K)	CONDK169
	DO 200 I=1,NR	CONDK170
200	DUMMY2(I,JU)=D(I,K)	CONDK171
	GO TO 160	CONDK172
220	CONTINUE	CONDK173
	IF(IUU.EQ.1)NCL=JU	CONDK174
	IF(IUU.EQ.2)NGT=JU-NCL	CONDK175
	IF(IUU.EQ.3)NCD=JU-NCL-NGT	CONDK176
	GO TO 120	CONDK177
240	CONTINUE	CONDK178
	IF(IFLAG.EQ.1)GO TO 300	CONDK179
	NUN=NCL+NGT+NCD	CONDK180
	IF(NUN.EQ.0)GO TO 300	CONDK181
	NU=NUN	CONDK182
	DO 260 I=1,NX	CONDK183
	DO 260 J=1,NU	CONDK184
260	B(I,J)=DUMMY1(I,J)	CONDK185
	DO 280 I=1,NR	CONDK186
	DO 280 J=1,NU	CONDK187
280	D(I,J)=DUMMY2(I,J)	CONDK188
	IFLAG=1	CONDK189
C		CONDK190
C	READ OUTPUT SPECIFICATION AND COMPUTE C AND D MATRICES	CONDK191
C		CONDK192
300	CONTINUE	CONDK193
	IF(CARD(4).EQ.HSIGN)IRR=1	CONDK194
	IF(CARD(4).EQ.HRFOR)IRR=2	CONDK195
		CONDK196

Figure 40. Subroutine CONDK Program Listing (Continued)

```

IF(CARD(3).EQ.HENSO)IRR=3
IM=0 * IMF=0 $ J=0
320 CONTINUE
READ(IR,140)CARD
WRITE(JS,140)CARD
IF(CARD(1).EQ.HENDP)GO TO 550
J=J+1
DECODE(4,340,CARD(2))D1,K,D2
340 FORMAT(A1,I2,A1)
ID=0 * IX=0
IF(CARD(1).EQ.HRRRR)GO TO 360
ID=0 * IX=1
IF(CARD(1).EQ.HXBHR)GO TO 360
ID=1 * IX=0
IF(CARD(1).EQ.HRDOT)GO TO 360
ID=1 * IX=1
IF(CARD(1).EQ.HXDOT)GO TO 360
CALL FPRM(3,4HRESP,4HK ,5,0,IW)
360 CONTINUE
IF(IPRINT.EQ.6)CALL DERUG(3,4HRESP,4HK ,5,0,IW)
IF(ID.EQ.1)GO TO 440
IF(IX.EQ.1)GO TO 420
DO 38 L=1,NX
380 CS(J,L)=C(K,L)
DO 40 L=1,NU
400 DS(J,L)=D(K,L)
GO TO 320
420 CONTINUE
DO 42 L=1,NX
425 CS(J,L)=0.0
CS(J,K)=1.0
DO 43 L=1,NU
430 DS(J,L)=0.0
GO TO 320
440 CONTINUE
IF(IX.EQ.1)GO TO 560
NRS=1
IRS(1)=K
IF(K.LE.NRA)GO TO 450
C
C DEFINE FLAGS AND RESPONSES FOR IMPLICIT MODEL OPERATION
C
IMF=1
IM=IM+1
DO 44 L=1,NX
443 CW(IM,L)=C(K,L)
DO 44 L=1,NU
446 DW(IM,L)=D(K,L)
450 CONTINUE
IF(IPRINT.EQ.6)CALL DERUG(4,4HRESP,4HK ,5,0,IW)
C
C COMPUTE DERIVATIVES OF RESPONSES
C
CALL DIFFK(A,R,C,D,DUMMY1,DUMMY2,NX,NR,NU,
INXM,NPM,NUM,NRS,IRS,ID,IW,IPRINT,NDM11,NDM12,NDM21,NDM22)
DO 46 L=1,NX
460 CS(J,L)=C(NR,L)
DO 48 L=1,NU
480 DS(J,L)=D(NR,L)
NR=NR-1
GO TO 320
500 CONTINUE
DO 52 L=1,NX
520 CS(J,L)=A(K,L)
DO 54 L=1,NU
540 DS(J,L)=B(K,L)

```

```

CONDK197
CONDK198
CONDK199
CONDK200
CONDK201
CONDK202
CONDK203
CONDK204
CONDK205
CONDK206
CONDK207
CONDK208
CONDK209
CONDK210
CONDK211
CONDK212
CONDK213
CONDK214
CONDK215
CONDK216
CONDK217
CONDK218
CONDK219
CONDK220
CONDK221
CONDK222
CONDK223
CONDK224
CONDK225
CONDK226
CONDK227
CONDK228
CONDK229
CONDK230
CONDK231
CONDK232
CONDK233
CONDK234
CONDK235
CONDK236
CONDK237
CONDK238
CONDK239
CONDK240
CONDK241
CONDK242
CONDK243
CONDK244
CONDK245
CONDK246
CONDK247
CONDK248
CONDK249
CONDK250
CONDK251
CONDK252
CONDK253
CONDK254
CONDK255
CONDK256
CONDK257
CONDK258
CONDK259
CONDK260
CONDK261
CONDK262

```

Figure 40. Subroutine CONDK Program Listing (Continued)

	GO TO 320	CONDK263
550	CONTINUE	CONDK264
	NR=J	CONDK265
	IF (IMF.EQ.0) GO TO 560	CONDK266
C		CONDK267
C	COMPUTE IMPLICIT MODEL ERROR RATES AND TRUNCATE THE	CONDK268
C	IMPLICIT MODEL	CONDK269
C		CONDK270
	NXR=NX-NXA	CONDK271
	IF (IM.NE.NXR) CALL FRRM(4.4HRESP,4HK .5,0,IW)	CONDK272
	IF (IPRINT.EQ.6) CALL DEBUG(5.4HRESP,4HK .5,0,IW)	CONDK273
	CALL IMRATE(CS,DS,CW,OW,DUMMY),DUMMY?,NX,NR,NU,	CONDK274
	INXM,NRM,NUM,NXA,NFA,NUA,IW,IPRINT,NDM1,NDM2,NDM3,NDM22)	CONDK275
560	CONTINUE	CONDK276
C		CONDK277
C	COMPUTE NEW C AND D MATRICES (OUTPUTS)	CONDK278
C		CONDK279
	NXN=NXA	CONDK280
	K=0	CONDK281
	IF (IRP.EQ.0) NNR=NR	CONDK282
	IF (IRP.EQ.1) NDR=NR	CONDK283
	IF (IRP.EQ.2) NPR=NR	CONDK284
	IF (IRP.EQ.2) K=NDR	CONDK285
	IF (IRR.EQ.3) NSR=NR	CONDK286
	IF (IRP.EQ.3) K=NDR+NPR	CONDK287
	DO 564 I=1,NR	CONDK288
	IK=I+K	CONDK289
	DO 562 J=1,NX	CONDK290
562	CM(IK,J)=CS(I,J)	CONDK291
	DO 564 J=1,NU	CONDK292
564	DM(IK,J)=DS(I,J)	CONDK293
	NX=NXN \$ NR=NRO	CONDK294
	GO TO 120	CONDK295
570	CONTINUE	CONDK296
	IF (IRFSP.EQ.0) GO TO 595	CONDK297
	IF (IPRINT.EQ.6) CALL DEBUG(6.4HRESP,4HK .5,0,IW)	CONDK298
	IF (IRP.EQ.0) NR=NNR	CONDK299
	IF (IRP.EQ.0) GO TO 575	CONDK300
	NX=NXN \$ NR=NDR+NPR+NSR	CONDK301
575	CONTINUE	CONDK302
	DO 590 I=1,NR	CONDK303
	DO 580 J=1,NX	CONDK304
580	C(I,J)=CM(I,J)	CONDK305
	DO 590 J=1,NU	CONDK306
590	D(I,J)=DM(I,J)	CONDK307
C		CONDK308
C	MODIFY NAME LIST DATA	CONDK309
C		CONDK310
	MFLAG=?	CONDK311
	CALL MNAME(INNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,	CONDK312
	INNI,VJI,DESI,UNITI,NNNS,VNNS,DESNS,UNITNS,NNNO,VNNO,DESNO,	CONDK313
	ZUNITNO,NNNI,VNNI,DESNi,UNITVI,NX,NR,NU,NXM,NRM,NUM,	CONDK314
	3?NU1,NU2,NU3,NR1,NR2,NR3,MFLAG)	CONDK315
	IF (IPRINT.EQ.6) CALL DEBUG(7.4HRESP,4HK .5,0,IW)	CONDK316
595	CONTINUE	CONDK317
C		CONDK318
C	READ SHUFFLING AND REDUCTION DATA	CONDK319
C		CONDK320
	CALL RSDRD(DUMMY),NSHUFS,DUMMY2,NSHUFO,DUMMY3,NSHUFI,	CONDK321
	INX,NR,NU,NXR,NXN,NXR,NRN,NRR,NRT,NUX,NXM,NRM,NUM,	CONDK322
	2IR,IW,IPRINT,IRSS)	CONDK323
	IF (IRSS.EQ.0) GO TO 600	CONDK324
	IF (IPRINT.EQ.6) CALL DEBUG(8.4HRESP,4HK .5,0,IW)	CONDK325
	IF (IPRINT.LT.6) GO TO 598	CONDK326
	WRITE (IW,596) NX,NR,NU,NXM,NRM,NUM,NDM	CONDK327
	WRITE (IW,596) NSHUFS,NSHUFO,NSHUFI	CONDK328

Figure 40. Subroutine CONDK Program Listing (Continued)

596	FORMAT(1X,(201Z,1X))	CONDK329
598	CONTINUE	CONDK330
C		CONDK331
C	SHUFFLE QUADRUPLE DATA AND NAME LIST DATA	CONDK332
C		CONDK333
	CALL SHUFF(A,B,C,D,NNS,VNS,DESNS,UNITNS,NNO,	CONDK334
	1VNO,DESNO,UNITNO,NNI,VNI,DESNI,UNITNI,	CONDK335
	2NSHUF5,NSHUF0,NSHUF1,DUMMY1,DUMMY2,NX,NR,NU,NXM,NRM,NUM,	CONDK336
	3NDM11,NDM12,NDM21,NDM22)	CONDK337
C		CONDK338
C	TRUNCATE THE SYSTEM VARIABLES AS SPECIFIED	CONDK339
		CONDK340
	NX=NXRN	CONDK341
	NR=NRN	CONDK342
	IF(NRN.EQ.0)NR=NRT,NRR	CONDK343
	NU=NUN	CONDK344
	IF(NX4.LE.0)GO TO 600	CONDK345
C		CONDK346
C	REDUCE THE QUADRUPLE DATA	CONDK347
C		CONDK348
	IF(IPRINT.EQ.5)CALL HPR(HEAD,IW)	CONDK349
	CALL REDUCE(A,B,C,D,DUMMY1,DUMMY2,ES,EP,	CONDK350
	1NX,NR,NU,NXR,NRR,NRT,NXM,NRM,NUM,T,IW,IPRINT,	CONDK351
	2NDM11,NDM12,NDM21,NDM22)	CONDK352
	IF(IPRINT.EQ.6)CALL DEBUG(9,4HRESP,4HK .5.0,IW)	CONDK353
C		CONDK354
C	WRITE NAME LIST DATA	CONDK355
C		CONDK356
	600 CONTINUE	CONDK357
	MFLAG=3	CONDK358
	CALL MNAME(NNS,VNS,DESS,UNITNS,NNO,VNO,DESO,UNITO,	CONDK359
	1NNI,VNI,DESI,UNITI,NNNS,VNNS,DESNS,UNITNS,NNNO,VNNO,DESNO,	CONDK360
	2UNITNO,NNNI,VNNI,DESNI,UNITNI,NX,NR,NU,NXM,NRM,NUM,	CONDK361
	3NU1,NU2,NU3,NR1,NR2,NR3,MFLAG)	CONDK362
	IF(IPRINT.EQ.6)CALL DEBUG(19,4HRESP,4HK .5.0,IW)	CONDK363
C		CONDK364
C	WRITE QUADRUPLE DATA	CONDK365
C		CONDK366
	IQ=0 \$ MFLAG=2	CONDK367
	NXA=NX \$ NRA=NR \$ NUA=NU	CONDK368
	CALL QDIO(A,B,C,D,Q,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA,	CONDK369
	1NR1,NR2,NR3,NU1,NU2,NU3,T,IQ,IPRINT,1d,JQ,HEAD,MARK,	CONDK370
	2LOCATF,NULL,INSERT,MFLAG)	CONDK371
	IF(IPRINT.EQ.6)CALL DEBUG(11,4HRESP,4HK .5.0,IW)	CONDK372
	RETURN	CONDK373
	END	CONDK374

Figure 40. Subroutine CONDK Program Listing (Concluded)

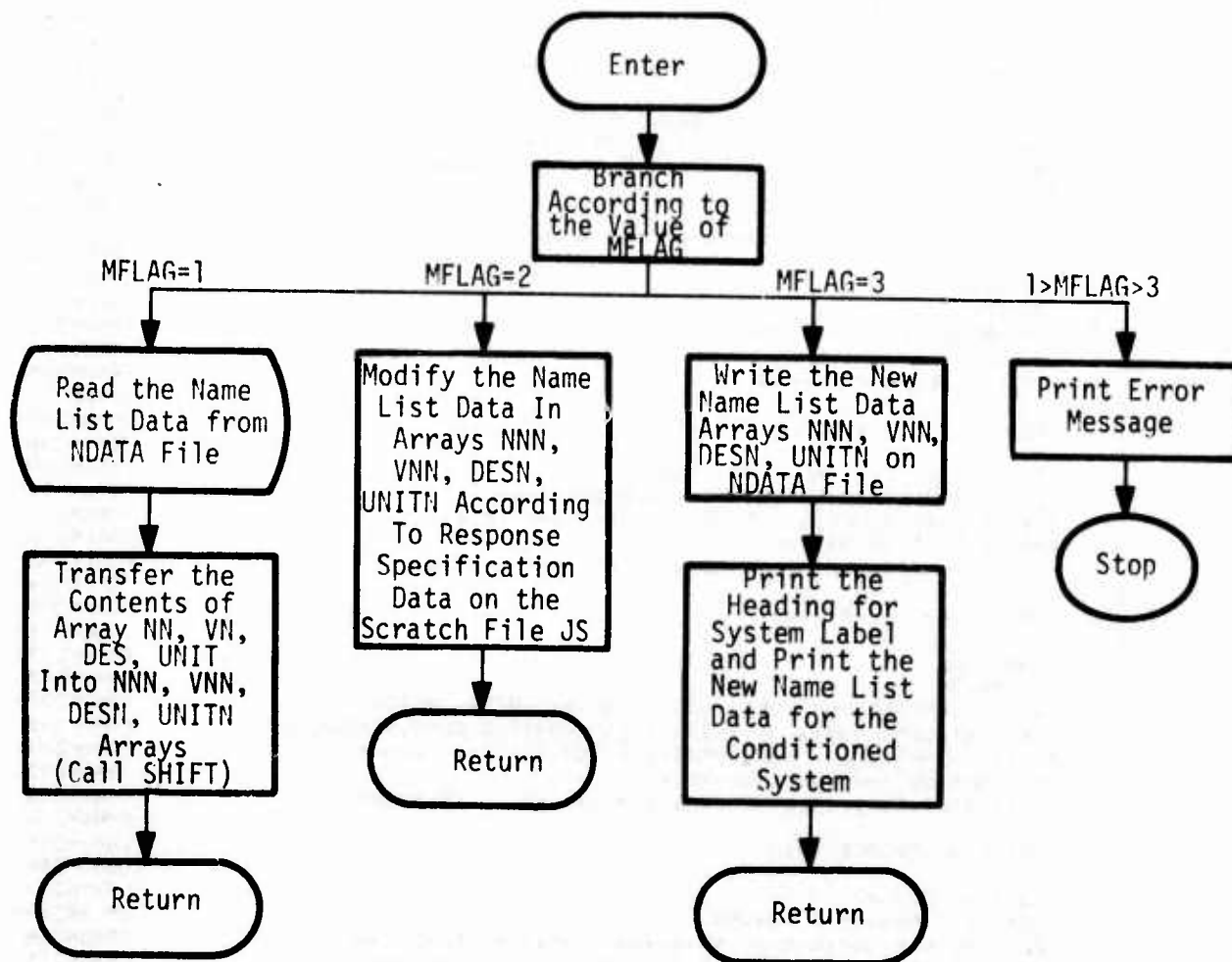


Figure 41. Subroutine MNAME Flow Chart

11W,IPRINT)	MNAME 65
CALL SHIFT(NNO,VNO,DESO,UNITO,NNNO,VNNO,DESN0,UNITNO,NR,NRM,	MNAME 66
11W,IPRINT)	MNAME 67
CALL SHIFT(NNI,VNI,DESI,UNITI,NNNI,VNVI,DESVI,UNITNI,NU,NUM,	MNAME 68
11W,IPRINT)	MNAME 69
IF(IPRINT.EQ.6)CALL DFRUG(2.4HMNAM,4HE .5.0,1W)	MNAME 70
RETURN	MNAME 71
C	MNAME 72
C MODIFY THE NAME LIST TABLE FOR INPUT VARIABLES	MNAME 73
C IF THE INPUT SPACE IS EXPANDED	MNAME 74
C	MNAME 75
160 CONTINUE	MNAME 76
IF(MFLAG.NE.2)GO TO 760	MNAME 77
IF(IPRINT.EQ.6)CALL DFRUG(3.4HMNAM,4HE .5.0,1W)	MNAME 78
IF(NUM.LT.NU)CALL FRRM(1.4HMNAM,4HE .6.0,1W)	MNAME 79
IF(NUM.EQ.NU)GO TO 280	MNAME 80
NUS=NU	MNAME 81
180 CONTINUE	MNAME 82
DO 26 I=1,NUS	MNAME 83
II=NU+I	MNAME 84
NNNI(II)=II	MNAME 85
ENCODE(4,200,VNVI(II,1))HUP,II	MNAME 86
200 FORMAT(A2,I2)	MNAME 87
VNVI(II,2)=HP	MNAME 88
DO 22 J=1,4	MNAME 89
220 UNITNI(II,J)=UNITI(II,J)	MNAME 90
DO 24 J=1,10	MNAME 91
240 DESNI(II,J)=DESI(II,J)	MNAME 92
260 CONTINUE	MNAME 93
NU=NU+NUS	MNAME 94
IF(NUM.LT.NU)CALL FRRM(2.4HMNAM,4HE .6.0,1W)	MNAME 95
IF(NUM.EQ.NU)GO TO 280	MNAME 96
GO TO 180	MNAME 97
C	MNAME 98
C MODIFY THE NAME LIST FOR OUTPUT VARIABLES AND INPUT VARIABLES	MNAME 99
C	MNAME100
280 CONTINUE	MNAME101
IF(IPRINT.EQ.6)CALL DFRUG(4.4HMNAM,4HE .5.0,1W)	MNAME102
IFLAG=0	MNAME103
J=0 \$ IRR=0 \$ JU=0 \$ JUI=0	MNAME104
340 CONTINUE	MNAME105
READ(15,480)CARD	MNAME106
IF(CARD(1).EQ.HENDR)RETURN	MNAME107
IF((CARD(7).NE.HONTR).AND.(CARD(3).NE.HUSTB).AND.(CARD(3).NE.	MNAME108
1HOMMA))GO TO 460	MNAME109
C	MNAME110
C OBTAIN NAME LIST DATA FOR SPECIFIED INPUTS	MNAME111
C	MNAME112
360 CONTINUE	MNAME113
READ(15,480)CARD	MNAME114
IF(CARD(1).EQ.HENDR)GO TO 340	MNAME115
JU=JU+1	MNAME116
DECODE(4,500,CARD(2))01,K,D2	MNAME117
NNNI(JU)=JU	MNAME118
NNI(JU)=JU	MNAME119
ENCODE(4,200,VNVI(JU,1))HUP,JU	MNAME120
VNVI(JU,2)=HP	MNAME121
VNI(JU,1)=VNVI(JU,1)	MNAME122
VNI(JU,2)=VNVI(JU,2)	MNAME123
DO 38 L=1,10	MNAME124
380 DESNI(JU,L)=DESI(K,L)	MNAME125
DO 40 L=1,4	MNAME126
400 UNITNI(JU,L)=UNITI(K,L)	MNAME127
GO TO 360	MNAME128
C	MNAME129
C OBTAIN NAME LIST DATA FOR SPECIFIED OUTPUTS	MNAME130

Figure 42. Subroutine MNAME Program Listing (Continued)

C	460 CONTINUE	MNAME131
	READ(15,4H0)CARD	MNAME132
480	FORMAT(20A4)	MNAME133
	IF(CARD(1).EQ.HENDRIG0 TO 360	MNAME134
	J=J+1	MNAME135
	DECODE(4,530,CARD(2)ID).K,02	MNAME136
500	FORMAT(A1,I2,A1)	MNAME137
	ID=0 * IX=0	MNAME138
	IF(CARD(1).EQ.HRRRRIG0 TO 520	MNAME139
	ID=0 * IX=1	MNAME140
	IF(CARD(1).EQ.HXRRRIG0 TO 520	MNAME141
	ID=1 * IX=0	MNAME142
	IF(CARD(1).EQ.HRDOTIG0 TO 520	MNAME143
	ID=1 * IX=1	MNAME144
	IF(CARD(1).EQ.HXDUTIG0 TO 520	MNAME145
	CALL FRRM(3,4HMNAM,4HE .6,0,[W)	MNAME146
520	CONTINUE	MNAME147
	VNNO(J)=J	MNAME148
	NNO(J)=J	MNAME149
	ENCODE(4,200,VNNO(J,1))HRP,J	MNAME150
	VNNO(J,2)=HP	MNAME151
	VNO(J,1)=VNNO(J,1)	MNAME152
	VNO(J,2)=VNNO(J,2)	MNAME153
	IF(ID.EQ.1)GO TO 640	MNAME154
	IF(IX.EQ.1)GO TO 540	MNAME155
	DO 540 L=1,10	MNAME156
540	DESN0(J,L)=DESO(K,L)	MNAME157
	DO 560 L=1,4	MNAME158
560	UNITNO(J,L)=UNITO(K,L)	MNAME159
	GO TO 460	MNAME160
580	CONTINUE	MNAME161
	DO 600 L=1,10	MNAME162
600	DESN0(J,L)=DESS(K,L)	MNAME163
	DO 620 L=1,4	MNAME164
620	UNITNO(J,L)=UNITS(K,L)	MNAME165
	GO TO 460	MNAME166
C		MNAME167
C	FORM NAME LIST DATA FOR DERIVATIVES OF SPECIFIED STATES	MNAME168
C	OR OUTPUTS	MNAME169
C		MNAME170
640	CONTINUE	MNAME171
	DESN0(J,1)=HDSOT	MNAME172
	DESN0(J,2)=HOF	MNAME173
	DESN0(J,3)=HPRRR	MNAME174
	DESN0(J,10)=HRRRR	MNAME175
	UNITNO(J,4)=HSSFC	MNAME176
	IF(IX.EQ.1)GO TO 740	MNAME177
	DO 660 L=1,6	MNAME178
	LL=3+L	MNAME179
660	DESN0(J,LL)=DESO(K,L)	MNAME180
	DO 680 L=1,3	MNAME181
680	UNITNO(J,L)=UNITO(K,L)	MNAME182
	GO TO 460	MNAME183
700	CONTINUE	MNAME184
	DO 720 L=1,6	MNAME185
	LL=3+L	MNAME186
720	DESN0(J,LL)=DESS(K,L)	MNAME187
	DO 740 L=1,3	MNAME188
740	UNITNO(J,L)=UNITS(K,L)	MNAME189
	GO TO 460	MNAME190
760	CONTINUE	MNAME191
	IF(IPRINT.EQ.6)CALL DERUG(5,4HMNAM,4HE .5,0,[W)	MNAME192
C		MNAME193
C	WRITE THE NEW NAME LIST TABLE ON THE DATA FILE	MNAME194
C		MNAME195
		MNAME196

Figure 42. Subroutine MNAME Program Listing (Continued)

```

IF (MFLAG.NE.3) CALL ERRM(4.4,MNAM,4HE .6.0,IW)
CALL FILE(JN,INSERT,HEAD)
WRITE(JN)NXN,NPN,NUN,
1 (NNNS(I),(VNNS(I,J),J=1,2),
2 (DESNS(I,J),J=1,10),(UNITNS(I,J),J=1,4),I=1,NKN),
3 (NNNO(I),(VNNO(I,J),J=1,2),
4 (DESNO(I,J),J=1,10),(UNITNO(I,J),J=1,4),I=1,NRN),
5 (NNNI(I),(VNNI(I,J),J=1,2),
6 (DESNI(I,J),J=1,10),(UNITNI(I,J),J=1,4),I=1,NUN)
CALL FILE(JN,INSERT,MARK)
C
C PRINT NAME LIST DATA
C
IF(IPINT.LT.2)RETURN
CALL HPR(HEAD,IW)
WRITE(IW,765)NXN,NPN,NUN
765 FORMAT(//,1X,18HNUMBER OF STATES =,I2,/,1X,
18HNUMBER OF OUTPUTS =,I2,/,1X,18HNUMBER OF INPUTS =,I2,/)
WRITE(IW,770)
770 FORMAT(//,20X,23H*** NAME LIST TABLE ***/)
WRITE(IW,780)
780 FORMAT(//,1X,8HVARIABLE,6H NAME ,6X,13H DESCRIPTION ,
131X,6X UNIT ,/)
C
C PRINT NAME LIST DATA FOR STATES
C
WRITE(IW,790)
790 FORMAT(//,1X,6HSTATE ,/)
WRITE(IW,800) (NNNS(I),(VNNS(I,J),J=1,2),(DESNS(I,J),J=1,10),
1 (UNITNS(I,J),J=1,4),I=1,NKN)
800 FORMAT(1X,I2,6X,2A4,4X,10A4,4X,4A4)
IF(NDR.EQ.0)GO TO 840
C
C PRINT NAME LIST DATA FOR DESIGN OUTPUTS
C
WRITE(IW,810)
810 FORMAT(//,1X,13HDESIGN OUTPUT,/)
WRITE(IW,800) (NNNO(I),(VNNO(I,J),J=1,2),(DESNO(I,J),J=1,10),
1 (UNITNO(I,J),J=1,4),I=1,NDR)
820 CONTINUE
NDRP=NDR-NPR
IF(NDR.EQ.0)GO TO 840
C
C PRINT NAME LIST DATA FOR PERFORMANCE OUTPUTS
C
WRITE(IW,830)
830 FORMAT(//,1X,18HPERFORMANCE OUTPUT,/)
NDRP1=NDR+1
WRITE(IW,800) (NNNO(I),(VNNO(I,J),J=1,2),(DESNO(I,J),J=1,10),
1 (UNITNO(I,J),J=1,4),I=NDRP1,NDR)
840 CONTINUE
IF(NSP.EQ.0)GO TO 860
C
C PRINT NAME LIST DATA FOR SENSOR OUTPUTS
C
WRITE(IW,850)
850 FORMAT(//,1X,13HSENSOR OUTPUT,/)
NDRP1=NDR+1
WRITE(IW,800) (NNNO(I),(VNNO(I,J),J=1,2),(DESNO(I,J),J=1,10),
1 (UNITNO(I,J),J=1,4),I=NDRP1,NRN)
860 CONTINUE
IF(NCL.EQ.0)GO TO 880
C
C PRINT NAME LIST DATA FOR CONTROL INPUTS
C
WRITE(IW,870)
MNAME197
MNAME198
MNAME199
MNAME200
MNAME201
MNAME202
MNAME203
MNAME204
MNAME205
MNAME206
MNAME207
MNAME208
MNAME209
MNAME210
MNAME211
MNAME212
MNAME213
MNAME214
MNAME215
MNAME216
MNAME217
MNAME218
MNAME219
MNAME220
MNAME221
MNAME222
MNAME223
MNAME224
MNAME225
MNAME226
MNAME227
MNAME228
MNAME229
MNAME230
MNAME231
MNAME232
MNAME233
MNAME234
MNAME235
MNAME236
MNAME237
MNAME238
MNAME239
MNAME240
MNAME241
MNAME242
MNAME243
MNAME244
MNAME245
MNAME246
MNAME247
MNAME248
MNAME249
MNAME250
MNAME251
MNAME252
MNAME253
MNAME254
MNAME255
MNAME256
MNAME257
MNAME258
MNAME259
MNAME260
MNAME261
MNAME262

```

Figure 42. Subroutine MNAME Program Listing (Continued)

870	FORMAT(/,1X,13CONTROL INPUT,/) WRITE(IW,850) (NINI(I), (VNNI(I,J),J=1,2), (DESNI(I,J),J=1,10), 1 (UNITNI(I,J),J=1,4), I=1,NCL)	MNAME263 MNAME264 MNAME265 MNAME266
880	CONTINUE NCG=NCL*NGT IF(NGT.EQ.0)GO TO 900	MNAME267 MNAME268 MNAME269
C	PRINT NAME LIST DATA FOR GUST INPUTS	MNAME270
C	WRITE(IW,890)	MNAME271
890	FORMAT(/,1X,10GUST INPUT,/) NCLP1=NCL*I WRITE(IW,890) (NINI(I), (VNNI(I,J),J=1,2), (DESNI(I,J),J=1,10), 1 (UNITNI(I,J),J=1,4), I=NCLP1*NGG)	MNAME272 MNAME273 MNAME274 MNAME275 MNAME276
900	CONTINUE IF(NCG.EQ.0)GO TO 920	MNAME277 MNAME278
C	PRINT NAME LIST DATA FOR COMMAND INPUTS	MNAME279
C	WRITE(IW,910)	MNAME280
910	FORMAT(/,1X,13COMMAND INPUT,/) NCGP1=NCG*I WRITE(IW,890) (NINI(I), (VNNI(I,J),J=1,2), (DESNI(I,J),J=1,10), 1 (UNITNI(I,J),J=1,4), I=NCGP1*NGG)	MNAME281 MNAME282 MNAME283 MNAME284 MNAME285
920	CONTINUE IF(IPRINT.EQ.6)CALL DFRUG(6,4*MMNAM,4HE .5,0,IW) RETURN	MNAME286 MNAME287 MNAME288 MNAME289
940	CONTINUE	MNAME290
C	PRINT NAME LIST DATA FOR OUTPUTS	MNAME291
C	WRITE(IW,950)	MNAME292
950	FORMAT(/,1X,6OUTPUT,/) WRITE(IW,800) (NINO(I), (VNNO(I,J),J=1,2), (DESNO(I,J),J=1,10), 1 (UNITNO(I,J),J=1,4), I=1,NRN)	MNAME293 MNAME294 MNAME295 MNAME296
C	PRINT NAME LIST DATA FOR INPUTS	MNAME297
C	WRITE(IW,960)	MNAME298
960	FORMAT(/,1X,5INPUT,/) WRITE(IW,800) (NINI(I), (VNNI(I,J),J=1,2), (DESNI(I,J),J=1,10), 1 (UNITNI(I,J),J=1,4), I=1,NUN) IF(IPRINT.EQ.6)CALL DERUG(7,4*MMNAM,4HE .5,0,IW) RETURN END	MNAME299 MNAME300 MNAME301 MNAME302 MNAME303 MNAME304 MNAME305 MNAME306 MNAME307

Figure 42. Subroutine MNAME Program Listing (Concluded)

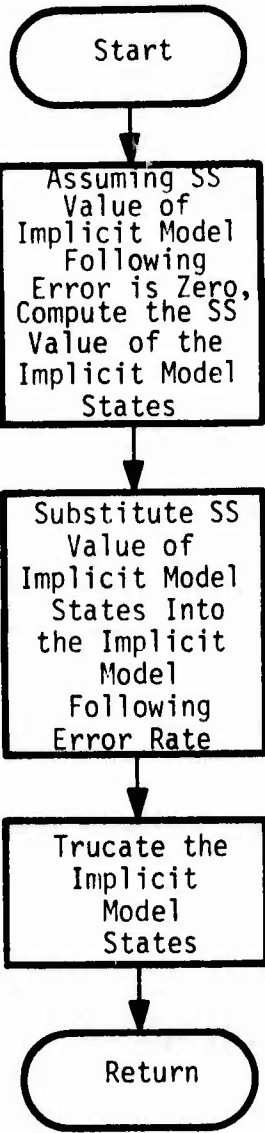


Figure 43. Subroutine IMRATE Flow Chart

DO 280 I=1, NR	IMRATE65
DO 280 J=1, NXA	IMRATE66
JJ=NXP+J	IMRATE67
DO 280 K=1, NXR	IMRATE68
KK=NXA+K	IMRATE69
280 CM(I, J)=CM(I, J)-DUMMY2(I, KK)*DUMMY1(K, JJ)	IMRATE70
DO 300 I=1, NR	IMRATE71
DO 300 J=1, NU	IMRATE72
JJ=NXP+J	IMRATE73
DO 300 K=1, NXR	IMRATE74
KK=NXA+K	IMRATE75
300 DM(I, J)=DM(I, J)-DUMMY2(I, KK)*DUMMY1(K, JJ)	IMRATE76
NX=NXP	IMRATE77
IF(IPRINT.LT.6)GO TO 320	IMRATE78
CALL MPRS(CM, NRM, NXM, NR, NXA, 0.0, 4HCM)	IMRATE79
CALL MPRS(DM, NRM, NUM, NR, NU, 0.0, 4HDM)	IMRATE80
CALL MPRS(DUMMY1, NDM11, NDM12, NDR, NDC, 0.0, 4HDMY1)	IMRATE81
320 CONTINUE	IMRATE82
RETURN	IMRATE83
END	IMRATE84

Figure 44. Subroutine IMRATE Program Listing (Concluded)

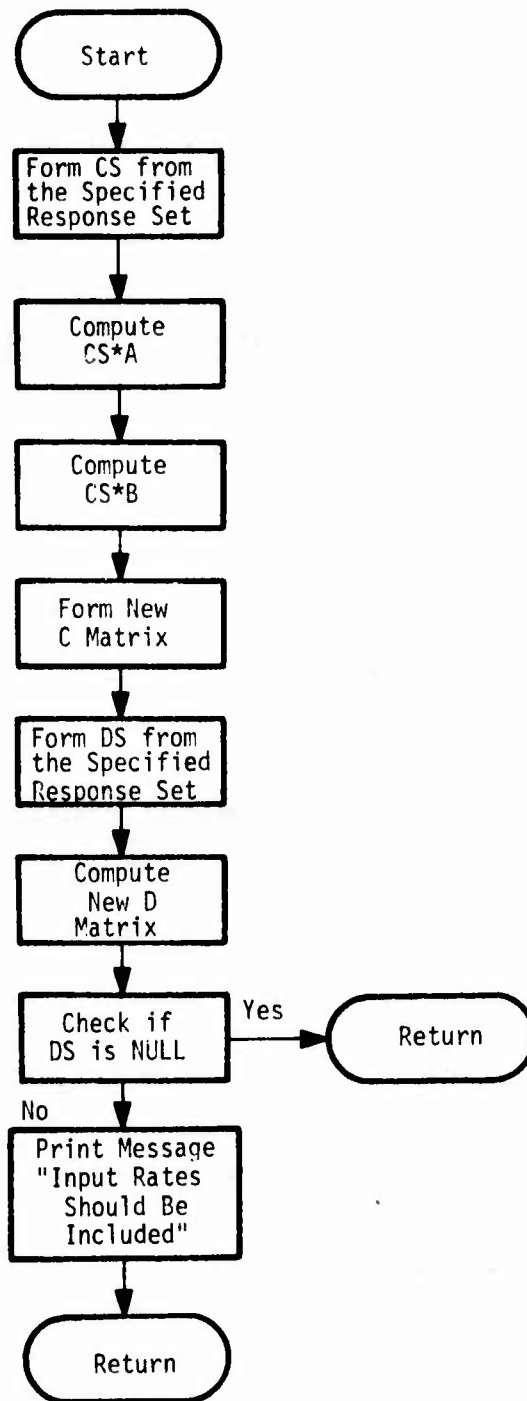


Figure 45. Subroutine DIFFK Flow Chart

```

SUBROUTINE DIFFK(A,B,C,D,DUMMY1,DUMMY2,NX,NR,NI,
INXM,NPM,NUM,NRS,IRS,IO,IW,IPOINT,NDM11,NDM12,NDM21,NDM22)
C
C PURPOSE - TO OBTAIN DERIVATIVES OF RESPONSES
C ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC
C DATE WRITTEN - 1975
C
C ARGUMENTS LIST
C NX INPUT NUMBER OF STATES
C NR INPUT NUMBER OF OUTPTS
C NU INPUT NUMBER OF INPUTS
C NRS INPUT NO OF RESPONSES TO BE DIFFERENTIATED
C IO INPUT CONTROLS ENTRY POINT IN THE SUBROUTINE
C IW INPUT FILE NO FOR LINE PRINTER
C IPOINT INPUT PRINT CONTROL FLAG
C OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
C DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NPM,NUM)
C DIMENSION IRS(NRM),DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22)
C NUS=NI
C IF (IO.GT.1)GO TO 140
C
C OBTAIN FIRST DERIVATIVES ONLY
C
C FORM CS MATRIX
C
C DO 10 I=1,NRS
C II=IR(I)
C DO 10 J=1,NX
10 DUMMY1(I,J)=C(II,J)
C
C COMPUTE CS*A MATRIX
C
C DO 30 I=1,NRS
C DO 30 J=1,NX
C DUMMY2(I,J)=0.0
C DO 30 K=1,NX
30 DUMMY2(I,J)=DUMMY2(I,J)+DUMMY1(I,K)*A(K,J)
C
C COMPUTE NEW C MATRIX
C
C DO 50 I=1,NRS
C II=NR+I
C DO 50 J=1,NX
50 C(II,J)=DUMMY2(I,J)
C
C FORM CS*B MATRIX
C
C DO 60 I=1,NRS
C DO 60 J=1,NU
C DUMMY2(I,J)=0.0
C DO 60 K=1,NX
60 DUMMY2(I,J)=DUMMY2(I,J)+DUMMY1(I,K)*B(K,J)
C
C FORM DS MATRIX
C
C DO 70 I=1,NRS
C II=IR(I)
C DO 70 J=1,NU
70 DUMMY1(I,J)=D(II,J)
C
C COMPUTE NEW D MATRIX
C
DIFFK 2
DIFFK 3
DIFFK 4
DIFFK 5
DIFFK 6
DIFFK 7
DIFFK 8
DIFFK 9
DIFFK 10
DIFFK 11
DIFFK 12
DIFFK 13
DIFFK 14
DIFFK 15
DIFFK 16
DIFFK 17
DIFFK 18
DIFFK 19
DIFFK 20
DIFFK 21
DIFFK 22
DIFFK 23
DIFFK 24
DIFFK 25
DIFFK 26
DIFFK 27
DIFFK 28
DIFFK 29
DIFFK 30
DIFFK 31
DIFFK 32
DIFFK 33
DIFFK 34
DIFFK 35
DIFFK 36
DIFFK 37
DIFFK 38
DIFFK 39
DIFFK 40
DIFFK 41
DIFFK 42
DIFFK 43
DIFFK 44
DIFFK 45
DIFFK 46
DIFFK 47
DIFFK 48
DIFFK 49
DIFFK 50
DIFFK 51
DIFFK 52
DIFFK 53
DIFFK 54
DIFFK 55
DIFFK 56
DIFFK 57
DIFFK 58
DIFFK 59
DIFFK 60
DIFFK 61
DIFFK 62
DIFFK 63
DIFFK 64

```

Figure 46. Subroutine DIFFK Program Listing

	DO 80 I=1,NRS	DIFFK 65
	DO 80 J=1,NU	DIFFK 66
	II=NR+1	DIFFK 67
80	D(I1,I)=DUMMY2(I,J)	DIFFK 68
C		DIFFK 69
C	CHECK IF DS MATRIX IS NULL	DIFFK 70
C		DIFFK 71
	DO 90 I=1,NRS	DIFFK 72
	DO 90 J=1,NRS	DIFFK 73
	IF(DUMMY1(I,J).NE.1.0)GO TO 100	DIFFK 74
90	CONTINUE	DIFFK 75
	NR=NR+NRS	DIFFK 76
	RETURN	DIFFK 77
C		DIFFK 78
C	PRINT A MESSAGE THAT THE INPUT RATES ARE NECESSARY FOR	DIFFK 79
C	CORRECTLY OBTAINING THE DERIVATIVES OF THE RESPONSES	DIFFK 80
C		DIFFK 81
100	CONTINUE	DIFFK 82
	WRITE(IW,120)	DIFFK 83
120	FORMAT(1H1,/,/,1X,A5H*** THE INPUT RATES SHOULD BE INCLUDED IN TAKING THE DERIVATIVES OF THE RESPONSES **/,/)	DIFFK 84
	RETURN	DIFFK 85
140	CONTINUE	DIFFK 86
		DIFFK 87
C		DIFFK 88
C	OBTAIN FIRST AND SECOND DERIVATIVES	DIFFK 89
C		DIFFK 90
	WRITE(IW,160)	DIFFK 91
160	FORMAT(/,/,1X,A5H*** THE SECOND DERIVATIVE OPTION IS NOT IMPLEMENTED	DIFFK 92
	ID **/,/)	DIFFK 93
	RETURN	DIFFK 94
	END	DIFFK 95

Figure 46. Subroutine DIFFK Program Listing (Concluded)

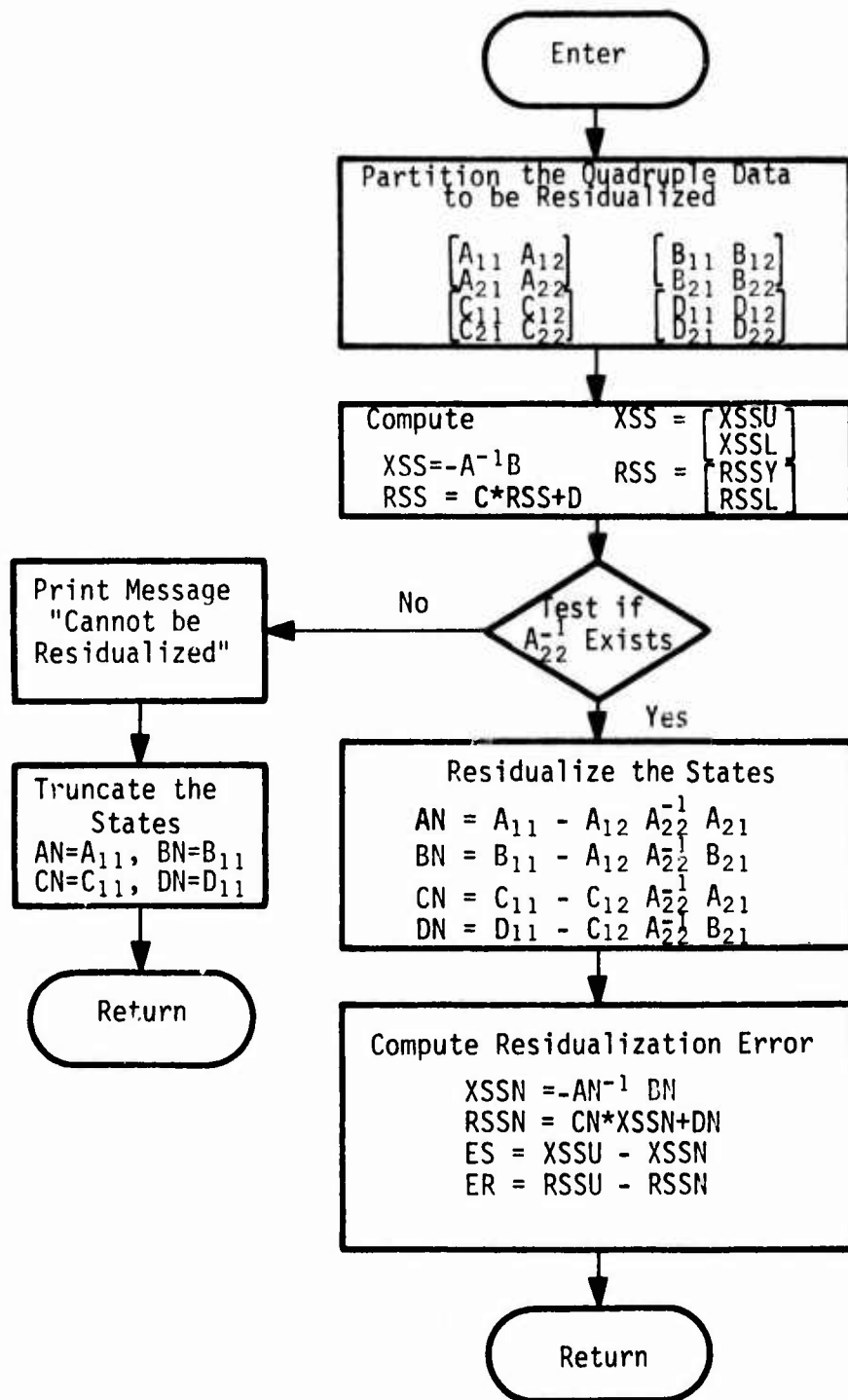


Figure 47. Subroutine REDUCE Flow Chart

	SUBROUTINE REDUCE(A,B,C,D,DUMMY1,DUMMY2,ES,ER, INX,NR,NU,NXR,NPR,NRT,NXM,NR4,NUM,T,IW,IPRINT, 2NDM11,NDM12,NDM21,NDM22)	REDUCE 2
		REDUCE 3
		REDUCE 4
		REDUCE 5
C		REDUCE 6
C	PURPOSE - TO REDUCE THE ORDER OF STATE SPACE DATA ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC	REDUCE 7
C	DATE WRITTEN - 1975	REDUCE 8
C		REDUCE 9
C	SUBPROGRAMS CALLED	REDUCE 10
C	TDINVR	REDUCE 11
C	MPRS	REDUCE 12
C		REDUCE 13
C	ARGUMENT LIST	REDUCE 14
C	NX INPUT NUMBER OF STATES	REDUCE 15
C	NR INPUT NUMBER OF OUTPUTS	REDUCE 16
C	NU INPUT NUMBER OF INPUTS	REDUCE 17
C	NXR INPUT NO OF STATES TO BE RESIDUALIZED	REDUCE 18
C	NR2 INPUT NO OF OUTPUTS TO BE RESIDUALIZED	REDUCE 19
C	NRT INPUT NO OF OUTPUTS TO BE TRUNCATED	REDUCE 20
C	T INPUT SAMPLE TIME	REDUCE 21
C	IW INPUT FILE NO FOR LINE PRINTER	REDUCE 22
C	IPOINT INPUT PRINT CONTROL FLAG	REDUCE 23
C	OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM	REDUCE 24
C		REDUCE 25
	DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM)	REDUCE 26
	DIMENSION DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22)	REDUCE 27
	DIMENSION ES(NXM,NUM),ER(NRM,NUM)	REDUCE 28
	NXM=NX-NXR	REDUCE 29
	NRM=NDT+NRR	REDUCE 30
	NUM=NI	REDUCE 31
C		REDUCE 32
C	COMPUTE SS VALUE OF STATE AND OUTPUT FOR ORIGINAL SYSTEM	REDUCE 33
C		REDUCE 34
	IF(IPRINT.LT.5)GO TO 260	REDUCE 35
	DO 140 I=1,NX	REDUCE 36
	DO 140 J=1,NX	REDUCE 37
140	DUMMY1(I,J)=A(I,J)	REDUCE 38
	DO 160 I=1,NX	REDUCE 39
	DO 160 J=1,NU	REDUCE 40
	JJ=NX+J	REDUCE 41
160	DUMMY1(I,JJ)=R(I,J)	REDUCE 42
	NDR=NX	REDUCE 43
	NDC=NX+NU	REDUCE 44
	IF(IPOINT.GE.6)CALL MPRES(DUMMY1,NDM11,NDM12,NDR,NDC,T,4,DUMMY1)	REDUCE 45
	CALL TDINVR(ISOL,IDSOL,NDR,-NDC,DUMMY1,NDM11,DUMMY2,DET)	REDUCE 46
	IF((ISOL.EQ.1).AND.(IDSOL.EQ.1))GO TO 200	REDUCE 47
	WRITE(IW,180)ISOL,IDSOL	REDUCE 48
180	FORMAT(/,IX,14HTDINVR FAILURE,6H ISOL=,I2,7H IDSOL=,I2)	REDUCE 49
	WRITE(IW,190)	REDUCE 50
190	FORMAT(/,IX,40HRESIDUALIZATION ERROR CANNOT BE COMPUTED, I/,IX,30H SINCE SS VALUES DOES NOT EXIST)	REDUCE 51
	JPRINT=2	REDUCE 52
	GO TO 260	REDUCE 53
200	CONTINUE	REDUCE 54
	JPRINT=IPRINT	REDUCE 55
	DO 220 I=1,NX	REDUCE 56
	DO 220 J=1,NU	REDUCE 57
	JJ=NX+J	REDUCE 58
220	ES(I,J)=-DUMMY1(I,JJ)	REDUCE 59
	DO 240 I=1,NR	REDUCE 60
	DO 240 J=1,NU	REDUCE 61
	ER(I,J)=D(I,J)	REDUCE 62
	DO 240 K=1,NX	REDUCE 63
		REDUCE 64

Figure 48. Subroutine REDUCE Program Listing

240	ER(I, I)=ER(I, J)+C(I, K)*ES(K, J)	REDUCE65
	IF(IPRINT.GE.6)CALL MPRS(ES,NXM,NUM,NX,NU,T,4HES)	REDUCE66
	IF(IPRINT.GE.6)CALL MPRS(EP,NPM,NUM,NR,NU,T,4HER)	REDUCE67
260	CONTINUE	REDUCE68
C	QUADRUPLE DATA IS REDUCED BY RESIDUALIZATION	REDUCE69
C		REDUCE70
	DO 280 I=1,NXR	REDUCE71
	II=NXN+I	REDUCE72
	DO 280 J=1,NXR	REDUCE73
	JJ=NXN+J	REDUCE74
280	DUMMY1(I, J)=A(I, J)	REDUCE75
	DO 300 I=1,NXR	REDUCE76
	II=NXN+I	REDUCE77
	DO 300 J=1,NXR	REDUCE78
	JJ=NXN+J	REDUCE79
300	DUMMY1(I, JJ)=A(II, J)	REDUCE80
	DO 320 I=1,NXR	REDUCE81
	II=NXN+I	REDUCE82
	DO 320 J=1,NUN	REDUCE83
	JJ=NXN+NXN+J	REDUCE84
320	DUMMY1(I, JJ)=B(II, J)	REDUCE85
	NDR=NXR	REDUCE86
	NDC=NXR+NXN+NUN	REDUCE87
	IF(IPRINT.GE.6)CALL MPRS(DUMMY1,NDM11,NDM12,NDR,NDC,T,4HDMY1)	REDUCE88
	CALL TDINVR(ISOL,IDSOL,NDR,-NDC,DUMMY1,NDM11,DUMMY2,DET)	REDUCE89
	IF((ISOL.EQ.1).AND.(IDSOL.EQ.1))GO TO 360	REDUCE90
C		REDUCE91
C	RESIDUALIZATION IS NOT POSSIBLE AND SO	REDUCE92
C	QUADRUPLE DATA IS REDUCED BY TRUNCATION	REDUCE93
C		REDUCE94
	WRITE(IW,189)ISOL,IDSOL	REDUCE95
	WRITE(IW,340)	REDUCE96
340	FORMAT(/,1X,47HCANNOT BE RESIDUALIZED SINCE SS VALUE OF STATES,	REDUCE97
	1/,1X,31HBEING ELIMINATED DOES NOT EXIST)	REDUCE98
	WRITE(IW,350)	REDUCE99
350	FORMAT(/,1X,31HQ DATA IS REDUCED BY TRUNCATION)	REDUC100
	NX=NXN	REDUC101
	NR=NRN	REDUC102
	NU=NUM	REDUC103
	RETURN	REDUC104
C		REDUC105
C	COMPUTE RESIDUALIZED QUADRUPLES	REDUC106
C		REDUC107
360	CONTINUE	REDUC108
	DO 380 I=1,NX	REDUC109
	DO 380 J=1,NX	REDUC110
380	DUMMY2(I, J)=A(I, J)	REDUC111
	DO 400 I=1,NXN	REDUC112
	DO 400 J=1,NXN	REDUC113
	JJ=NXR+J	REDUC114
	DO 400 K=1,NXR	REDUC115
	KK=NXN+K	REDUC116
400	A(I, J)=A(I, J)-DUMMY2(I, KK)*DUMMY1(K, JJ)	REDUC117
	DO 420 I=1,NXN	REDUC118
	DO 420 J=1,NUN	REDUC119
	JJ=NXN+NXN+J	REDUC120
	DO 420 K=1,NXR	REDUC121
	KK=NXN+K	REDUC122
420	B(I, J)=B(I, J)-DUMMY2(I, KK)*DUMMY1(K, JJ)	REDUC123
	DO 440 I=1,NR	REDUC124
	DO 440 J=1,NX	REDUC125
440	DUMMY2(I, J)=C(I, J)	REDUC126
	DO 460 I=1,NRN	REDUC127
	DO 460 J=1,NXN	REDUC128
	JJ=NXR+J	REDUC129
		REDUC130

Figure 48. Subroutine REDUCE Program Listing (Continued)

	DO 460 K=1,NXR	REDUC131
	KK=NXN+K	REDUC132
460	C(I,J)=C(I,J)-DUMMY2(I,KK)*DUMMY1(K,JJ)	REDUC133
	DO 480 I=1,NRR	REDUC134
	DO 480 J=1,NUN	REDUC135
	JJ=NXR+NXN+J	REDUC136
	DO 480 K=1,NXR	REDUC137
	KK=NXN+K	REDUC138
480	D(I,J)=D(I,J)-DUMMY2(I,KK)*DUMMY1(K,JJ)	REDUC139
C		REDUC140
C	COMPUTE SS VALUE OF STATE AND OUTPUT FOR REDUCED SYSTEM	REDUC141
C	AND SUBTRACT IT FROM SS VALUE OBTAINED EARLIER TO GET	REDUC142
C	THE ERROR OF RESIDUALIZATION	REDUC143
C		REDUC144
	IF(JPRINT.LT.3)GO TO 600	REDUC145
	DO 500 I=1,NXN	REDUC146
	DO 500 J=1,NXN	REDUC147
500	DUMMY1(I,J)=A(I,J)	REDUC148
	DO 520 I=1,NXN	REDUC149
	DO 520 J=1,NUN	REDUC150
	JJ=NXN+J	REDUC151
520	DUMMY1(I,JJ)=B(I,J)	REDUC152
	NDR=NXN	REDUC153
	NDC=NXN+NUN	REDUC154
	IF(JPRINT.GE.6)CALL MPRS(DUMMY1,NDM11,NDM12,NDR,NDC,T,4HDMY1)	REDUC155
	CALL TDINVR(ISOL,IDSOL,NDR,-NDC,DUMMY1,NDM11,DUMMY2,DET)	REDUC156
	IF((ISOL.GT.1).OR.(IDSOL.GT.1))GO TO 620	REDUC157
C		REDUC158
C	COMPUTE RESIDUALIZATION ERROR	REDUC159
C		REDUC160
	DO 540 I=1,NXN	REDUC161
	DO 540 J=1,NUN	REDUC162
	JJ=NXN+J	REDUC163
540	ES(I,J)=ES(I,J)+DUMMY1(I,JJ)	REDUC164
	DO 560 I=1,NRN	REDUC165
	DO 560 J=1,NUN	REDUC166
	ER(I,J)=-D(I,J)*ER(I,J)	REDUC167
	JJ=NXN+J	REDUC168
	DO 560 K=1,NXN	REDUC169
560	ER(I,J)=ER(I,J)+G(I,K)*DUMMY1(K,JJ)	REDUC170
	WRITE(IW,580)	REDUC171
580	FORMAT(/,IX,40HRESIDUALIZATION ERROR MATRICES ES AND ER,/)	REDUC172
	CALL MPRS(ES,NXM,NUM,NXN,NUN,T,4HES)	REDUC173
	CALL MPRS(ER,NRM,NUM,NRN,NUN,T,4HER)	REDUC174
600	CONTINUE	REDUC175
	NX=NXN	REDUC176
	NR=NRN	REDUC177
	NU=NUN	REDUC178
	RETURN	REDUC179
620	CONTINUE	REDUC180
	STOP 0030	REDUC181
	END	REDUC182

Figure 48. Subroutine REDUCE Program Listing (Concluded)

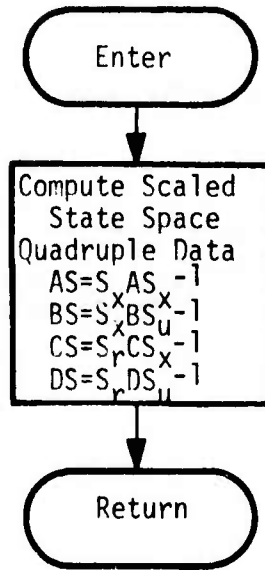


Figure 49. Subroutine SCAL Flow Chart

	SUBROUTINE SCAL (A,R,C,D,SCFS,SCFO,SCFI, INXN,NFN,NUN,NXM,NRN,NUM)	SCAL 2
		SCAL 3
C		SCAL 4
C	PURPOSE - TO COMPUTE SCALED QUADRUPLES	SCAL 5
C	ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC	SCAL 6
C	DATE WRITTEN - 1975	SCAL 7
C		SCAL 8
C	ARGUMENT LIST	SCAL 9
C	SCFS INPUT SCALING ARRAY FOR STATE	SCAL 10
C	SCFO INPUT SCALING ARRAY FOR OUTPUT	SCAL 11
C	SCFI INPUT SCALING ARRAY FOR INPUT	SCAL 12
C	NXM INPUT NUMBER OF REDUCED STATES	SCAL 13
C	NRN INPUT NUMBER OF REDUCED OUTPUTS	SCAL 14
C	NUM INPUT NUMBER OF REDUCED INPUTS	SCAL 15
C	OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM	SCAL 16
C		SCAL 17
	DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM)	SCAL 18
	DIMENSION SCFS(NXM),SCFO(NRM),SCFI(NUM)	SCAL 19
	DO 15 I=1,NXN	SCAL 20
	DO 15 J=1,NXM	SCAL 21
150	A(I,J)=SCFS(I)*A(I,J)/SCFS(J)	SCAL 22
	DO 17 I=1,NXN	SCAL 23
	DO 17 J=1,NUN	SCAL 24
170	R(I,J)=SCFS(I)*R(I,J)/SCFI(J)	SCAL 25
	DO 19 I=1,NRN	SCAL 26
	DO 19 J=1,NXM	SCAL 27
190	C(I,J)=SCFO(I)*C(I,J)/SCFS(J)	SCAL 28
	DO 21 I=1,NRN	SCAL 29
	DO 21 J=1,NUN	SCAL 30
210	D(I,J)=SCFO(I)*D(I,J)/SCFI(J)	SCAL 31
	RETURN	SCAL 32
	END	SCAL 33

Figure 50. Subroutine SCAL Program Listing

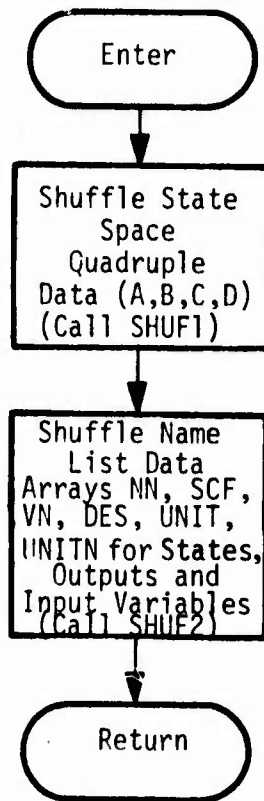


Figure 51. Subroutine SHUFF Flow Chart

```

SUBROUTINE SHUFF(A,R,C,D,NNS,VNS,DESS,UNITS,
1NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI,
2NSHUFF,NSHUF0,NSHUF1,DUMMY1,DUMMY2,NX,NR,NU,NXM,NRM,NUM,
3NDM11,NDM12,NDM21,NDM22)
C
C PURPOSE - TO SHUFFLE QUADRUPLE DATA AND NAME LIST DATA
C ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC
C DATE WRITTEN - MAY, 1975
C
C SUBPROGRAMS CALLED
C SHUF1
C SHUF2
C
C ARGUMENT LIST
C NX INPUT NUMBER OF STATES
C NR INPUT NUMBER OF OUTPTS
C NU INPUT NUMBER OF INPUTS
C OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
C DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
C DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
C DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
C DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
C DIMENSION NSHUFF(NXM),NSHUF0(NRM),NSHUF1(NUM)
C DIMENSION DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22)
C
C SHUFFLE A R C D MATRICES
C
C CALL SHUF1(A,NSHUFF,NSHUFF,DUMMY1,NXM,NXM,NX,NX,NDM11,NDM12)
C CALL SHUF1(B,NSHUFF,NSHUF1,DUMMY1,NXM,NUM,NX,NU,NDM11,NDM12)
C CALL SHUF1(C,NSHUF0,NSHUFF,DUMMY1,NRM,NXM,NR,NX,NDM11,NDM12)
C CALL SHUF1(D,NSHUF0,NSHUF1,DUMMY1,NRM,NUM,NR,NU,NDM11,NDM12)
C
C SHUFFLE SCALING, UNIT AND DESCRIPTION ARRAYS
C
C CALL SHUF2(NNS,VNS,DESS,UNITS,NSHUFF,DUMMY1,DUMMY2,
1NXM,NX,NDM11,NDM12,NDM21,NDM22)
C CALL SHUF2(NNO,VNO,DESO,UNITO,NSHUF0,DUMMY1,DUMMY2,
1NRM,NR,NDM11,NDM12,NDM21,NDM22)
C CALL SHUF2(NNI,VNI,DESI,UNITI,NSHUF1,DUMMY1,DUMMY2,
1NUM,NU,NDM11,NDM12,NDM21,NDM22)
RETURN
END
SHUFF 2
SHUFF 3
SHUFF 4
SHUFF 5
SHUFF 6
SHUFF 7
SHUFF 8
SHUFF 9
SHUFF 10
SHUFF 11
SHUFF 12
SHUFF 13
SHUFF 14
SHUFF 15
SHUFF 16
SHUFF 17
SHUFF 18
SHUFF 19
SHUFF 20
SHUFF 21
SHUFF 22
SHUFF 23
SHUFF 24
SHUFF 25
SHUFF 26
SHUFF 27
SHUFF 28
SHUFF 29
SHUFF 30
SHUFF 31
SHUFF 32
SHUFF 33
SHUFF 34
SHUFF 35
SHUFF 36
SHUFF 37
SHUFF 38
SHUFF 39
SHUFF 40
SHUFF 41
SHUFF 42
SHUFF 43
SHUFF 44

```

Figure 52. Subroutine SHUFF Program Listing

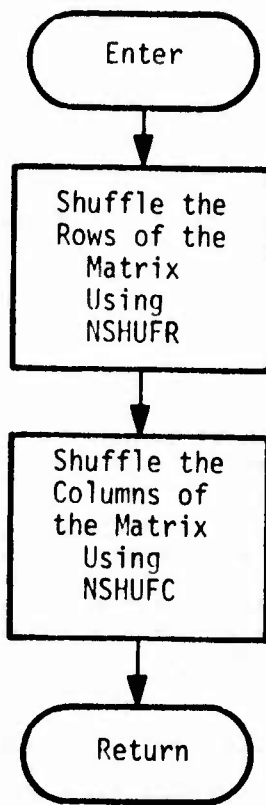


Figure 53. Subroutine SHUFF1 Flow Chart

	SUBROUTINE SHUF1 (ARCD, NSHUF, NSHUF, DUMMY, NRM, NCM, NR, NC, INDM11, NDM12)	SHUF1 2
C		SHUF1 3
C	PURPOSE - TO SHUFFLE THE MATRIX ARCD	SHUF1 4
C	ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC	SHUF1 5
C	DATE WRITTEN - 1975	SHUF1 6
C		SHUF1 7
C	ARGUMENT LIST	SHUF1 8
C	ABCD IN/OUT MATRIX TO BE SHUFFLED	SHUF1 9
C	NSHUF INPUT ROW SHUFFLING ARRAY	SHUF1 10
C	NSHUF INPUT COLUMN SHUFFLING ARRAY	SHUF1 11
C	NR INPUT MAXIMUM NUMBER OF ROWS	SHUF1 12
C	NCM INPUT MAXIMUM NUMBER OF COLUMNS	SHUF1 13
C	NR INPUT NUMBER OF ROWS	SHUF1 14
C	NC INPUT NUMBER OF COLUMNS	SHUF1 15
C	OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM	SHUF1 16
C		SHUF1 17
C	DIMENSION ARCD (NRM, NCM), DUMMY (NDM11, NDM12), NSHUF (NRM	SHUF1 18
	DIMENSION NSHUF (NCM)	SHUF1 19
	DO 12 I=1, NR	SHUF1 20
	II=NSHUF(I)	SHUF1 21
	DO 12 J=1, NC	SHUF1 22
120	DUMMY (I, J)=ARCD (II, J)	SHUF1 23
	DO 14 J=1, NC	SHUF1 24
	JJ=NSHUF(J)	SHUF1 25
	DO 14 I=1, NR	SHUF1 26
140	ARCD (I, J)=DUMMY (I, JJ)	SHUF1 27
	RETURN	SHUF1 28
	END	SHUF1 29
		SHUF1 30

Figure 54. Subroutine SHUF1 Program Listing

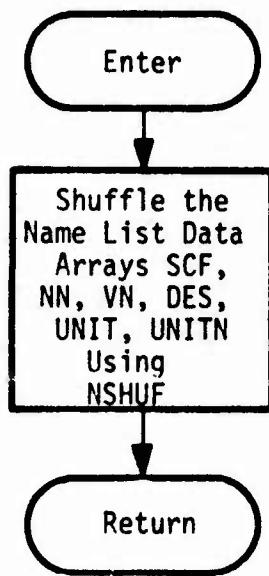


Figure 55. Subroutine SHUF2 Flow Chart

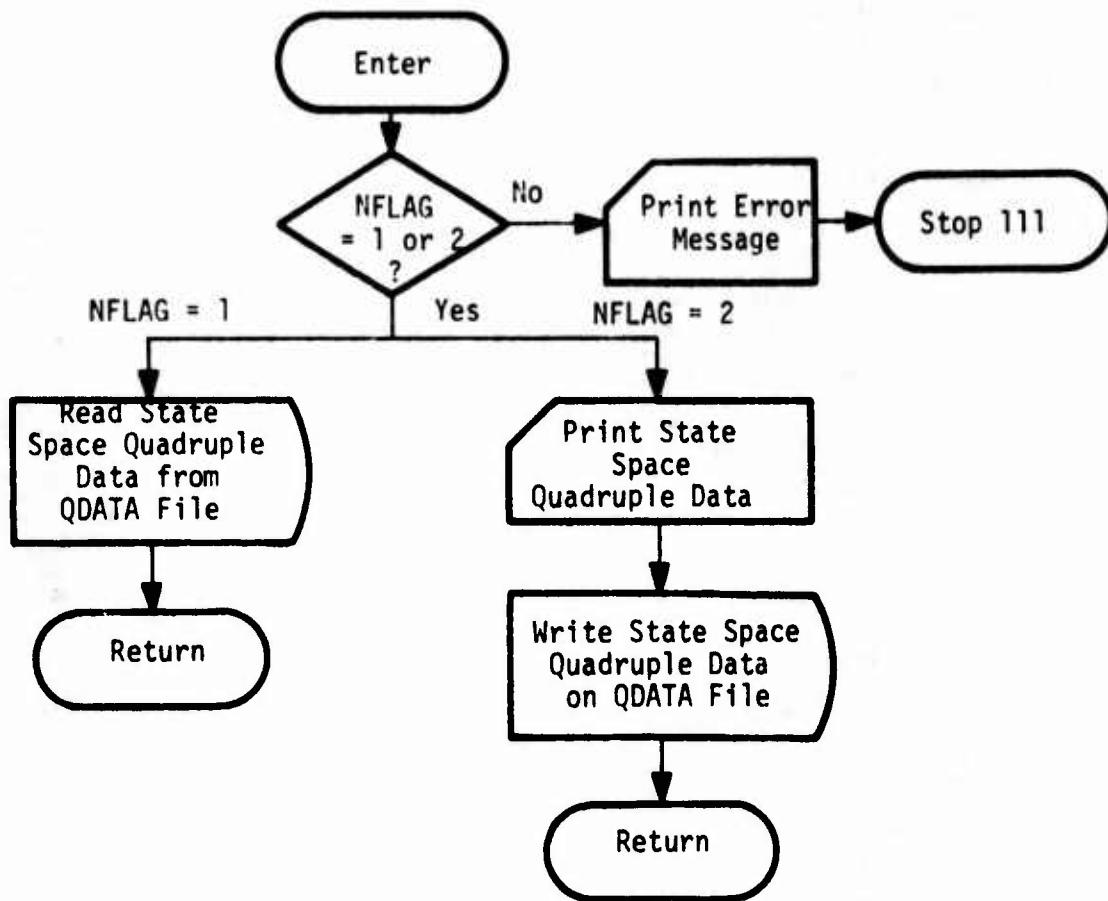


Figure 57. Subroutine QDIO Flow Chart

CALL MPRS(B,NM,NUM,NX,NU,T,4HG)	QDIO 65
CALL MPRS(C,NRM,NXM,NR,NX,T,4HM)	QDIO 66
CALL MPRS(D,NRM,NUM,NR,NU,T,4HE)	QDIO 67
160 CONTINUE	QDIO 68
IF(IQ,NE.1)GO TO 200	QDIO 69
C	QDIO 70
C PRINT WEIGHTING MATRIX Q	QDIO 71
C	QDIO 72
WRITE(IW,180)	QDIO 73
180 FORMAT(//,20X,47H*** STARTING WEIGHTS FOR OPTIMAL CONTROL DESIGN,	QDIO 74
14H ***,//)	QDIO 75
CALL MPRS(Q,NRM,NRM,NR,NR,T,4HQ0)	QDIO 76
200 CONTINUE	QDIO 77
CALL FILE(JQ,INSERT,LABEL)	QDIO 78
IF(IQ,NE.1)GO TO 210	QDIO 79
C	QDIO 80
C WRITE QUADRUPLE DATA AND WEIGHTING MATRIX Q ON FILE QDATA	QDIO 81
C	QDIO 82
WRITE(JQ)T,NX,NR,NU,	QDIO 83
1((A(I,J),I=1,NX),J=1,NX),	QDIO 84
2((B(I,J),I=1,NX),J=1,NU),	QDIO 85
3((C(I,J),I=1,NR),J=1,NX),	QDIO 86
4((D(I,J),I=1,NR),J=1,NU),	QDIO 87
5NXA,NPA,NUA,NR1,NR2,NR3,NU1,NU2,NU3,	QDIO 88
6((Q(I,J),I=1,NR1),J=1,NR1)	QDIO 89
CALL FILE(JQ,INSERT,MARK)	QDIO 90
RETURN	QDIO 91
210 CONTINUE	QDIO 92
C	QDIO 93
C WRITE QUADRUPLE DATA ON FILE QDATA	QDIO 94
C	QDIO 95
WRITE(JQ)T,NX,NR,NU,	QDIO 96
1((A(I,J),I=1,NX),J=1,NX),	QDIO 97
2((B(I,J),I=1,NX),J=1,NU),	QDIO 98
3((C(I,J),I=1,NR),J=1,NX),	QDIO 99
4((D(I,J),I=1,NR),J=1,NU),	QDIO 100
5NXA,NPA,NUA,NR1,NR2,NR3,NU1,NU2,NU3	QDIO 101
CALL FILE(JQ,INSERT,MARK)	QDIO 102
RETURN	QDIO 103
220 CONTINUE	QDIO 104
IF(INFLAG,NE.1)CALL ERPM(1.4HQDIO,4H .0.0,IW)	QDIO 105
IF(IPRINT.EQ.6)WRITE(IW,225)LABEL	QDIO 106
225 FORMAT(1X,20A4)	QDIO 107
CALL FILE(JQ,LOCATE,LABEL)	QDIO 108
IF(IQ,NE.1)GO TO 230	QDIO 109
C	QDIO 110
C READ QUADRUPLE DATA AND WEIGHTING MATRIX Q FROM FILE QDATA	QDIO 111
C	QDIO 112
READ(JQ)T,NX,NR,NU,	QDIO 113
1((A(I,J),I=1,NX),J=1,NX),	QDIO 114
2((B(I,J),I=1,NX),J=1,NU),	QDIO 115
3((C(I,J),I=1,NR),J=1,NX),	QDIO 116
4((D(I,J),I=1,NR),J=1,NU),	QDIO 117
5NXA,NPA,NUA,NR1,NR2,NR3,NU1,NU2,NU3,	QDIO 118
6((Q(I,J),I=1,NR1),J=1,NR1)	QDIO 119
RETURN	QDIO 120
C	QDIO 121
C READ QUADRUPLE DATA FROM FILE QDATA	QDIO 122
C	QDIO 123
230 CONTINUE	QDIO 124
READ(IQ)T,NX,NR,NU,	QDIO 125
1((A(I,J),I=1,NX),J=1,NX),	QDIO 126
2((B(I,J),I=1,NX),J=1,NU),	QDIO 127
3((C(I,J),I=1,NR),J=1,NX),	QDIO 128
4((D(I,J),I=1,NR),J=1,NU),	QDIO 129
5NXA,NPA,NUA,NR1,NR2,NR3,NU1,NU2,NU3	QDIO 130

Figure 58. Subroutine QDIO Program Listing (Continued)

RETURN
END

0010 131
0010 132

Figure 58. Subroutine QDIO Program Listing (Concluded)

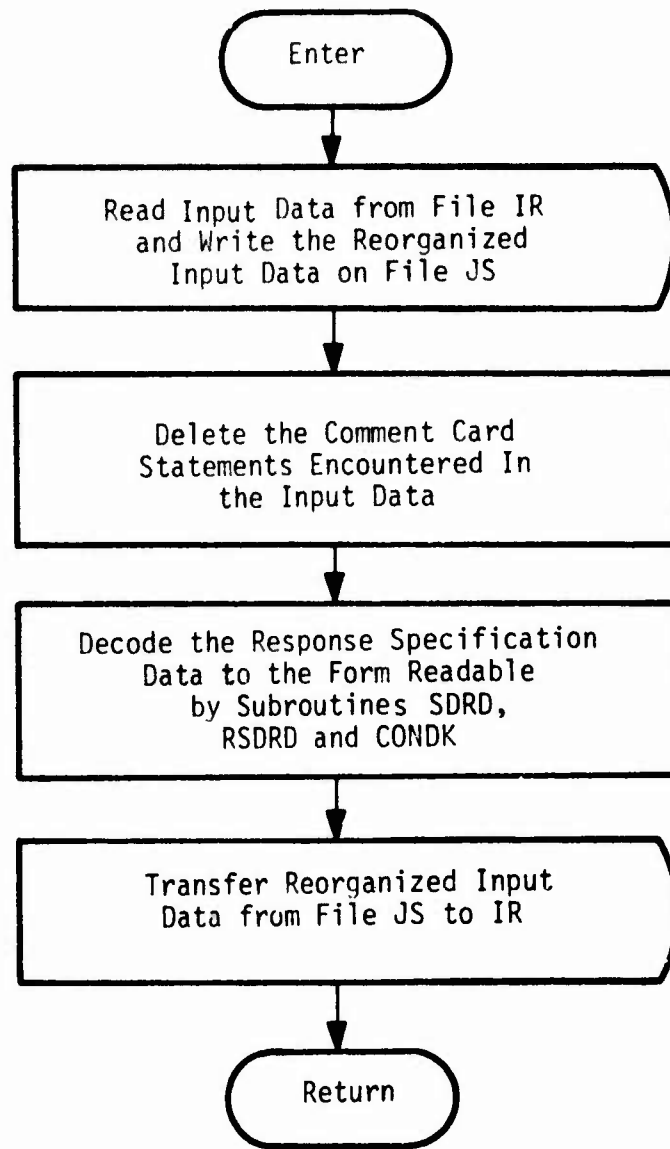


Figure 59. Subroutine IDRO Flow Chart

C	SUBROUTINE IDRO(IR,IW,JS)	IDRO	2
C		IDRO	3
C	PURPOSE - TO REORGANIZE INPUT DATA	IDRO	4
C	ANALYSIS - A F KONAR / J K VAHESH - THE MONEYWELL INC	IDRO	5
C	DATE WRITTEN - 1975	IDRO	6
C		IDRO	7
C	ARGUMENTS LIST	IDRO	8
C	IR INPUT FILE NO FOR CARD READER BUFFER	IDRO	9
C	IW INPUT FILE NO FOR LINE PRINTED	IDRO	10
C	JS INPUT FILE NO FOR SCRATCH FILE	IDRO	11
C		IDRO	12
	DIMENSION CARD1(20),CARD2(8),CARD3(76),CARD4(20),CHEAD(6)	IDRO	13
	DATA CHEAD /4MCONS,4MSELE,4HRITA,4MTRUN,4HREST,4MSCAL/	IDRO	14
	DATA MRRH,MENOR,MR,HRP,MRP /4M *4MEND *1M *1M(1M)/	IDRO	15
	DATA MCOM,HDAS,MDOT /1M *1M *1M /	IDRO	16
	DATA MHR,MHI/1M *1M *1M /	IDRO	17
	DATA MCR,MR/2MC *1M /	IDRO	18
	NID=6	IDRO	19
	RFWIN=IR	IDRO	20
	RFWIN=JS	IDRO	21
	CARD3(2)=MRP	IDRO	22
	CARD3(4)=MRP	IDRO	23
	DO 10 I=5,76	IDRO	24
100	CARD3(I)=MR	IDRO	25
	CARD4(I)=MENOR	IDRO	26
	DO 11 I=2,20	IDRO	27
110	CARD4(I)=MRRH	IDRO	28
C		IDRO	29
C	READ CARD IMAGES FROM FILE IR AND IGNORE THE COMMENT CARDS	IDRO	30
C		IDRO	31
	120 CONTINUE	IDRO	32
	READ(IR,140)CARD1	IDRO	33
	140 FORMAT(20A4)	IDRO	34
	IF(EOF(IR))540,160	IDRO	35
	160 CONTINUE	IDRO	36
	DECODE(4,17),CARD1(1)CC,DUMMY	IDRO	37
	170 FORMAT(A2,A?)	IDRO	38
	IF(CC,EQ,MCR)GO TO 120	IDRO	39
	WRITE(JS,140)CARD1	IDRO	40
	DO 18 J=1,NID	IDRO	41
	IF(CARD1(1).EQ,CHEAD(J))GO TO 200	IDRO	42
	180 CONTINUE	IDRO	43
	GO TO 120	IDRO	44
C		IDRO	45
C	READ RESPONSE SPECIFICATION DATA AND ENCODE	IDRO	46
C	INTO SIMPLER RESPONSE SPECIFICATIONS	IDRO	47
C		IDRO	48
	200 CONTINUE	IDRO	49
	READ(IR,220)CARD2	IDRO	50
	220 FORMAT(MCA1)	IDRO	51
	I=0	IDRO	52
	240 CONTINUE	IDRO	53
	I=I+1	IDRO	54
	IF(I.GE,81)GO TO 200	IDRO	55
	IF(CARD2(I).EQ,HE)GO TO 520	IDRO	56
	IF(CARD2(I).EQ,HR)GO TO 240	IDRO	57
	IF(CARD2(I).EQ,MCOM)GO TO 240	IDRO	58
	IF(CARD2(I).EQ,HDAS)GO TO 400	IDRO	59
	IF(CARD2(I).EQ,MDOT)GO TO 520	IDRO	60
	IF((CARD2(I).NE,MR).AND.(CARD2(I).NE,MRP).AND.(CARD2(I).NE,MHI))	IDRO	61
	GO TO 200	IDRO	62
	I=I+1	IDRO	63
	IF(CARD2(I).EQ,HRP)GO TO 240	IDRO	64

Figure 60. Subroutine IDRO Program Listing

I=I+1	IDRO 65
IF(CARD2(I).EQ.HRP)GO TO 25	IDRO 66
I=I+2	IDRO 67
IF(CARD2(I).NE.HRP)GO TO 62	IDRO 68
ENCODE(4,250,CARD3(1))CARD2(I-4),CARD2(I-3),CARD2(I-2),CARD2(I-1)	IDRO 69
250 FORMAT(4A1)	IDRO 70
GO TO 300	IDRO 71
260 CONTINUE	IDRO 72
ENCODE(4,250,CARD3(1))CARD2(I-2),CARD2(I-1),HR.HR	IDRO 73
GO TO 300	IDRO 74
280 CONTINUE	IDRO 75
ENCODE(4,250,CARD3(1))CARD2(I-1),HR.HR.HR	IDRO 76
300 CONTINUE	IDRO 77
I=I+2	IDRO 78
IF(CARD2(I).EQ.HRP)GO TO 32	IDRO 79
I=I+1	IDRO 80
IF(CARD2(I).NE.HRP)GO TO 62	IDRO 81
ENCODE(2,310,CARD3(3))CARD2(I-2),CARD2(I-1)	IDRO 82
310 FORMAT(2A1)	IDRO 83
GO TO 340	IDRO 84
320 CONTINUE	IDRO 85
ENCODE(2,310,CARD3(3))HR,CARD2(I-1)	IDRO 86
340 CONTINUE	IDRO 87
DECODE(2,360,CARD3(3))NP	IDRO 88
360 FORMAT(I2)	IDRO 89
IF(CARD1(I).EQ.CHEAD(6))GO TO 380	IDRO 90
WRITE(JS,370)CARD3	IDRO 91
370 FORMAT(A4,A1,A2,73A1)	IDRO 92
GO TO 240	IDRO 93
380 CONTINUE	IDRO 94
WRITE(JS,390)(CARD3(I),I=1,4),(CARD2(I),I=9,80)	IDRO 95
390 FORMAT(A4,A1,A2,73A1)	IDRO 96
GO TO 200	IDRO 97
400 CONTINUE	IDRO 98
I=I+2	IDRO 99
IF(CARD2(I).NE.HRP)I=I+1	IDRO 100
IF(CARD2(I).NE.HRP)I=I+2	IDRO 101
IF(CARD2(I).NE.HRP)GO TO 62	IDRO 102
I=I+2	IDRO 103
IF(CARD2(I).EQ.HRP)GO TO 44	IDRO 104
I=I+1	IDRO 105
IF(CARD2(I).NE.HRP)GO TO 62	IDRO 106
ENCODE(2,310,CARD3(3))CARD2(I-2),CARD2(I-1)	IDRO 107
GO TO 460	IDRO 108
440 CONTINUE	IDRO 109
ENCODE(2,310,CARD3(3))HR,CARD2(I-1)	IDRO 110
460 CONTINUE	IDRO 111
DECODE(2,360,CARD3(3))NP	IDRO 112
IF(NN.LE.NP)GO TO 420	IDRO 113
NPP=NPP+1	IDRO 114
DO 500 J=NPP,NN	IDRO 115
ENCODE(2,480,CARD3(3))J	IDRO 116
480 FORMAT(I2)	IDRO 117
WRITE(JS,370)CARD3	IDRO 118
500 CONTINUE	IDRO 119
GO TO 240	IDRO 120
520 CONTINUE	IDRO 121
WRITE(JS,140)CARD4	IDRO 122
GO TO 120	IDRO 123
540 CONTINUE	IDRO 124
ENDIF JS	IDRO 125
REWIND IR	IDRO 126
REWIND JS	IDRO 127
C	IDRO 128
C	IDRO 129
C	IDRO 130
TRANSFER THE CARD IMAGES FROM FILE JS TO FILE IR	

Figure 60. Subroutine IDRO Program Listing (Continued)

560	CONTINUE	IDRO 131
	READ(JS,140)CARD1	IDRO 132
	IF(EOF(JS))600,590	IDRO 133
580	CONTINUE	IDRO 134
	WRITE(IR,140)CARD1	IDRO 135
	GO TO 560	IDRO 136
600	CONTINUE	IDRO 137
	ENDFILE IR	IDRO 138
	REWIND IR	IDRO 139
	REWIND JS	IDRO 140
	RETURN	IDRO 141
C		IDRO 142
C	PRINT ERROR MESSAGE	IDRO 143
C		IDRO 144
620	CONTINUE	IDRO 145
	WRITE(IW,640)CARD2	IDRO 146
640	FORMAT(1H1,/,/,1X,32HEXERRR IN REORGANIZING INPUT DATA,/,1X,80A1)	IDRO 147
	STOP 111	IDRO 148
	END	IDRO 149

Figure 60. Subroutine IDRO Program Listing (Concluded)

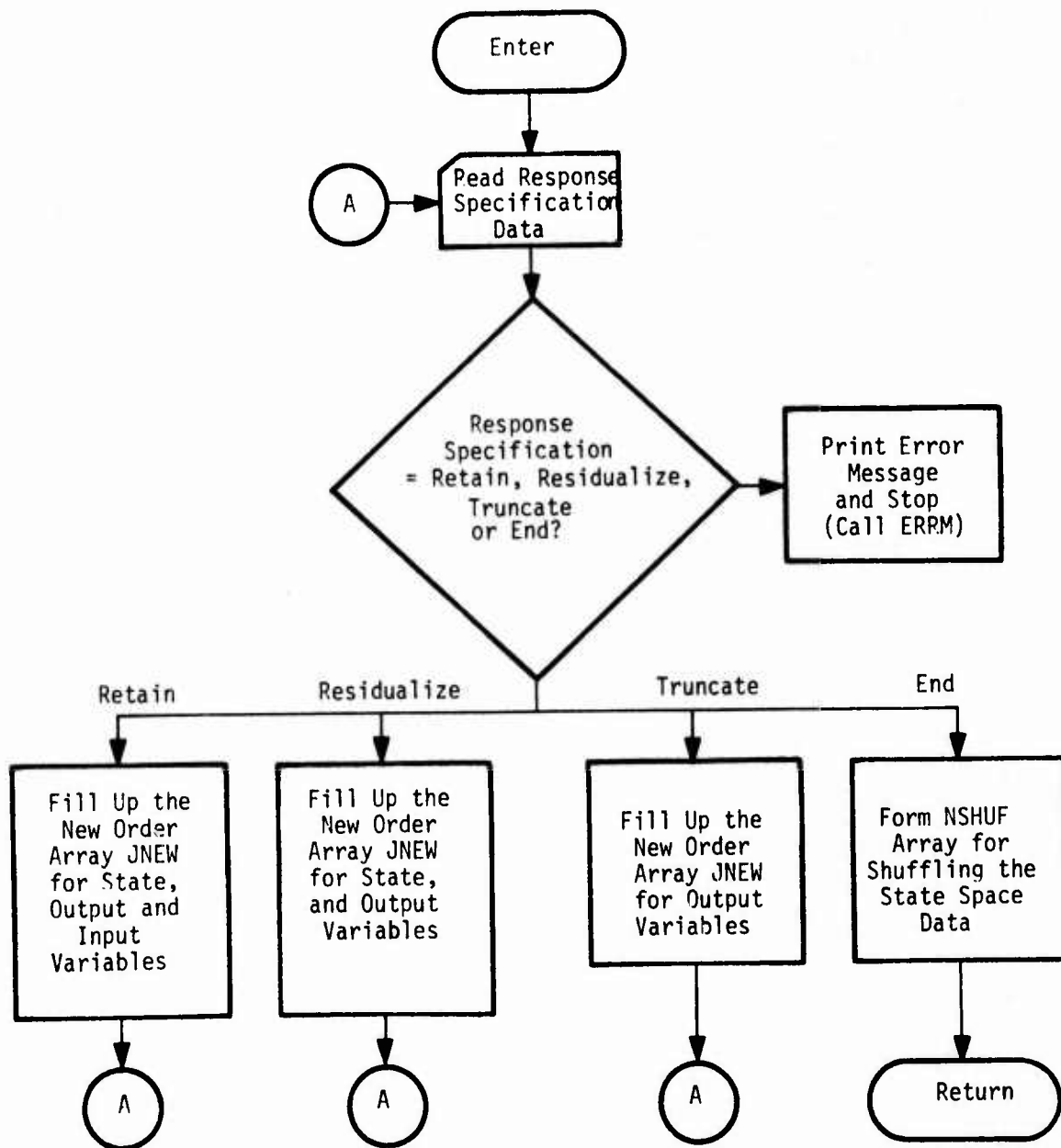


Figure 61. Subroutine RSDRD Flow Chart


```

SUBROUTINE RSDRD(JNEWS,NSHUF5,JNEW0,NSHUFO,JNEW1,NSHUF1,
INX,NR,NU,NXR,NXN,NXR,NRN,NRR,NRT,NUN,NXM,NRM,NUM,
ZIR,IW,IPOINT,IRS)
C
C   PURPOSE - TO READ REDUCTION AND SHUFFLING DATA
C   ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC
C   DATE WRITTEN - 1975
C
C   SUBPROGRAMS CALLED
C   DEBUG
C   ERRM
C
C   ARGUMENTS LIST
C   JNEWS    OUTPUT    ARRAY FOR NEW ORDER OF STATES
C   JNEW0    OUTPUT    ARRAY FOR NEW ORDER OF OUTPUTS
C   JNEW1    OUTPUT    ARRAY FOR NEW ORDER OF INPUTS
C   NSHUF5   OUTPUT    SHUFFLING ARRAY FOR STATE
C   NSHUFO   OUTPUT    SHUFFLING ARRAY FOR OUTPUT
C   NSHUF1   OUTPUT    SHUFFLING ARRAY FOR INPUT
C   NX       INPUT     NUMBER OF STATES
C   NR       INPUT     NUMBER OF OUTPUTS
C   NU       INPUT     NUMBER OF INPUTS
C   NXN      OUTPUT    NO OF STATES TO BE RETAINED AND RESIDUALI
C   NXN      OUTPUT    NUMBER OF REDUCED STATES
C   NRN      OUTPUT    NUMBER OF REDUCED OUTPUTS
C   NRN      OUTPUT    NO OF OUTPUTS TO BE RESIDUALIZED
C   NRT      OUTPUT    NO OF OUTPUTS TO BE TRUNCATED
C   NUN      OUTPUT    NUMBER OF REDUCED INPUTS
C   IR       INPUT     FILE NO FOR INPUT DATA BUFFER
C   IW       INPUT     FILE NO FOR LINE PRINTER
C   IPOINT   INPUT     PRINT CONTROL FLAG
C   IRS      OUTPUT    RESIDUALIZATION FLAG
C   OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
C   DIMENSION JNEWS(NXM),NSHUF5(NXM)
C   DIMENSION JNEW0(NRM),NSHUFO(NRM)
C   DIMENSION JNEW1(NUM),NSHUF1(NUM)
C   DIMENSION CARD(20)
C   DATA HENDB,HRETA,HTATE, HXBBB/4HEND ,4HRETA,4HTATE,4HX /
C   DATA HNPUR,HUBBB,HRESI,HBSTA/4HNPUR,4HU ,4HRESI,4HSTA/
C   DATA HRBBB,HTRUN,HUTPU,HSTAT/4HR ,4HTRUN,4HUTPU,4HSTAT/
C   IN=0
C   IRS=0
C   KS=0 * KR=0 $ KU=0
C   NXRN=0 $ NXN=0 $ NXR=0 $ NRN=0 $ NRR=0 $ NRT=0 $ NUN=0
C   IF(IPOINT.EQ.6)CALL DEBUG(1,4HRSDR,4HD ,5,0,IW)
120 CONTINUE
   READ(IR,140)CARD
140 FORMAT(20A4)
   IF(CARD(1).EQ.HENDB)GO TO 420
   IRS=1
   IF(CARD(1).NE.HRETA)GO TO 240
   IF((CARD(3).NE.HTATE).AND.(CARD(3).NE.HNPUR).AND.
1(CARD(3).NE.HUTPU))CALL ERRM(1,4HRSDR,4HD ,5,0,IW)
160 CONTINUE
C
C   READ SHUFFLE DATA FOR THE RETAINED SYSTEM VARIABLES
C
C   READ(IR,180)HD,N
180 FORMAT(A4,1X,I2)
   IF(HD.EQ.HENDB)GO TO 220
   IF(HD.NE.HXBBB)GO TO 200
C
C   FORM INEWS ARRAY FOR STATES
C
   KS=KS+1

```

```

RSDRD 2
RSDRD 3
RSDRD 4
RSDRD 5
RSDRD 6
RSDRD 7
RSDRD 8
RSDRD 9
RSDRD 10
RSDRD 11
RSDRD 12
RSDRD 13
RSDRD 14
RSDRD 15
RSDRD 16
RSDRD 17
RSDRD 18
RSDRD 19
RSDRD 20
RSDRD 21
RSDRD 22
RSDRD 23
RSDRD 24
RSDRD 25
RSDRD 26
RSDRD 27
RSDRD 28
RSDRD 29
RSDRD 30
RSDRD 31
RSDRD 32
RSDRD 33
RSDRD 34
RSDRD 35
RSDRD 36
RSDRD 37
RSDRD 38
RSDRD 39
RSDRD 40
RSDRD 41
RSDRD 42
RSDRD 43
RSDRD 44
RSDRD 45
RSDRD 46
RSDRD 47
RSDRD 48
RSDRD 49
RSDRD 50
RSDRD 51
RSDRD 52
RSDRD 53
RSDRD 54
RSDRD 55
RSDRD 56
RSDRD 57
RSDRD 58
RSDRD 59
RSDRD 60
RSDRD 61
RSDRD 62
RSDRD 63
RSDRD 64
RSDRD 65
RSDRD 66
RSDRD 67

```

Figure 62. Subroutine RSDRD Program Listing

	JNEWS(KS)=N	RSDRD 68
	GO TO 160	RSDRD 69
200	CONTINUE	RSDRD 70
	IF(MD,NE,MURR)GO TO 210	RSDRD 71
C		RSDRD 72
C	FORM JNEWI ARRAY FOR INPUTS	RSDRD 73
		RSDRD 74
	KU=KU-1	RSDRD 75
	JNEWI(KU)=N	RSDRD 76
	GO TO 160	RSDRD 77
210	CONTINUE	RSDRD 78
	IF(MD,NE,MRRR)CALL ERRM(2,4,HRSDR,4MD .5,0,IW)	RSDRD 79
C		RSDRD 80
C	FORM JNEWO ARRAY FOR OUTPUTS	RSDRD 81
		RSDRD 82
	KR=KR+1	RSDRD 83
	JNEWO(KR)=N	RSDRD 84
	GO TO 160	RSDRD 85
220	CONTINUE	RSDRD 86
	NXN=KS	RSDRD 87
	NUN=KI	RSDRD 88
	NRN=KR	RSDRD 89
	NXRN=NXN	RSDRD 90
	GO TO 120	RSDRD 91
240	CONTINUE	RSDRD 92
	IF(CARD(1),NE,MRES)GO TO 360	RSDRD 93
C		RSDRD 94
C	READ SHUFFLE DATA FOR THE RESIDUALIZED SYSTEM VARIABLES	RSDRD 95
		RSDRD 96
	IF(CARD(4),NE,MSTAT)CALL ERRM(3,4,HRSDR,4MD .5,0,IW)	RSDRD 97
260	CONTINUE	RSDRD 98
	READ(IR,100)MD,N	RSDRD 99
	IF(MD,EQ,MENDB)GO TO 350	RSDRD100
	IF(MD,NE,MXBBB)GO TO 280	RSDRD101
C		RSDRD102
C	FORM JNEWS ARRAY FOR STATES	RSDRD103
		RSDRD104
	KS=KS+1	RSDRD105
	JNEWS(KS)=N	RSDRD106
	GO TO 260	RSDRD107
280	CONTINUE	RSDRD108
	IF(MD,NE,MRBBB)CALL ERRM(4,4,HRSDR,4MD .5,0,IW)	RSDRD109
C		RSDRD110
C	FORM JNEWO ARRAY FOR OUTPUTS	RSDRD111
		RSDRD112
	KR=KR+1	RSDRD113
	JNEWO(KR)=N	RSDRD114
	GO TO 260	RSDRD115
350	CONTINUE	RSDRD116
	NXRN=KS	RSDRD117
	NXR=NXRN-NXN	RSDRD118
	NRR=KR	RSDRD119
	GO TO 120	RSDRD120
360	CONTINUE	RSDRD121
	IF(CARD(1),NE,MTRUN)CALL ERRM(5,4,HRSDR,4MD .5,0,IW)	RSDRD122
	IF(CARD(3),NE,MSTA)CALL ERRM(6,4,HRSDR,4MD .5,0,IW)	RSDRD123
C		RSDRD124
C	READ SHUFFLE DATA FOR THE TRUNCATED SYSTEM VARIABLES	RSDRD125
		RSDRD126
400	CONTINUE	RSDRD127
	READ(IR,100)MD,N	RSDRD128
	IF(MD,EQ,MENDB)GO TO 410	RSDRD129
	IF(MD,NE,MRBBB)CALL ERRM(7,4,HRSDR,4MD .5,0,IW)	RSDRD130
C		RSDRD131
C	FORM JNEWO ARRAY FOR OUTPUTS	RSDRD132
		RSDRD133

Figure 62. Subroutine RSDRD Program Listing (Continued)

	KR=KR+1	RSORD134
	JNEWO(KR)=N	RSORD135
	GO TO 400	RSORD136
410	CONTINUE	RSORD137
	NRT=KR-NRR	RSORD138
	GO TO 120	RSORD139
420	CONTINUE	RSORD140
	IF(I.RS.EQ.0)RETURN	RSORD141
	IF(IPRINT.EQ.6)CALL DEBUG(2,4MRSOR,4MD .5,0,IN)	RSORD142
	II=NXPXN	RSORD143
C		RSORD144
C	FORM SHUFFLE ARRAY FOR STATES	RSORD145
C		RSORD146
	DO 470 I=1,NX	RSORD147
	DO 430 J=1,NXN	RSORD148
	JJ=J	RSORD149
	IF(I.EQ.JNEWS(JJ))GO TO 460	RSORD150
430	CONTINUE	RSORD151
	IF(NXP.EQ.0)GO TO 450	RSORD152
	DO 440 J=1,NXR	RSORD153
	JJ=NXN+J	RSORD154
	IF(I.EQ.JNEWS(JJ))GO TO 460	RSORD155
440	CONTINUE	RSORD156
450	CONTINUE	RSORD157
	II=II+1	RSORD158
	NSHUF0(II)=I	RSORD159
	GO TO 470	RSORD160
460	CONTINUE	RSORD161
	NSHUF0(JJ)=I	RSORD162
470	CONTINUE	RSORD163
	IF(IPRINT.EQ.6)CALL DEBUG(3,4MRSOR,4MD .5,0,IN)	RSORD164
C		RSORD165
C	FORM SHUFFLE ARRAY FOR OUTPUTS	RSORD166
C		RSORD167
	II=NRN	RSORD168
	IF(NRN.LE.0)II=NRT+NRR	RSORD169
	IF(II.EQ.0)IN=1	RSORD170
	DO 570 I=1,NR	RSORD171
	IF(NRN.LE.0)GO TO 520	RSORD172
	DO 510 J=1,NRN	RSORD173
	JJ=J	RSORD174
	IF(I.EQ.JNEWO(JJ))GO TO 560	RSORD175
510	CONTINUE	RSORD176
	GO TO 550	RSORD177
520	CONTINUE	RSORD178
	IF(NRR.EQ.0)GO TO 535	RSORD179
	DO 530 J=1,NRR	RSORD180
	JJ=J	RSORD181
	IF(I.EQ.JNEWO(JJ))GO TO 560	RSORD182
530	CONTINUE	RSORD183
535	CONTINUE	RSORD184
	IF(NRT.EQ.0)GO TO 550	RSORD185
	DO 540 J=1,NRT	RSORD186
	JJ=NRP+J	RSORD187
	IF(I.EQ.JNEWO(JJ))GO TO 560	RSORD188
540	CONTINUE	RSORD189
550	CONTINUE	RSORD190
	II=II+1	RSORD191
	NSHUF0(II)=I	RSORD192
	GO TO 570	RSORD193
560	CONTINUE	RSORD194
	NSHUF0(JJ)=I	RSORD195
570	CONTINUE	RSORD196
	IF(IN.EQ.1)NRN=NR	RSORD197
	IN=0	RSORD198
	IF(IPRINT.EQ.6)CALL DEBUG(4,4MRSOR,4MD .5,0,IN)	RSORD199

Figure 62. Subroutine RSDRD Program Listing (Continued)

<pre> C C FORM SHUFFLE ARRAY FOR INPUTS C II=NUM IF (II.EQ.0) IN=1 DO 67 I=1,NU IF (NUM.LE.0) GO TO 640 DO 63 J=1,NUM JJ=J IF (I.EQ.JNEW(JJ)) GO TO 660 630 CONTINUE 640 CONTINUE II=II+1 NSHUFF(II)=I GO TO 670 650 CONTINUE NSHUFF(JJ)=I 670 CONTINUE IF (IN.EQ.1) NUN=NU IF (IP=INT.EQ.6) CALL DEBUG(5.4HRSOR.4HD .5,0,1W) IF (IP=INT.LT.6) GO TO 680 WRITE (IW,675) NX,NXN,NR,NRN,NU,NUN 675 FORMAT (1X,20(1P,1X)) WRITE (IW,675) JNEWS,JNFWO,JNEWJ WRITE (IW,675) NSHUFFS,NSHUFFO,NSHUFFI 680 CONTINUE RETURN END </pre>	<pre> RSDRD200 RSDRD201 RSDRD202 RSDRD203 RSDRD204 RSDRD205 RSDRD206 RSDRD207 RSDRD208 RSDRD209 RSDRD210 RSDRD211 RSDRD212 RSDRD213 RSDRD214 RSDRD215 RSDRD216 RSDRD217 RSDRD218 RSDRD219 RSDRD220 RSDRD221 RSDRD222 RSDRD223 RSDRD224 RSDRD225 RSDRD226 RSDRD227 </pre>
---	--

Figure 62. Subroutine RSDRD Program Listing (Concluded)

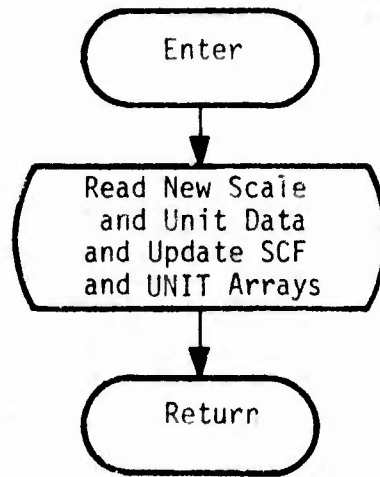


Figure 63. Subroutine SDRD Flow Chart

```

SUBROUTINE SDRD(SCFS,UNITNS,UNITS,SCFO,UNITNO,UNITO,SCFI,UNITNI,
IUNITI,NX,NR,NU,NXM,NRM,NUM,IR,IW,IPRINT)  SDRD 2
C SDRD 3
C SDRD 4
C PURPOSE - TO READ SCALE DATA SDRD 5
C ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC SDRD 6
C DATE WRITTEN - 1975 SDRD 7
C SDRD 8
C SUBPROGRAMS CALLED SDRD 9
C   DERUG SDRD 10
C   ERRM SDRD 11
C ARGUMENTS LIST SDRD 12
C   SCFS OUTPUT SCALING ARRAY FOR STATE SDRD 14
C   SCFO OUTPUT SCALING ARRAY FOR OUTPUT SDRD 15
C   SCFI OUTPUT SCALING ARRAY FOR INPUT SDRD 16
C   NX INPUT NUMBER OF STATES SDRD 17
C   NR INPUT NUMBER OF OUTPUTS SDRD 18
C   NU INPUT NUMBER OF INPUTS SDRD 19
C   IR INPUT FILE NO FOR INPUT DATA BUFFER SDRD 20
C   IW INPUT FILE NO FOR LINE PRINTER SDRD 21
C   IPRINT INPUT PRINT CONTROL FLAG SDRD 22
C OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM SDRD 23
C SDRD 24
C DIMENSION SCFS(NXM),UNITNS(NXM,4),UNITNS(NXM,4) SDRD 25
C DIMENSION SCFO(NRM),UNITO(NRM,4),UNITNO(NRM,4) SDRD 26
C DIMENSION SCFI(NUM),UNITI(NUM,4),UNITNI(NUM,4) SDRD 27
C DIMENSION UN(4),UNN(4) SDRD 28
C DIMENSION CARD(20) SDRD 29
C DATA MENDB,MXBRR,MRBRR,MUBRR/4MEND ,4MX ,4MR ,4MU / SDRD 30
C DATA MSCAL/4MSCAL/ SDRD 31
C ISC=0 SDRD 32
C IF(IPRINT.EQ.6)CALL DERUG(1,4MSDRD,4M ,5.0,IW) SDRD 33
C SDRD 34
C INITIALIZE SCF ARRAY SDRD 35
C SDRD 36
C DO 140 I=1,NXM SDRD 37
140 SCFS(I)=1.0 SDRD 38
C DO 160 I=1,NRM SDRD 39
160 SCFO(I)=1.0 SDRD 40
C DO 180 I=1,NUM SDRD 41
180 SCFI(I)=1.0 SDRD 42
C SDRD 43
C READ NEW SCALE AND UNIT DATA AND UPDATE SCF AND UNIT ARRAYS SDRD 44
C SDRD 45
C 260 CONTINUE SDRD 46
C READ(IR,20)MD,N,SC,(UN(J),J=1,4),(UNN(J),J=1,4) SDRD 47
280 FORMAT(A4,1X,I2,3X,E14.6,6X,4A4,4X,4A4) SDRD 48
C IF(IPRINT.EQ.6)CALL DERUG(2,4MSDRD,4M ,5.0,IW) SDRD 49
C IF(MD.EQ.MENDB)RETURN SDRD 50
C ISC=1 SDRD 51
C IF(MD.NE.MXBRR)GO TO 320 SDRD 52
C SDRD 53
C FOR STATES SDRD 54
C SDRD 55
C SCFS(N)=SC SDRD 56
C DO 300 J=1,4 SDRD 57
300 UNITNS(N,J)=UNN(J) SDRD 58
C UNITS(N,J)=UNN(J) SDRD 59
C GO TO 260 SDRD 60
320 CONTINUE SDRD 61
C IF(MD.NE.MRBRR)GO TO 360 SDRD 62
C SDRD 63
C FOR OUTPUTS SDRD 64
C SDRD 65
C SCFO(N)=SC SDRD 66
C DO 340 J=1,4 SDRD 67

```

Figure 64. Subroutine SDRD Program Listing

	UNITN(N,J)=UNN(J)	SDRD 68
340	UNITO(N,J)=UNN(J)	SDRD 69
	GO TO 260	SDRD 70
350	CONTINUE	SDRD 71
	IF (MO,NE,HURR)CALL FPRM(2.4HSDRD.4H .6.0.IW)	SDRD 72
C		SDRD 73
C	FOR I,POTS	SDRD 74
C		SDRD 75
	SCFI(I)=SC	SDRD 76
	DO 38 J=1,4	SDRD 77
	UNITNI(N,J)=UNN(J)	SDRD 78
380	UNITI(N,J)=UNN(J)	SDRD 79
	GO TO 260	SDRD 80
	END	SDRD 81

Figure 64. Subroutine SDRD Program Listing. (Concluded)

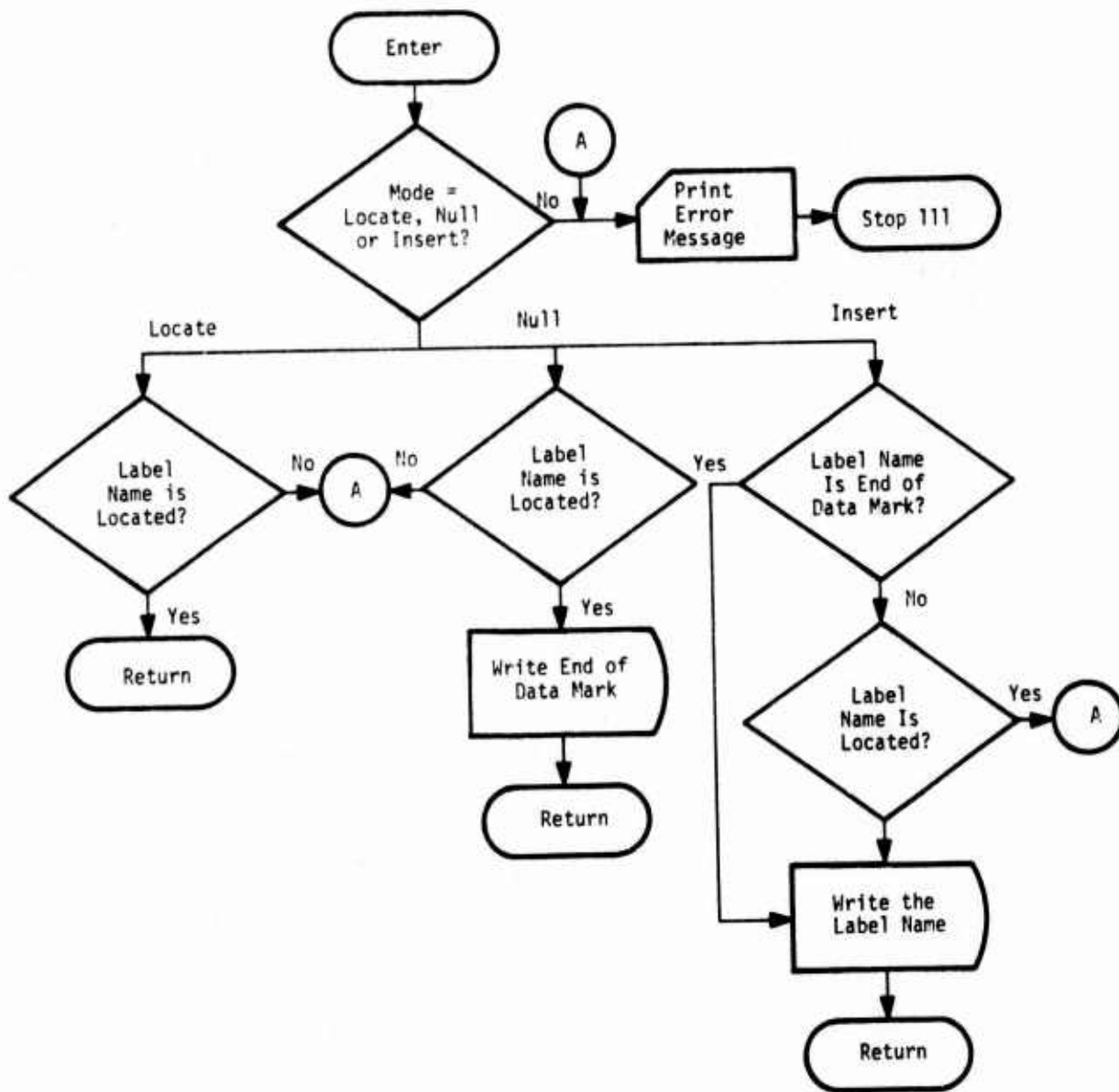


Figure 65. Subroutine FILE Flow Chart

	SUBROUTINE FILE(NFILE,MODE,NAME)	FILE 2
C		FILE 3
C	PURPOSE - TO POSITION THE DATA FILE	FILE 4
C	ANALYSIS - A F KONAR / J K MAMESH - THE MONEYWELL INC	FILE 5
C	DATE WRITTEN - 1975	FILE 6
C		FILE 7
C	ARGUMENT LIST	FILE 8
C	NFILE FILE NUMBER OF THE DISK FILE	FILE 9
C	MODE MODE PARAMETER (LOCATE,INSERT,NULL)	FILE 10
C	NAME LABEL NAME	FILE 11
C		FILE 12
	DIMENSION NAME(20),LABEL(20),LAST(20)	FILE 13
	INTEGER MDOLR	FILE 14
	DATA LOCATE,INSERT,NULL/4HL0CA,4MINSE,4MNULL/	FILE 15
	DATA MDOLR/4MSSSS/	FILE 16
	IW=9	FILE 17
	DO 10 I=1,20	FILE 18
100	LAST(I)=MDOLR	FILE 19
	IF(MODE.EQ.LOCATE)GO TO 140	FILE 20
	IF(MODE.EQ.INSERT)GO TO 120	FILE 21
	IF(MODE.EQ.NULL)GO TO 140	FILE 22
C		FILE 23
C	PRINT ERROR MESSAGE	FILE 24
C		FILE 25
	WRITE(IW,110)	FILE 26
110	FORMAT(1H1,/,/,)X,4SMODE OF OPERATION FOR DATA FILE NOT SPECIFIED	FILE 27
	STOP 111	FILE 28
C		FILE 29
C	CHECK IF END OF DATA MARK IS BEING INSERTED	FILE 30
C		FILE 31
120	CONTINUE	FILE 32
	DO 130 I=1,20	FILE 33
	IF(NAME(I).NE.LAST(I))GO TO 140	FILE 34
130	CONTINUE	FILE 35
C		FILE 36
C	WRITE END OF DATA MARK AND ALSO AN END OF FILE	FILE 37
C		FILE 38
	WRITE(NFILE)(NAME(I),I=1,20)	FILE 39
	ENDFILE NFILE	FILE 40
	REWIND NFILE	FILE 41
	RETURN	FILE 42
C		FILE 43
C	CHECK IF LABEL ON FILE MATCHES WITH NAME	FILE 44
C		FILE 45
140	CONTINUE	FILE 46
	REWIND NFILE	FILE 47
150	CONTINUE	FILE 48
	READ(NFILE)(LABEL(I),I=1,20)	FILE 49
	DO 160 I=1,20	FILE 50
	IF(LABEL(I).NE.NAME(I))GO TO 170	FILE 51
160	CONTINUE	FILE 52
	GO TO 220	FILE 53
C		FILE 54
C	CHECK IF LABEL IS THE END OF DATA MARK	FILE 55
C		FILE 56
170	CONTINUE	FILE 57
	DO 180 I=1,20	FILE 58
	IF(LABEL(I).NE.LAST(I))GO TO 200	FILE 59
180	CONTINUE	FILE 60
	IF(MODE.EQ.INSERT)GO TO 210	FILE 61
C		FILE 62
C	PRINT ERROR MESSAGE	FILE 63
C		FILE 64

Figure 66. Subroutine FILE Program Listing

	WRITE(IW,190)NAME,NFILE	FILE 65
190	FORMAT(IH1,/,/,1X,20A4,/,/,1X,20)CANNOT BE FOUND ON DATA FILE ,I2)	FILE 66
	STOP III	FILE 67
C		FILE 68
C	POSITION THE FILE TO THE BEGINNING OF NEXT RECORD	FILE 69
C		FILE 70
200	CONTINUE	FILE 71
	READ(NFILE)	FILE 72
	GO TO 150	FILE 73
C		FILE 74
C	WRITE NAME ON THE FILE	FILE 75
C		FILE 76
210	BACKSPACE NFILE	FILE 77
	WRITE(NFILE)(NAME(I),I=1,20)	FILE 78
	RETURN	FILE 79
220	CONTINUE	FILE 80
	IF(MODE.EQ.INSERT)GO TO 230	FILE 81
	IF(MODE.EQ.LOCATE)RETURN	FILE 82
C		FILE 83
C	WRITE END OF DATA MARK	FILE 84
C		FILE 85
	BACKSPACE NFILE	FILE 86
	WRITE(NFILE)(LAST(I),I=1,20)	FILE 87
	RETURN	FILE 88
C		FILE 89
C	PRINT ERROR MESSAGE	FILE 90
C		FILE 91
230	CONTINUE	FILE 92
	WRITE(IW,240)NAME,NFILE	FILE 93
240	FORMAT(IH1,/,/,1X,20A4,/,/,1X,20)ALREADY ON DATA FILE ,I2)	FILE 94
	STOP III	FILE 95
	END	FILE 96

Figure 66. Subroutine FILE Program Listing (Concluded)

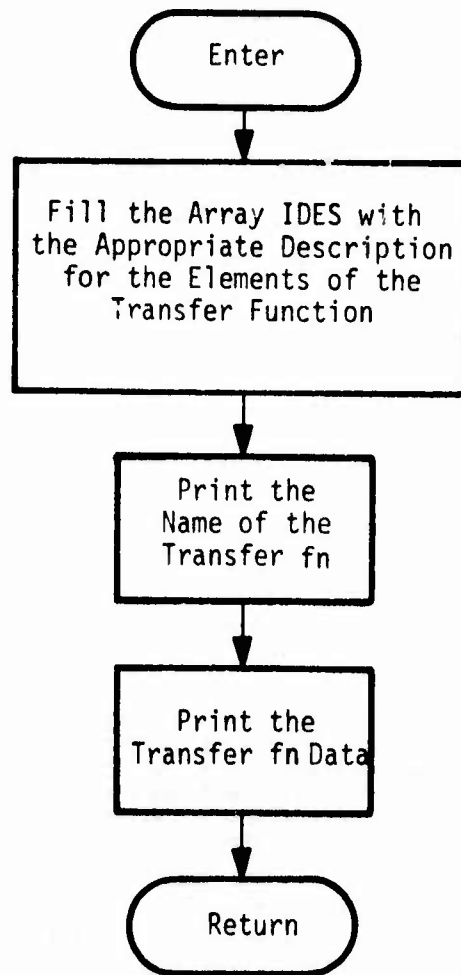


Figure 67. Subroutine TPR Flow Chart

```

SUBROUTINE TPR(H,NF,NFM,NAME,T,IW)                                TPR 2
C                                                                    TPR 3
C PURPOSE - TO PRINT TRANSFER FUNCTION DATA                      TPR 4
C ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL TNC          TPR 5
C DATE WRITTEN - 1975                                           TPR 6
C                                                                    TPR 7
C ARGUMENTS LIST                                                TPR 8
C   H      INPUT      TRANSFER FUNCTION                          TPR 9
C   NF     INPUT      NO OF ELEMENTS OF THE TRANSFER FN         TPR 10
C   NFM    INPUT      MAXIMUM NO OF ELEMENTS OF THE TRANSFER FN TPR 11
C   NAME   INPUT      NAME OF THE TRANSFER FN                    TPR 12
C   T      INPUT      SAMPLE TIME                                 TPR 13
C   IW     INPUT      FILE NO FOR LINE PRINTER                   TPR 14
C                                                                    TPR 15
C LABELLED COMMON LIST                                          TPR 16
C   IDES   LOCAL      ARRAY FOR DESCRIPTION OF THE TRANSFER FN  TPR 17
C   MI     LOCAL      CONSTANT                                    TPR 18
C   I      LOCAL      INDEX                                       TPR 19
C   K      LOCAL      INDEX                                       TPR 20
C                                                                    TPR 21
C   DIMENSION M(2,NFM),NAME(3)                                  TPR 22
C   COMMON /SC1/ IDES(6,3),MI,1,K                               TPR 23
C                                                                    TPR 24
C CHECK FOR DIMENSION ERROR                                     TPR 25
C                                                                    TPR 26
C IF((NFM.NE.6).OR.(NE.GT.NFM))GO TO 260                       TPR 27
C                                                                    TPR 28
C FILL THE ARRAY IDES WITH THE APPROPRIATE DESCRIPTION FOR THE  TPR 29
C ELEMENTS OF THE TRANSFER FUNCTION                             TPR 30
C                                                                    TPR 31
C IF(T.NE.0.0)GO TO 120                                         TPR 32
C IDES(1,1)=4HS**5 $ IDES(1,2)=4H TER $ IDES(1,3)=4HM         TPR 33
C IDES(2,1)=4HS**4 $ IDES(2,2)=4H TER $ IDES(2,3)=4HM         TPR 34
C IDES(3,1)=4HS**3 $ IDES(3,2)=4H TER $ IDES(3,3)=4HM         TPR 35
C IDES(4,1)=4HS**2 $ IDES(4,2)=4H TER $ IDES(4,3)=4HM         TPR 36
C IDES(5,1)=4HS**1 $ IDES(5,2)=4H TER $ IDES(5,3)=4HM         TPR 37
C IDES(6,1)=4HS**0 $ IDES(6,2)=4H TER $ IDES(6,3)=4HM         TPR 38
C GO TO 140                                                     TPR 39
120 CONTINUE                                                    TPR 40
C IDES(1,1)=4HZ**5 $ IDES(1,2)=4H TER $ IDES(1,3)=4HM         TPR 41
C IDES(2,1)=4HZ**4 $ IDES(2,2)=4H TER $ IDES(2,3)=4HM         TPR 42
C IDES(3,1)=4HZ**3 $ IDES(3,2)=4H TER $ IDES(3,3)=4HM         TPR 43
C IDES(4,1)=4HZ**2 $ IDES(4,2)=4H TER $ IDES(4,3)=4HM         TPR 44
C IDES(5,1)=4HZ**1 $ IDES(5,2)=4H TER $ IDES(5,3)=4HM         TPR 45
C IDES(6,1)=4HZ**0 $ IDES(6,2)=4H TER $ IDES(6,3)=4HM         TPR 46
140 CONTINUE                                                    TPR 47
C                                                                    TPR 48
C PRINT THE NAME OF THE TRANSFER FN                              TPR 49
C                                                                    TPR 50
C IF(T.FD.0.0)WRITE(IW,160)NAME                                 TPR 51
160 FORMAT(//,1X,3A4)                                           TPR 52
C IF(T.NE.0.0)WRITE(IW,180)NAME,T                               TPR 53
180 FORMAT(//,1X,3A4,3H(T=,G14.6,1H))                             TPR 54
C                                                                    TPR 55
C PRINT THE TRANSFER FN                                          TPR 56
C                                                                    TPR 57
C MI=6-NE+1                                                    TPR 58
C WRITE(IW,200)((IDES(I,K),K=1,3),I=MI,6)                       TPR 59
200 FORMAT(//,18X,5(3A4,2X))                                     TPR 60
C WRITE(IW,220)(M(1,I),I=1,NE)                                  TPR 61
220 FORMAT(//,1X,9HNUMERATOR,6X,5G14.6)                          TPR 62
C WRITE(IW,240)(M(2,I),I=1,NE)                                  TPR 63
240 FORMAT(//,1X,11HDENOMINATOR,6X,5G14.6)                       TPR 64

```

Figure 68. Subroutine TPR Program Listing

	RETURN	TPR	65
C		TPR	66
C	PRINT ERROR MESSAGE	TPR	67
C		TPR	68
250	CONTINUE	TPR	69
	WRITE(IW,280)	TPR	70
280	FORMAT(1H1,/,1X,42HDIMENSION ERROR DETECTED BY SUBROUTINE TPR)	TPR	71
	STOP 111	TPR	72
	END	TPR	73

Figure 68. Subroutine TPR Program Listing (Concluded)

C	SUBROUTINE HPR(CARD,IW)	HPR	2
C	PURPOSE - TO PRINT HEADING FOR SYSTEM LABEL NAMES	HPR	3
C	ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC	HPR	4
C	DATE WRITTEN - 1975	HPR	5
C		HPR	6
C	ARGUMENTS LIST	HPR	7
C	CARD INPUT SYSTEM LABEL NAME	HPR	8
C	IW INPUT FILE NO FOR LINE PRINTER	HPR	9
C		HPR	10
	DIMENSION CARD(20)	HPR	11
	INTEGER CARD	HPR	12
	WRITE(IW,120)	HPR	13
120	FORMAT(1H1,///,2GX,RR(1H*))	HPR	14
	WRITE(IW,140)	HPR	15
140	FORMAT(20X,1H*,R6X,1H*)	HPR	16
	WRITE(IW,160)CARD	HPR	17
160	FORMAT(20X,1H*,2X,2CA6,4X,1H*)	HPR	18
	WRITE(IW,140)	HPR	19
	WRITE(IW,180)	HPR	20
180	FORMAT(20X,RR(1H*))	HPR	21
	RETURN	HPR	22
		HPR	23

Figure 69. Subroutine HPR Program Listing

	SUBROUTINE IDPR(IR,IW)	IDPR 2
C		IDPR 3
C	PURPOSE - TO PRINT INPUT DATA	IDPR 4
C	ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC	IDPR 5
C	DATE WRITTEN - 1975	IDPR 6
C		IDPR 7
C	ARGUMENTS LIST	IDPR 8
C	IR INPUT FILE NO FOR INPUT DATA BUFFER	IDPR 9
C	IW INPUT FILE NO FOR LINE PRINTED	IDPR 10
C		IDPR 11
C	DIMENSION CARD(20)	IDPR 12
	REWIND IR	IDPR 13
120	CONTINUE	IDPR 14
	READ(IR,143)CARD	IDPR 15
140	FORMAT(20A4)	IDPR 16
	IF(EOF(IR))200,160	IDPR 17
160	CONTINUE	IDPR 18
	WRITE(IW,180)CARD	IDPR 19
180	FORMAT(1X,20A4)	IDPR 20
	GO TO 120	IDPR 21
200	CONTINUE	IDPR 22
	REWIND IR	IDPR 23
	RETURN	IDPR 24
	END	IDPR 25

Figure 70. Subroutine IDPR Program Listing

C	SUBROUTINE MPRS(A,MX,N,NC,T,ITITLE)	MPRS	2
C	PURPOSE - TO PRINT MATRIX DATA	MPRS	3
C	ANALYST - A F KONAR / J K RAHESH - THE HONEYWELL INC	MPRS	4
C	DATE - PITTS - 1975	MPRS	5
C		MPRS	6
C	ARGUMENTS LIST	MPRS	7
C	A INPUT MATRIX DATA	MPRS	8
C	MX INPUT MAXIMUM NO OF ROWS	MPRS	9
C	NC INPUT MAXIMUM NO OF COLUMNS	MPRS	10
C	NR INPUT NUMBER OF ROWS	MPRS	11
C	NC INPUT NUMBER OF COLUMNS	MPRS	12
C	T INPUT SAMPLE TIME	MPRS	13
C	ITITLE INPUT TITLE OR NAME OF THE MATRIX	MPRS	14
C		MPRS	15
C	DIMENSION A(MX,NC)	MPRS	16
C		MPRS	17
C	BEGINNING OF COLUMN SIZE LOOP	MPRS	18
C		MPRS	19
C	JC =	MPRS	20
C	100 CONTINUE	MPRS	21
C	IC = IC + 1	MPRS	22
C	JC = IC + 1	MPRS	23
C	IF (JC .GT. NC) JC = NC	MPRS	24
C		MPRS	25
C	BEGINNING OF ROW SIZE LOOP	MPRS	26
C		MPRS	27
C	JR=0	MPRS	28
C	150 CONTINUE	MPRS	29
C		MPRS	30
C	PRINT MATRIX NAME AND SIZE	MPRS	31
C		MPRS	32
C	IF (T .EQ. 0.) WRITE (9,8) ITITLE, NR, NC	MPRS	33
C	8) FORMAT (//8H MATRIX ,A4.16X.7H SIZE = ,I2.3H X ,I2)	MPRS	34
C	IF (T .NE. 0.) WRITE (9,9) ITITLE, T, NR, NC	MPRS	35
C	9) FORMAT (//8H MATRIX ,A4.3H(T=,E10.4.1H).2X.7H SIZE = ,I2.3H X ,I2)	MPRS	36
C		MPRS	37
C	PRINT COLUMN INDEX	MPRS	38
C		MPRS	39
C	WRITE (9, 160) (K, K = IC, JC)	MPRS	40
C	160 FORMAT (//8X.1H(2X.13.7H-COLUMN))	MPRS	41
C	WRITE (9,170)	MPRS	42
C	170 FORMAT (/)	MPRS	43
C	IR=JR+1	MPRS	44
C	JR=JR+1	MPRS	45
C	IF (JR .GT. NR) JR=NR	MPRS	46
C	DO 18 I=IR, JR	MPRS	47
C		MPRS	48
C		MPRS	49
C	PRINT ROW INDEX AND MATRIX DATA	MPRS	50
C		MPRS	51
C	WRITE (9, 190) I, (A(I,J), J = IC, NC)	MPRS	52
C	190 FORMAT (1X.13.4H-ROW,1X.15(1P.4))	MPRS	53
C	180 CONTINUE	MPRS	54
C		MPRS	55
C	END OF ROW SIZE LOOP	MPRS	56
C		MPRS	57
C	IF (JR .LT. NR) GO TO 150	MPRS	58
C		MPRS	59
C	END OF COLUMN SIZE LOOP	MPRS	60
C		MPRS	61
C	IF (JC .LT. NC) GO TO 100	MPRS	62
C		MPRS	63
C	RETURN TO CALLING PROGRAM	MPRS	64

Figure 71. Subroutine MPRS Program Listing

C
RETURN
END

MPRS 65
MPRS 66
MPRS 67

Figure 71. Subroutine MPRS Program Listing (Concluded)

C	SUBROUTINE MPRS1(A,NRM,NCM,NR,NC,MHEAD)	MPRS1 2
C	PURPOSE - TO PRINT LSA MATRIX DATA	MPRS1 3
C	ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC	MPRS1 4
C	DATE WRITTEN - 1975	MPRS1 5
C		MPRS1 6
C	ARGUMENTS LIST	MPRS1 7
C	A INPUT MATRIX DATA	MPRS1 8
C	NRM INPUT MAXIMUM NUMBER OF ROWS	MPRS1 9
C	NCM INPUT MAXIMUM NUMBER OF COLUMNS	MPRS1 10
C	NR INPUT NUMBER OF ROWS	MPRS1 11
C	NC INPUT NUMBER OF COLUMNS	MPRS1 12
C	MHEAD INPUT MATRIX TITLE OR NAME	MPRS1 13
C		MPRS1 14
C	DIMENSION A(NRM,NCM)	MPRS1 15
C	COMMON /INPUT/ IP,IV,IPRINT,JN,JQ,INSERT,LOCATE,NULL,MARK(20)	MPRS1 16
C	IF((IPRINT.NE.7).AND.(IPRINT.LT.5))RETURN	MPRS1 17
C		MPRS1 18
C	WRITE NAME AND SIZE OF THE MATRIX	MPRS1 19
C		MPRS1 20
C	WRITE(IW,80)MHEAD,NR,NC	MPRS1 21
C	80 FORMAT(//,12X,A10,16HMATRIX , SIZE = ,12,3H X ,12)	MPRS1 22
C	JC=0	MPRS1 23
C	100 JC=JC+1	MPRS1 24
C	JC=JC+7	MPRS1 25
C	IF(JC.GT.NC)JC=NC	MPRS1 26
C	KC=JC-IC+1	MPRS1 27
C		MPRS1 28
C	WRITE COLUMN HEADINGS	MPRS1 29
C		MPRS1 30
C	150 WRITE(IW,160)(K,K=IC,JC)	MPRS1 31
C	160 FORMAT(//,8X,7(2X,13,7H-COLUMN,3X))	MPRS1 32
C	WRITE(IW,170)	MPRS1 33
C	170 FORMAT(//)	MPRS1 34
C	DO 18 I=1,NR	MPRS1 35
C		MPRS1 36
C	WRITE ROW HEADINGS	MPRS1 37
C		MPRS1 38
C	180 WRITE(IW,190)I,(A(I,J),J=IC,JC)	MPRS1 39
C	190 FORMAT(1X,13,4H-ROW,1X,7(E15.7))	MPRS1 40
C	320 IF(JC.LT.NC)GO TO 150	MPRS1 41
C	WRITE(IW,170)	MPRS1 42
C	RETURN	MPRS1 43
C	END	MPRS1 44
		MPRS1 45

Figure 72. Subroutine MPRS1 Program Listing

	SUBROUTINE ZERO(A,NRM,NCM)	ZERO	2
C		ZERO	3
C	PURPOSE - TO ZERO THE ELEMENTS OF A MATRIX	ZERO	4
C	ANALYSIS - A F KONAR / J K MAMESH - THE HONEYWELL INC	ZERO	5
C	DATE WRITTEN - 1975	ZERO	6
C		ZERO	7
C	ARGUMENTS LIST	ZERO	8
C	A OUTPUT MATRIX DATA	ZERO	9
C	NRM INPUT MAXIMUM NUMBER OF ROWS	ZERO	10
C	NCM INPUT MAXIMUM NUMBER OF COLUMNS	ZERO	11
C		ZERO	12
	DIMENSION A(NRM,NCM)	ZERO	13
	DO 120 I=1,NRM	ZERO	14
	DO 120 J=1,NCM	ZERO	15
120	A(I,J)=0.0	ZERO	16
	RETURN	ZERO	17
	END	ZERO	18

Figure 73. Subroutine ZERO Program Listing

	SUBROUTINE INPT(A,II,JJ)	INPT	2
C		INPT	3
C	PURPOSE - TO READ NON ZERO ELEMENTS OF A MATRIX	INPT	4
C		INPT	5
C	ARGUMENTS LIST	INPT	6
C	A OUTPUT MATRIX DATA	INPT	7
C	II INPUT MAXIMUM NO OF ROWS	INPT	8
C	JJ INPUT MAXIMUM NO OF COLUMNS	INPT	9
C		INPT	10
	DIMENSION A(II,JJ),ID(5),JD(5),YD(5)	INPT	11
1	READ(5,2)(ID(I),JD(I),YD(I),I=1,5)	INPT	12
2	FORMAT(5(2I2,E12.5))	INPT	13
	IF(ID(1))10,10,3	INPT	14
3	DO 6 I=1,5	INPT	15
	IF(ID(I))4,1,4	INPT	16
4	CONTINUE	INPT	17
	I=ID(I)	INPT	18
	J=JD(I)	INPT	19
	A(I,J)=YD(I)	INPT	20
6	CONTINUE	INPT	21
	GO TO 1	INPT	22
10	CONTINUE	INPT	23
	RETURN	INPT	24
	END	INPT	25

Figure 74. Subroutine INPT Program Listing

```

C      SUBROUTINE INPT1(A,NRM,NCM,NR,NC,IR)
C
C      PURPOSE - TO READ ISA MATRIX DATA
C      ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C      DATE WRITTEN - 1975
C
C      ARGUMENTS LIST
C      A      OUTPUT      MATRIX DATA
C      NRM    INPUT      MAXIMUM NUMBER OF ROWS
C      NCM    INPUT      MAXIMUM NUMBER OF COLUMNS
C      NR     INPUT      NUMBER OF ROWS
C      NC     INPUT      NUMBER OF COLUMNS
C      IR     INPUT      FILE NO FOR INPUT DATA BUFFER
C
C      DIMENSION A(NRM,NCM)
C      READ(16,120)((A(I,J),J=1,NC),I=1,NR)
120  FORMAT(6G10.3)
C      RETURN
C      END

```

```

INPT1  2
INPT1  3
INPT1  4
INPT1  5
INPT1  6
INPT1  7
INPT1  8
INPT1  9
INPT1 10
INPT1 11
INPT1 12
INPT1 13
INPT1 14
INPT1 15
INPT1 16
INPT1 17
INPT1 18
INPT1 19
INPT1 20

```

Figure 75. Subroutine INPT1 Program Listing

C	SUBROUTINE DEBUG(N,A1,A2,N1,N2,IW)	DEBUG 2
C		DEBUG 3
C	PURPOSE - TO PRINT DEBUGGING MESSAGE	DEBUG 4
C	ANALYSTS - A F KONAR / J K NAGESH - THE HONEYWELL INC	DEBUG 5
C	DATE WRITTEN - 1975	DEBUG 6
C		DEBUG 7
C	ARGUMENTS LIST	DEBUG 8
C	N INPUT POSITION OF EXECUTION	DEBUG 9
C	A1 INPUT NAME OF THE SUBROUTINE	DEBUG 10
C	A2 INPUT NAME OF THE SUBROUTINE (CONTINUED)	DEBUG 11
C	N1 INPUT PRIMARY OVERLAY NO	DEBUG 12
C	N2 INPUT SECONDARY OVERLAY NO	DEBUG 13
C	IW INPUT FILE NO FOR LINE PRINTER	DEBUG 14
C		DEBUG 15
	WRITE(IW,12)N,A1,A2,N1,N2	DEBUG 16
120	FORMAT(//,1X,27HEXECUTION ENTERED POSITION ,12,1X,	DEBUG 17
	111HSUBROUTINE ,2A4,1X,12HIN OVERLAY (,11,1H,,11,1H))	DEBUG 18
	RETURN	DEBUG 19
	END	DEBUG 20

Figure 76. Subroutine DEBUG Program Listing

	SUBROUTINE ERRM(N,A1,A2,N1,N2,IW)	ERRM	2
		ERRM	3
C	PURPOSE - TO PRINT ERROR MESSEGE	ERRM	4
C	ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC	ERRM	5
C	DATE WRITTEN - 1975	ERRM	6
C		ERRM	7
C	ARGUMENTS LIST	ERRM	8
C	N INPUT POSITION OF EXECUTION	ERRM	9
C	A1 INPUT NAME OF THE SUBROUTINE	ERRM	10
C	A2 INPUT NAME OF THE SUBROUTINE (CONTINUED)	ERRM	11
C	N1 INPUT PRIMARY OVERLAY NO	ERRM	12
C	N2 INPUT SECONDARY OVERLAY NO	ERRM	13
C	IW INPUT FILE NO FOR LINE PRINTER	ERRM	14
C		ERRM	15
	WRITE(IW,120)N,A1,A2,N1,N2	ERRM	16
120	FORMAT(1H1,/,1X,27HERROR DETECTED AT POSITION ,12,1X,	ERRM	17
111	SUBROUTINE ,2A4,1X,17MIN OVERLAY (.11,1H,,11,1H))	ERRM	18
	STOP 111	ERRM	19
	END	ERRM	20

Figure 77. Subroutine ERRM Program Listing

```

C      SUBROUTINE DERRM(M1,M2,M3,M4,MS1,MS2,MS3,MS4,N1,N2,A1,A2,IW)      DERRM  2
C      PURPOSE - TO PRINT ERROR MESSEGE WHEN DIMENSIONS FOR          DERRM  3
C      SCRATCH ARRAYS IS NOT SUFFICIENT                               DERRM  4
C      ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC         DERRM  5
C      DATE WRITTEN - 1975                                           DERRM  6
C      ARGUMENTS LIST                                               DERRM  7
C      M1      INPUT      ACTUAL DIMENSION FOR SCRATCH ARRAY S1     DERRM  8
C      M2      INPUT      ACTUAL DIMENSION FOR SCRATCH ARRAY S2     DERRM  9
C      M3      INPUT      ACTUAL DIMENSION FOR SCRATCH ARRAY S3     DERRM 10
C      M4      INPUT      ACTUAL DIMENSION FOR SCRATCH ARRAY S4     DERRM 11
C      MS1     INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S1    DERRM 12
C      MS2     INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S2    DERRM 13
C      MS3     INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S3    DERRM 14
C      MS4     INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S4    DERRM 15
C      N1     INPUT      PRIMARY OVERLAY NO                          DERRM 16
C      N2     INPUT      SECONDARY OVERLAY NO                        DERRM 17
C      A1     INPUT      NAME OF THE SUBROUTINE                      DERRM 18
C      A2     INPUT      NAME OF THE SUBROUTINE (CONTINUED)         DERRM 19
C      IW     INPUT      FILE NO FOR LINE PRINTER                   DERRM 20
C                                                                    DERRM 21
C                                                                    DERRM 22
C                                                                    DERRM 23
C      DIMENSION M(4),MS(4)                                         DERRM 24
C      M(1)=M1 & M(2)=M2 & M(3)=M3 & M(4)=M4                       DERRM 25
C      MS(1)=MS1 & MS(2)=MS2 & MS(3)=MS3 & MS(4)=MS4             DERRM 26
C      WRITE(IW,240)N1,N2,A1,A2                                       DERRM 27
240  FORMAT(1H1,/,/,1X,2HDIMENSION ERROR IN OVERLAY (,11,1H,/,11,1H), DERRM 28
      113HIN SUBROUTINE,2X,2A4)                                       DERRM 29
      DO 260 I=1,4                                                    DERRM 30
      WRITE(IW,250)I,MS(I),M(I)                                       DERRM 31
250  FORMAT(//,1X,15HDIMENSION FOR S,11,2X,7HACTUAL=,15,2X,      DERRM 32
      19HREQUIRED=,15)                                               DERRM 33
260  CONTINUE                                                         DERRM 34
      STOP 111                                                         DERRM 35
      END                                                             DERRM 36

```

Figure 78. Subroutine DERRM Program Listing


```

SUBROUTINE DERRMS(M1,M2,M3,M4,MS1,MS2,MS3,MS4,N1,N2,A1,A2,IW) DERRMS 2
C DERRMS 3
C PURPOSE - TO PRINT ERROR MESSAGE WHEN SYSTEM DIMENSION DERRMS 4
C ARE NOT SUFFICIENT DERRMS 5
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC DERRMS 6
C DATE WRITTEN - 1976 DERRMS 7
C DERRMS 8
C ARGUMENTS LIST DERRMS 9
C M1 INPUT ACTUAL DIMENSION DERRMS10
C M2 INPUT ACTUAL DIMENSION DERRMS11
C M3 INPUT ACTUAL DIMENSION DERRMS12
C M4 INPUT ACTUAL DIMENSION DERRMS13
C MS1 INPUT MAXIMUM DIMENSION DERRMS14
C MS2 INPUT MAXIMUM DIMENSION DERRMS15
C MS3 INPUT MAXIMUM DIMENSION DERRMS16
C MS4 INPUT MAXIMUM DIMENSION DERRMS17
C N1 INPUT PRIMARY OVERLAY NO DERRMS18
C N2 INPUT SECONDARY OVERLAY NO DERRMS19
C A1 INPUT NAME OF THE SUBROUTINE DERRMS20
C A2 INPUT NAME OF THE SUBROUTINE (CONTINUED) DERRMS21
C IW INPUT FILE NO FOR LINE PRINTER DERRMS22
C DERRMS23
C DIMENSION M(4),MS(4),A(4) DERRMS24
C DATA A/4HNYM,4HNRM,4HNUM,4HNYM / DERRMS25
C M(1)=M1 $ M(2)=M2 $ M(3)=M3 $ M(4)=M4 DERRMS26
C MS(1)=MS1 $ MS(2)=MS2 $ MS(3)=MS3 $ MS(4)=MS4 DERRMS27
C WRITE(IW,240)N1,N2,A1,A2 DERRMS28
240 FORMAT(1H) //,1X,2PHDIMENSION ERROR IN OVERLAY (,1I,1H,,1I,1H), DERRMS29
113MIN SUBROUTINE,2X,2A4) DERRMS30
C DO 26, I=1,4 DERRMS31
C WRITE(IW,250)A(I),MS(I),M(I) DERRMS32
250 FORMAT(//,1X,10HD)MENSION ,A4,2X,7HACTUAL=,15,2X, DERRMS33
19HREQUIRED=,15) DERRMS34
260 CONTINUE DERRMS35
C STOP 111 DERRMS36
C END DERRMS37

```

Figure 79. Subroutine DERRMS Program Listing

```

SUBROUTINE SHIFT(NN,VN,DES,UNIT,NNN,VNN,DESN,UNITN,N,NM,IW,IPRINT)SHIFT 2
C                                     SHIFT 3
C PURPOSE - TO SHIFT CONTENTS OF OLD ARRAYS NN,VN,DES,UNIT  SHIFT 4
C INTO NEW ARRAYS NNN,VNN,DESN,UNITN  SHIFT 5
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC  SHIFT 6
C DATE WRITTEN - 1975  SHIFT 7
C                                     SHIFT 8
C SUBPROGRAMS CALLED  SHIFT 9
C   DEBUG  SHIFT 10
C                                     SHIFT 11
C ARGUMENTS LIST  SHIFT 12
C   NN      INPUT  OLD NUMBER ARRAY  SHIFT 13
C   VN      INPUT  OLD VARIABLE NAME ARRAY  SHIFT 14
C   DES     INPUT  OLD DESCRIPTION ARRAY  SHIFT 15
C   UNIT    INPUT  OLD UNIT ARRAY  SHIFT 16
C   NNN     OUTPUT NEW NUMBER ARRAY  SHIFT 17
C   VNN     OUTPUT NEW VARIABLE NAME ARRAY  SHIFT 18
C   DESN    OUTPUT NEW DESCRIPTION ARRAY  SHIFT 19
C   UNITN   OUTPUT NEW UNIT ARRAY  SHIFT 20
C   N       INPUT  NUMBER OF SYSTEM VARIABLES  SHIFT 21
C   NM      INPUT  MAX NO OF SYSTEM VARIABLES  SHIFT 22
C   IW      INPUT  FILE NO FOR LINE PRINTED  SHIFT 23
C   IPRINT  INPUT  PRINT CONTROL FLAG  SHIFT 24
C                                     SHIFT 25
C   DIMENSION NN(NM),VN(NM,2),DES(NM,10),UNIT(NM,4)  SHIFT 26
C   DIMENSION NNN(NM),VNN(NM,2),DESN(NM,10),UNITN(NM,4)  SHIFT 27
C   IF(IPRINT.EQ.6)CALL DEBUG(1,4HSHIF,4HT .5,0,IW)  SHIFT 28
C   DO 140 I=1,N  SHIFT 29
C   NNN(I)=NN(I)  SHIFT 30
C   DO 120 J=1,2  SHIFT 31
C 120 VNN(I,J)=VN(I,J)  SHIFT 32
C   DO 130 J=1,10  SHIFT 33
C 130 DESN(I,J)=DES(I,J)  SHIFT 34
C   DO 140 J=1,4  SHIFT 35
C 140 UNITN(I,J)=UNIT(I,J)  SHIFT 36
C   IF(IPRINT.EQ.6)CALL DEBUG(2,4HSHIF,4HT .5,0,IW)  SHIFT 37
C   RETURN  SHIFT 38
C   END  SHIFT 39

```

Figure 80. Subroutine SHIFT Program Listing

C	SUBROUTINE TDINVR (ISOL,IDSOL,NR,NC,A,MRA,KWA,DET)	TDINVR 2
C	PURPOSE - TO INVERT A NONSINGULAR MATRIX OR	TDINVR 3
C	TO SOLVE LINEAR EQUATIONS	TDINVR 4
C		TDINVR 5
C	ARGUMENTS LIST	TDINVR 6
C	ISOL	TDINVR 7
C		TDINVR 8
C	IDSOL	TDINVR 9
C		TDINVR10
C		TDINVR11
C		TDINVR12
C	NR	TDINVR13
C	NC	TDINVR14
C	A	TDINVR15
C		TDINVR16
C	MRA	TDINVR17
C	KWA	TDINVR18
C	DET	TDINVR19
C		TDINVR20
C		TDINVR21
C		TDINVR22
C		TDINVR23
C		TDINVR24
C		TDINVR25
C		TDINVR26
C		TDINVR27
C		TDINVR28
C		TDINVR29
C		TDINVR30
C		TDINVR31
C		TDINVR32
C		TDINVR33
C		TDINVR34
C		TDINVR35
C		TDINVR36
C		TDINVR37
C		TDINVR38
C		TDINVR39
C		TDINVR40
C		TDINVR41
C		TDINVR42
C		TDINVR43
C		TDINVR44
C		TDINVR45
C		TDINVR46
C		TDINVR47
C		TDINVR48
C		TDINVR49
C		TDINVR50
C		TDINVR51
C		TDINVR52
C		TDINVR53
C		TDINVR54
C		TDINVR55
C		TDINVR56
C		TDINVR57
C		TDINVR58
C		TDINVR59
C		TDINVR60
C		TDINVR61
C		TDINVR62
C		TDINVR63
C		TDINVR64


```

DIMENSION A(1),KWA(1)
IR=NR
ISOL =1
IDSOL=1
IF(NR) 61,61,11
11 IF(IR-MRA)12,12,61
12 IC=IAHS(NC)
IF(IC-IR) 13,14,14
13 IC=IR
14 IRMP=1
JBMP=MRA
KBMP=IRMP*IRMP
NES=IR*JBMP
NET=IC*JBMP
IF(NC) 15,61,16
15 MDIV=IRMP+1
IRIC=IR-IC
GO TO 17
16 MDIV=1
17 MAD =MDIV
MSER=1
KSER=IR
MZ =1
DET=1.0
18 PIV=0.0
I=MSER
19 IF(I<MSER) 20,20,23
20 IF(ABS(A(I))-PIV)22,22,21
21 PIV=ABS(A(I))
IP=I
22 I=I+IRMP
GO TO 19
23 IF(PIV) 24,62,24
24 IF(NC) 26,25,25
25 I=IP-((IP-1)/IRMP)*IRMP
J=MSER-((MSER-1)/IRMP)*JBMP
JJ=MSER/KBMP+1
II=JJ*(IP-MSER)
KWA(J)=II
GO TO 27
26 I=IP
J=MSER
27 IF(IP-MSER) 61,31,28

```

Figure 81. Subroutine TDINVR Program Listing

28 IF(J-NET) 29,29,30	TDINVR65
29 PSTO=A(I)	TDINVR66
A(I)=A(J)	TDINVR67
A(J)=PSTO	TDINVR68
I=I+JUMP	TDINVR69
J=J+JUMP	TDINVR70
GO TO 28	TDINVR71
30 DET=-DET	TDINVR72
31 PSTO=A(MSER)	TDINVR73
DET=DET*PSTO	TDINVR74
35 PSTO=1.0/PSTO	TDINVR75
A(MSER)=1.0	TDINVR76
I=MDIV	TDINVR77
36 IF(I-NET) 37,37,39	TDINVR78
37 A(I)=A(I)*PSTO	TDINVR79
I=I+JUMP	TDINVR80
GO TO 36	TDINVR81
39 IF(MZ-KSER) 40,40,145	TDINVR82
40 IF(MZ-MSER) 41,44,41	TDINVR83
41 I=MAD	TDINVR84
J=MDIV	TDINVR85
PSTO=A(MZ)	TDINVR86
IF(PSTO) 142,44,142	TDINVR87
142 A(MZ)=0.0	TDINVR88
42 IF(J-NET) 43,43,44	TDINVR89
43 A(I)=A(I)-A(J)*PSTO	TDINVR90
J=J+JUMP	TDINVR91
I=I+JUMP	TDINVR92
GO TO 42	TDINVR93
44 MAD=MAD+IBMP	TDINVR94
MZ=MZ+IBMP	TDINVR95
GO TO 39	TDINVR96
145 KSER=KSER+JBMP	TDINVR97
IF(KSER-NES) 46,46,53	TDINVR98
46 MSER=MSER+KBMP	TDINVR99
IF(NC) 48,47,47	TDINV100
47 MDIV=MDIV+IBMP	TDINV101
MZ=((MSER-1)/JUMP)*JUMP+1	TDINV102
MAD=1	TDINV103
GO TO 52	TDINV104
48 MDIV=MDIV+KBMP	TDINV105
IF(IRIC) 50,49,50	TDINV106
49 MZ=MSER+IBMP	TDINV107
GO TO 51	TDINV108
50 MZ=((MSER-1)/JUMP)*JBMP+1	TDINV109
51 MAD=MZ+JBMP	TDINV110
52 GO TO 18	TDINV111
53 IF(NC) 65,54,54	TDINV112
54 JR=JR	TDINV113
55 IF(JR) 61,65,56	TDINV114
56 IF(KWA(JR)-JR) 61,60,57	TDINV115
57 K=(JR-1)*JBMP	TDINV116
J=K+IR	TDINV117
L=(KWA(JR)-1)*JBMP+JR	TDINV118
58 IF(J-K) 61,60,59	TDINV119
59 PSTO=A(L)	TDINV120
A(L)=A(J)	TDINV121
A(J)=PSTO	TDINV122
J=J+JUMP	TDINV123
L=L+JUMP	TDINV124
GO TO 58	TDINV125
60 JR=JR-1	TDINV126
GO TO 55	TDINV127
61 ISOL=1	TDINV128
GO TO 65	TDINV129
62 DET=0.0	TDINV130

Figure 81. Subroutine TDINVR Program Listing (Continued)

```
ISOL=7  
IDSOL=1  
GO TO 65  
63 ISOL = 2  
IDSOL = 2  
65 RETURN  
END
```

```
TDINV131  
TDINV132  
TDINV133  
TDINV134  
TDINV135  
TDINV136  
TDINV137
```

Figure 81. Subroutine TDINVR Program Listing (Concluded)

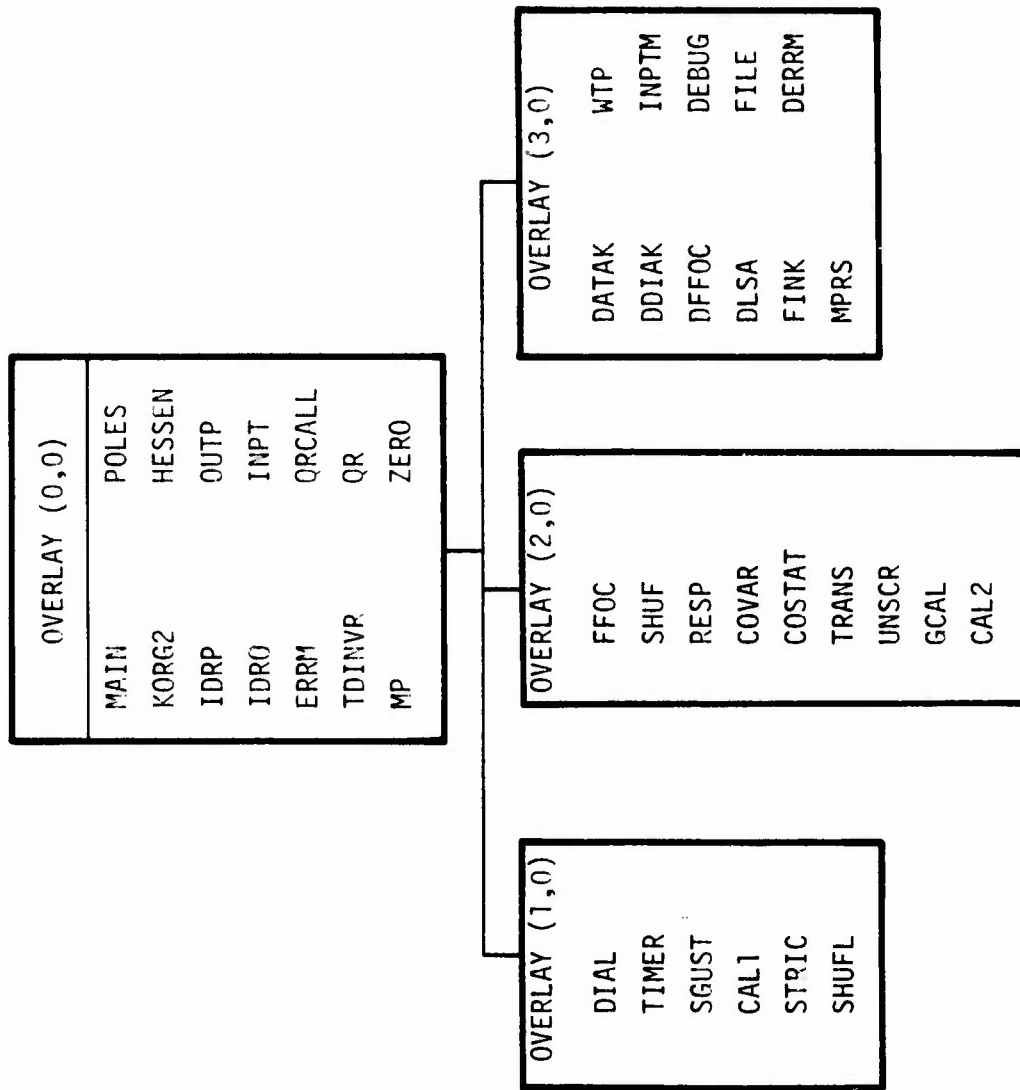


Figure 82. Overlay Structure and Subroutines in KONPACT-2

	OVERLAY (KON2,0,0)	MAIN	2
	PROGRAM MAIN (MINPUT, INPUT, TAPE7=MINPUT, TAPE4=INPUT,	MAIN	3
	1QDATA, OUTPUT, TAPE8=QDATA, TAPE9=OUTPUT, SCRATCH, TAPE5=SCRATCH,	MAIN	4
	2FDATA, DDATA, TAPE1=FDATA, TAPE6=DDATA, SDSTP, TAPE2=SDSTP)	MAIN	5
C		MAIN	6
C	PURPOSE - TO SET UP MAXIMUM DIMENSIONS	MAIN	7
C	ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC	MAIN	8
C	DATE WRITTEN - 1975	MAIN	9
C		MAIN	10
C	SUBPROGRAMS CALLED	MAIN	11
C	KORG2	MAIN	12
C		MAIN	13
C	LABELLED COMMON LIST	MAIN	14
C	NX=1 MAXIMUM NUMBER OF STATES	MAIN	15
C	NRM= MAXIMUM NUMBER OF OUTPUTS	MAIN	16
C	NUM= MAXIMUM NUMBER OF INPUTS	MAIN	17
C	CODE PROGRAM CODE WORD (DIAG, FFOC, LSA)	MAIN	18
C	MS1 MAXIMUM DIMENSION FOR SCRATCH ARRAY S1	MAIN	19
C	MS2 MAXIMUM DIMENSION FOR SCRATCH ARRAY S2	MAIN	20
C	MS3 MAXIMUM DIMENSION FOR SCRATCH ARRAY S3	MAIN	21
C	MS4 MAXIMUM DIMENSION FOR SCRATCH ARRAY S4	MAIN	22
C		MAIN	23
C	COMMON /INF/ NXM, NRM, NUM, CODE, MS1, MS2, MS3, MS4	MAIN	24
C		MAIN	25
C	MAXIMUM SYSTEM DIMENSIONS	MAIN	26
C		MAIN	27
C	NXM=51 \$ NRM=70 \$ NUM=20	MAIN	28
C		MAIN	29
C	MAXIMUM SCRATCH ARRAY DIMENSIONS	MAIN	30
C		MAIN	31
C	MS1=00500 \$ MS2=17000 \$ MS3=00001 \$ MS4=00001	MAIN	32
C		MAIN	33
C	*** NOTE *** SCRATCH ARRAY DIMENSIONS IN PROGRAM DATAK	MAIN	34
C	SHOULD BE CHANGED	MAIN	35
C		MAIN	36
C	CALL KONPACT ORGANIZING SUBROUTINE	MAIN	37
C		MAIN	38
C	CALL KORG2	MAIN	39
C	STOP	MAIN	40
C	END	MAIN	41

Figure 83. Program MAIN Program Listing

```

OVERLAY(KON2.1.0)
PROGRAM DIAK
C DOUBLY-ITERATIVE ALGORITHM FOR SOLVING ALGEBRAIC RICCATI EQUATION
C THIS PROGRAM COMPUTES QUADRATIC CONTROLLERS AND/OR COMPUTES COVARIANCE
C TIME RESPONSES FOR SYSTEMS MODELED AS
C
C XDOT = F*X + G*U + G2*ETA
C AND
C R = M*X + n*U
C WITH
C J = E(R#*Q#R)
C
DIMENSION F(40,40),G1(40,6),G2(40,2),A(40,40),AN(40,40),E(40,40)
DIMENSION Q(40,40),WR(40,40),EP(40,40),P(40,40),H(40,40),D(40,6)
DIMENSION AK(6,40),PI(40,40),DOD(6,6),KWA(40),W(6,40),W1(6,40)
DIMENSION QQ(40,40),RR(90),AM(40,40),BK(6,40),X(40),DX(40),DX1(40)
DIMENSION XI(40,2),XLDXL(40,2),GN(40,2),GS(40,2),R(8000),IPLR(80)
DIMENSION ITITL(80),YMAX(80),YMIN(80),CL(2,1),SCAL(80),NEWY(80)
DIMENSION NORD(40),OR(40,40),IUNIT(80)
COMMON A,E,Q,AN,WR,OR,EP,P,PI
EQUIVALENCE (F(1),P(1)),(H(1),PI(1)),(AM(1),EP(1))
EQUIVALENCE (P(1),E(1))
C DIMENSIONS OF THE ABOVE ARRAYS ARE DEFINED BELOW. CHANGE BOTH SIMULTANEOUSLY
C SEE DOCUMENTATION FOR DEFINITIONS OF ARRAY DIMENSIONS
C MX>NX
C MR>NR
C MU>NU
C MN>NM
C MXR>NOR
C MPOIN>(NOP + 1)*(T/ST)
C MR=40
C MXR=80
C MX=40
C MU=6
C MN=2
C MPOIN=8000
C CONVERGENCE TEST FACTOR
C EE=0.001
C ITERATION COUNTER
C RUN COUNTER
C IRUN=1
C READ AND PRINT ID
C READ(5,1274) IDATE,NAME1,NAME2
C 1274 FORMAT(3A10)
C WRITE(9,1275) IDATE,NAME1,NAME2
C 1275 FORMAT(1H1,7X,13HTODAY#S DATE ,A10,5X,16HIDENTIFICATION ,2A10//)
C READ NUMBER OF VARIABLES BEING PLOTTED
C READ(5,28) NOP
C 28 FORMAT(40I2)
C IF NOP = 0, SKIP TO STATEMENT 70
C IF(NOP.EQ.0) GO TO 70
C READ PLOTTING PARAMETERS - THEY ARE FIXED FOR ALL RUNS
C IPLR = ARRAY OF PLOTTING VARIABLE NOS. - READ IN ORDER
C ITITL = CORRESPONDING ARRAY OF LABELS
C IUNIT = CORRESPONDING ARRAY OF UNIT LABELS
C YMAX,YMIN = CORRESPONDING ARRAYS OF DESIGNATED MAX AND MIN VALUES
C SCAL = CORRESPONDING ARRAY OF SCALE FACTORS
C READ(5,1272) ((IPLR(I),ITITL(I),IUNIT(I),YMIN(I),YMAX(I),SCAL(I))
C 1,I=1,NOP)
C 1272 FORMAT(12,2X,A10,2X,A10,2X,G11.3,2X,G11.3,2X,G11.3)
C DEFINE PLOTTING SCALES - FIXED FOR ALL RUNS
C IF YMIN AND YMAX ARE 0, USE COMPUTED MAX AND MIN (NEWY=1)
C IF SCAL = 0, USE SCALE FACTOR OF 1

```

Figure 84. Program DIAK Program Listing


```

DO 1 I=1,NOP
NEWY(I)=0
IF(YMIN(I).EQ.0..AND.YMAX(I).EQ.0.) NEWY(I)=1
IF(SCAL (I).EQ.0.) SCAL (I)=1.
1 CONTINUE
C READ AND PRINT PLOTTING TIME PARAMETERS - FIXED FOR ALL RUNS
C T = TOTAL PLOTTING TIME
C DT = SAMPLING INTERVAL
C ST = PLOTTING SAMPLING INTERVAL
C T1 = FIRST DELAY IN GUST PROFILE
C T2 = SECOND DELAY IN GUST PROFILE
READ(5,1270) T,DT,ST,T1,T2
1270 FORMAT(5G12.4)
C PRINT PLOTTING PARAMETERS
WRITE(9,1279) T,DT,ST,T1,T2
1279 FORMAT(1H0/7X,31H TIME RESPONSES PLOTTING TIME =,G12.4/22X,18H SAMDIAK 80
1PLE INTERVAL =,G12.4/22X,27H PLOTTING SAMPLE INTERVAL =,G12.4/ DIAK 81
222X,19H FIRST DELAY TIME =,G12.4/22X,20H SECONDD DELAY TIME =,G12.4DIAK 82
3//) DIAK 83
WRITE(9,1285) ((IPLR(I),ITITL(I),IUNIT(I),YMIN(I),YMAX(I),SCAL (I)DIAK 84
I),I=1,NOP) DIAK 85
1285 FORMAT(7X,18HPLOTTING VARIABLES//2X,86HRESPONSE NUMBER RESPONSE VOIAK 86
1ARIABLE RESPONSE UNITS MIN SCALE MAX SCALE SCALE FACTOR//(2X, DIAK 87
2110,9X,A10,9X,A10,3X,G11.3,2X,G11.3,3X,G11.3)) DIAK 88
C READ AND PRINT MAX NO. OF INNER AND OUTER LOOP ITERATIONS
70 READ(5,28) IMAX,ITER,ITER0
WRITE(9,4002)IMAX,ITER,ITER0
4002 FORMAT(///7X,37H MAX NUMBER OF INNER-LOOP ITERATIONS I3,37H MAX NDIAK 92
NUMBER OF OUTER-LOOP ITERATIONS I3/7X,67H MAX NUMBER OF ITERATIONS DIAK 93
20N ELIMINATING CONTROL SURFACE FEEDBACKS I3//) DIAK 94
C DEFINITION OF PROGRAM OPTIONS
C INPD=1 COMPLETELY NEW DATA
C INPD=2 CHANGE SELECTED QUADRATIC WEIGHTS ONLY - USE SOME GAINS IN SDIAK 97
C INPD=3 CHANGE SELECTED QUADRATIC WEIGHTS ONLY WITH OPTION FOR NEW GADIAK 98
C INPD=4 CHANGE SELECTED DATA
C INPD=5 CHANGE SELECTED DATA IN MEASUREMENT MATRIX, QUADRATIC WEIGHTSDIAK 100
OPTION FOR NEW GAINS
C INPK=1 NEW INPUT GAINS
C INPK=2 NEW STARTING ROUTINE GAINS
C INPK=3 USE GAINS IN STORAGE
C INPK=4 USE INPUT GAINS IN STORAGE
C NCONT=0 DONOT COMPUTE OPTIMAL GAINS - USE INPUT GAINS AND DATA IN CODIAK 106
AND TIME RESPONSE ANALYSIS ONLY
C NCONT=1 COMPUTE OPTIMAL GAINS
C NCONT=2 DO AUTOMATIC SELECTION OF Q ON CONTROL RATES
C SEE SUBROUTINE TIMER FOR PLOTTING OPTIONS USING NPLOT, NPRIN, NSTEP, DIAK 110
C NOCOV=1 NO COVARIANCE ANALYSIS
C NOCOV=2 COVARIANCE ANALYSIS
C NOCOV=3 SKIP CORRELATION ANALYSIS
C READ AND PRINT PROGRAM OPTIONS
READ(5,28) NOCOV,NSTEP,NRAND,NPRIN ,NPLOT
READ(5,28) INPK
INPD=1
READ(5,28) NCONT
WRITE(9,37) INPD,INPK,NCONT,NOCOV,NSTEP,NRAND,NPRIN ,NPLOT
37 FORMAT(1H1/7X,23HNEW PROBLEM WITH INPD =,I3,2X,6HINPK =,I3,2X,
17HNCONT =,I3/7X,7HNOCOV =,I3/7X,7HNSTEP =,I3,2X,7HNRAND =,I3/7X,
27HNPRIN =,I3,2X,7HNPL0T =,I3//)
1210 CONTINUE
C READ FLIGHT CONDITION ID
READ(5,1270)IFLT
1270 FORMAT(A10)
C PRINT FLIGHT CONDITION ID AND RUN NO.
WRITE(9,1271) IFLT, IRUN
1271 FORMAT(1H1/7X,18H FLIGHT CONDITION A10,5X,3HRUN,I3)
C READ AND PRINT SYSTEM PARAMFTERS

```

Figure 84. Program DIAK Program Listing (Continued)

```

C      NX = NO. OF STATES                                DIAK 131
C      NR = NO. OF RESPONSES                            DIAK 132
C      NU = NO. OF CONTROLS                             DIAK 133
C      NN = NO. OF DISTURRANCE INPUTS                  DIAK 134
C      NSCRR = RESPONSE STARTING CONTROL RATE RESPONSES DIAK 135
C      PARAMETERS FOR PLOTTING                          DIAK 136
C      NF = NO. OF FEEDBACK STATES = NX - NO. OF DISTURBANCE STATES (NOT DIAK 137
C      NG = NO. OF GUST INPUTS                          DIAK 138
C      NCS = NO. OF COMMAND INPUTS = NO. OF COMMAND STATES DIAK 139
C      NGLG = NO. OF GUST LIFT GROWTH STATES            DIAK 140
C      READ(5,28) NX,NR,NU,NN,NF,NG,NCS,NGLG,NSCRR     DIAK 141
C      WRITE(9,4003) NX,NR,NU,NN,NF,NG,NCS,NGLG,NSCRR DIAK 142
4003  FORMAT(//7X,18H ORDER OF SYSTEM =I3/7X,22H NUMBER OF RESPONSES =I3DIAK 143
1/7X,21H NUMBER OF CONTROLS =I3/7X,31H NUMRER OF DISTURBANCE INPUTSDIAK 144
2 =,I3/7X,27HNUMBER OF FEEDBACK STATES =,I3/7X,24H NUMBER OF GUST IDIAK 145
3NPUTS =,I3/7X,27H NUMBER OF COMMAND STATES =,I3/7X,36H NUMBER OF GDIAK 146
4UST LIFT GROWTH STATES =,I3/7X,43H CONTROL RATE RESPONSES START WIDIAK 147
5TH RESPONSEI3//)
C      NC IS THE NUMBER OF UPPER TRIANGULAR ELEMENTS IN P DIAK 149
C      NC=(NX*(NX+1))/2                                  DIAK 150
C
C      ZERO ARRAYS
C
C      RIGHT HAND PARAMETERS DEFINED BELOW              DIAK 154
C      DO 8020 I=1,MX
C      DO 8013 J=1,MX
C      F (I,J)=0.                                       DIAK 155
C      A (I,J)=0.                                       DIAK 156
C      AN (I,J)=0.                                       DIAK 157
C      E (I,J)=0.                                       DIAK 158
C      EP (I,J)=0.                                       DIAK 159
C      PI (I,J)=0.                                       DIAK 160
C      AM(I, )=0.                                       DIAK 161
C      AM(I, )=0.                                       DIAK 162
C      AM(I, )=0.                                       DIAK 163
8013  CONTINUE                                          DIAK 164
C      DO 8014 J=1,NU
C      G1(I, )=0.                                       DIAK 165
8014  DO 8015 J=1,NN
C      XI(I, )=0.                                       DIAK 166
C      CL(J,1)=0.                                       DIAK 167
C      CL(J,1)=0.                                       DIAK 168
8015  G2(I, )=0.                                       DIAK 169
C      DO 8016 J=1,2
C      XLDXL(I,J)=0.                                       DIAK 170
8016  XLDXL(I,J)=0.                                       DIAK 171
C      8020 CONTINUE                                          DIAK 172
C      DO 3060 I=1,NR
C      DO 3061 J=1,NR
C      3061 Q0(I, )=0.                                       DIAK 173
C      DO 3062 J=1,MX
C      3062 H(I,J)=0.                                       DIAK 174
C      DO 3060 J=1,NU
C      3060 D(I,J)=0.                                       DIAK 175
C      3061 Q0(I, )=0.                                       DIAK 176
C      DO 3062 J=1,MX
C      3062 H(I,J)=0.                                       DIAK 177
C      DO 3060 J=1,NU
C      3060 D(I,J)=0.                                       DIAK 178
C      READ DATA FOR THIS RUN                               DIAK 179
1240  IF (INPD.GT.1) GO TO 53                               DIAK 180
C      IF INPD = 1 (NEW DATA), READ ORDERING OF STATES DIAK 181
C      NORD = ARRAY OF THE ORDER OF STATES                DIAK 182
C      READ(5,28) (NORD(I),I=1,NX)                       DIAK 183
C      WRITE(9,67) (NORD(I),I=1,NX)                      DIAK 184
67    FORMAT(//7X,22H STATES ARE ORDERED AS//(7X,20I4)//) DIAK 185
53    CONTINUE                                          DIAK 186
C      READ CHANGES IN F,G1,G2 (ROW, COLUMN, ELEMENT VALUE) DIAK 187
C      F = STABILITY MATRIX (OPEN LOOP)                  DIAK 188
C      G1 = CONTROL INPUT MATRIX                         DIAK 189
C      G2 = DISTURBANCE INPUT MATRIX                    DIAK 190
C      WHERE                                              DIAK 191
C      XDOT = F*X + G1*U + G2*ETA                        DIAK 192
C      IF INPD>1 (CHANGES TO EXISTING DATA), ROW AND COLUMNS CORRESPOND DIAK 193
C      RE-ORDERED STATES                                  DIAK 194
C      RE-ORDERED STATES                                  DIAK 195
C      RE-ORDERED STATES                                  DIAK 196

```

Figure 84. Program DIAK Program Listing (Continued)

CALL INPT (F,MX,MX)	DIAK 197
CALL INPT(G1,MX,MU)	DIAK 198
CALL INPT (G2,MX,MN)	DIAK 199
IF(INPD.GT.1) GO TO 54	DIAK 200
C IF DATA IS NEW, RE-ORDER THE STATES (CALL SHUFL)	DIAK 201
CALL SHUFL(F,MX,MX,NX,NX,1.0,NORD,0,MX)	DIAK 202
CALL SHUFL(G1,MX,MU,NX,NU,1.0,NORD,0,MX)	DIAK 203
CALL SHUFL(G2,MX,MN,NX,NN,1.0,NORD,0,MX)	DIAK 204
54 CONTINUE	DIAK 205
C PRINT F,G1,G2	DIAK 206
WRITE(9,20)	DIAK 207
CALL MP(MX,MX,NX,NX,F)	DIAK 208
WRITE(9,21)	DIAK 209
CALL MP(MX,MU,NX,NU,G1)	DIAK 210
WRITE(9,22)	DIAK 211
22 FORMAT(1H1/7X,10H G2 MATRIX//)	DIAK 212
CALL MP(MX,MN,NX,NN,G2)	DIAK 213
20 FORMAT(1H1/7X,10H F MATRIX//)	DIAK 214
21 FORMAT(1H1/7X,10H G1 MATRIX//)	DIAK 215
C READ CHANGES IN XI AND XLDXL	DIAK 216
XI = INITIAL STATE VALUES IN SIMULATION	DIAK 217
C XLDXL = STATE AND STATE RATE LIMITS	DIAK 218
CALL INPT(XI,MX,MN)	DIAK 219
CALL INPT(XLDXL,MX,2)	DIAK 220
IF(INPD.GT.1) GO TO 55	DIAK 221
C IF DATA IS NEW, RE-ORDER THE STATES (CALL SHUFL)	DIAK 222
CALL SHUFL(XI,MX,MN,NX,NN,1.0,NORD,0,MX)	DIAK 223
CALL SHUFL(XLDXL,MX,2,NX,2,1.0,NORD,0,MX)	DIAK 224
55 CONTINUE	DIAK 225
C READ CL = STEP GUST AND COMMAND INPUT LEVELS	DIAK 226
CALL INPT(CL,MN,1)	DIAK 227
C PRINT XI, XLDXL, CL	DIAK 228
WRITE(9,1276)	DIAK 229
1276 FORMAT(1H1/7X,24HINITIAL CONDITION MATRIX//)	DIAK 230
CALL MP(MX,MN,NX,NN,XI)	DIAK 231
WRITE(9,1277)	DIAK 232
1277 FORMAT(1H1/7X,31HSTATE LIMIT - RATE LIMIT MATRIX//)	DIAK 233
CALL MP(MX,2,NX,2,XLDXL)	DIAK 234
WRITE(9,1273)	DIAK 235
1273 FORMAT(1H1/7X,20HCOMMAND LEVEL MATRIX//)	DIAK 236
CALL MP(MN,1,NN,1,CL)	DIAK 237
C READ IN CHANGES IN H AND D	DIAK 238
H = STATE-RESPONSE OUTPUT MATRIX	DIAK 239
D = CONTROL-RESPONSE OUTPUT MATRIX	DIAK 240
C	DIAK 241
R = H*X + D*U	DIAK 242
C	DIAK 243
WHERE	DIAK 244
C	DIAK 245
73 CALL INPT (H,MR,MX)	DIAK 246
CALL INPT(D,MR,MU)	DIAK 247
IF(INPD.GT.1) GO TO 1250	DIAK 248
C IF DATA IS NEW, RE-ORDER STATES (CALL SHUFL)	DIAK 249
CALL SHUFL(H,MR,MX,NR,NX,0.1,NORD,0,MX)	DIAK 250
C READ CHANGES IN M (AM)	DIAK 251
AM = MEASUREMENT MATRIX - USED FOR RESPONSE ANALYSIS ONLY	DIAK 252
C	DIAK 253
WHERE	DIAK 254
Y = M*X	DIAK 255
C	DIAK 256
1250 CALL INPT (AM,MX,MX)	DIAK 257
IF(INPD.GT.1) GO TO 56	DIAK 258
C IF DATA IS NEW, RE-ORDER STATES (CALL SHUFL)	DIAK 259
CALL SHUFL(AM,MX,MX,NX,NX,0.1,NORD,0,MX)	DIAK 260
56 CONTINUE	DIAK 261
C PRINT H, D, AM	DIAK 262
WRITE(9,23)	
CALL MP(MR,MX,NR,NX,H)	

Figure 84. Program DIAK Program Listing (Continued)

WRITE(9,24)	DIAK 263
CALL MP(MR,MU,NR,NU,D)	DIAK 264
WRITE(9,29)	DIAK 265
29 FORMAT(1H1/7X,10H M MATRIX//)	DIAK 266
CALL MP(MX,MX,NX,NX,AM)	DIAK 267
23 FORMAT(1H1/7X,10H M MATRIX//)	DIAK 268
24 FORMAT(1H1/7X,10H N MATRIX//)	DIAK 269
C CHECK GAINS INPUT OPTION	DIAK 270
1230 GO TO (300),3002,3010,3011),INPK	DIAK 271
C NEW INPUT GAINS	DIAK 272
C BK = INPUT GAINS MATRIX	DIAK 273
C WHERE	DIAK 274
C U = BK*X (WHEN COMPUTING OPTIMAL GAINS)	DIAK 275
C OR	DIAK 276
C U = BK*Y = BK*AM*X (WHEN COMPUTING RESPONSES ONLY)	DIAK 277
C ZERO AND READ BK	DIAK 278
3001 DO 3003 I=1,NU	DIAK 279
DO 3003 J=1,MX	DIAK 280
3003 BK(I,J)=0.	DIAK 281
CALL INPT (BK,MU,MX)	DIAK 282
IF(NCONT.EQ.0) GO TO 57	DIAK 283
C IF NCONT>0, RE-ORDER STATES (BECAUSE U = BK*X)	DIAK 284
CALL SHUFL(BK,MU,MX,NU,NX,0,1,NORD,0,4X)	DIAK 285
57 CONTINUE	DIAK 286
C PRINT BK	DIAK 287
WRITE(9,30)	DIAK 288
30 FORMAT(1H1/7X,19H INPUT GAINS MATRIX//)	DIAK 289
CALL MP(MU,MX,NU,NX,BK)	DIAK 290
C SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS	DIAK 291
GO TO 1220	DIAK 292
C USE STARTING ROUTINE (STRIC) TO COMPUTE STARTING GAINS - AS A LAST	DIAK 293
C BK = -G1*(W(T)); (A: MEANS INVERSE OF MATRIX A)	DIAK 294
C WHERE	DIAK 295
C W(T) = INTEGRAL(0,BT)OF(EXP(F*T)*G1*G1#*EXP(F#*T))DT	DIAK 296
C FOR AN ARBITRARY TIME BT	DIAK 297
3002 CALL STRIC(F,G1,A,AN,E,Q,MF,NU,MX,MU)	DIAK 298
CALL TDINVR(ISOL,IDSOL,NF,NF,AN,MX,KWA,DET)	DIAK 299
IF((ISOL*IDSOL)-2) 3004;3004,3005	DIAK 300
C (W(T)); IS NO GOOD - GO TO NEXT RUN - BUT FIRST, READ REMAINING DATA	DIAK 301
C THIS RUN AND CHECK TO SEE IF THE NEXT RUN IS SOLVABLE - THE START	DIAK 302
C GAINS MAY NOT BE GOOD - IF SO, STOP	DIAK 303
3005 WRITE(9,3006)	DIAK 304
3006 FORMAT(1H1/7X,31H INVERSE OF W(T) DOES NOT EXIST/7X,10H CHECK NEXT	DIAK 305
PROBLEM//)	DIAK 306
CALL INPT (QQ,MR,MR)	DIAK 307
READ(5,1215) IDUM	DIAK 308
IF(IDUM.GT.0) STOP 77	DIAK 309
READ(5,28) INPD,INPK	DIAK 310
IF(INPK.EQ.1) GO TO 1216	DIAK 311
IF(INPK.EQ.2.AND.(INPD.EQ.1.OR.INPD.EQ.4))GO TO 1216	DIAK 312
WRITE(9,3008)	DIAK 313
3008 FORMAT(/7X,51HNEW PROBLEM NOT SOLVABLE WITHOUT NEW STARTING GAINS/	DIAK 314
1/)	DIAK 315
STOP 11	DIAK 316
C DEFINE BK	DIAK 317
C AN = (WCT));	DIAK 318
C	DIAK 319
C	DIAK 320
3004 DO 3009 I=1,NU	DIAK 321
DO 3009 J=1,NF	DIAK 322
BK(I,J)=0.	DIAK 323
DO 3009 K=1,NF	DIAK 324
3009 BK(I,J)=BK(I,J)-G1(K,I)*AN(K,J)	DIAK 325
C PRINT BK	DIAK 326
WRITE(9,31)	DIAK 327
31 FORMAT(1H1/7X,22H STARTING GAINS MATRIX//)	DIAK 328

Figure 84. Program DIAK Program Listing (Continued)

CALL MP(MU, MX, NU, NF, BK)	DIAK 329
C SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS	DIAK 330
GO TO 1220	DIAK 331
C USE USE LAST COMPUTED GAINS IN STORAGE FOR STARTING GAINS	DIAK 332
C DEFINE BK = AK	DIAK 333
3010 WRITE(9,33)	DIAK 334
33 FORMAT(1H1/7X,28H USE GAINS MATRIX IN STORAGE//)	DIAK 335
C SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS	DIAK 336
GO TO 1220	DIAK 337
C USE INPUT GAINS IN STORAGE - RK = BK	DIAK 338
3011 WRITE(9,34)	DIAK 339
34 FORMAT(1H1/7X,34H USE INPUT GAINS MATRIX IN STORAGE//)	DIAK 340
C READ CHANGES IN QUADRATIC WEIGHTS FOR PERFORMANCE INDEX	DIAK 341
C	DIAK 342
C J = E(R#*Q#R)	DIAK 343
C WHERE Q IS THE MATRIX OF QUADRATIC WEIGHTS	DIAK 344
C Q0 = Q	DIAK 345
1220 CONTINUE	DIAK 346
CALL INPT(Q0, MR, MR)	DIAK 347
NO=1	DIAK 348
81 CONTINUE	DIAK 349
C PRINT Q0	DIAK 350
WRITE(9,36)	DIAK 351
36 FORMAT(1H1/7X,27H QUADRATIC WEIGHTING MATRIX//)	DIAK 352
CALL MP(MR, MR, NR, NP, Q0)	DIAK 353
C IF NCONT = 0 (NO OPTIMAL CONTROL COMPUTATIONS), SKIP TO STATEMENT 89	DIAK 354
C RESPONSE COMPUTATIONS	DIAK 355
IF(NCONT.EQ.0) GO TO 893	DIAK 356
C CALCULATE A,E,Q FOR PICCATI EQUATION $0 = PA + A#P + Q - PEP$	DIAK 357
C $W = D#*Q$	DIAK 358
DO 4 I=1,NU	DIAK 359
DO 4 J=1,NR	DIAK 360
W(I,J)=0.	DIAK 361
DO 4 K=1,NR	DIAK 362
4 W(I,J)=W(I,J)+D(K,I)*Q0(K,J)	DIAK 363
C DDD = D#*Q#D	DIAK 364
DO 5 I=1,NU	DIAK 365
DO 5 J=1,NU	DIAK 366
DDD(I,J)=0.	DIAK 367
DO 5 K=1,NR	DIAK 368
5 DDD(I,J)=DDD(I,J)+W(I,K)*D(K,J)	DIAK 369
C INVERT DDD - DDD = (D#*Q#D):	DIAK 370
IF(NU-1)302,302,301	DIAK 371
302 DDD(1,1)=1./DDD(1,1)	DIAK 372
GOTO 303	DIAK 373
301 CONTINUE	DIAK 374
CALL TOINVR(ISOL,IDSOL,NU,NU,DDD,MU,KWA,DET)	DIAK 375
IF((ISOL+IDSOL)-2)6,6,7	DIAK 376
C (D#*Q#D): DOES NOT EXIST - GO TO NEXT RUN	DIAK 377
7 WRITE(9,35)	DIAK 378
35 FOPMAT(1H1/7X,30H INVERSE OF DDD DOES NOT EXIST//7X,19H CHECK NEXT	DIAK 379
1 PROBLEM//)	DIAK 380
GO TO 1200	DIAK 381
6 CONTINUE	DIAK 382
303 CONTINUE	DIAK 383
C W1 = D#*Q#H	DIAK 384
DO 8 I=1,NU	DIAK 385
DO 8 J=1,NX	DIAK 386
W1(I,J)=0.	DIAK 387
DO 8 K=1,NR	DIAK 388
8 W1(I,J)=W1(I,J)+W(I,K)*H(K,J)	DIAK 389
C W = (D#*Q#D):#D#*Q#H	DIAK 390
C STORF W FOR OPTIMAL CONTROL COMPUTATION	DIAK 391
DO 9 I=1,NU	DIAK 392
DO 9 J=1,NX	DIAK 393
W(I,J)=0.	DIAK 394

Figure 84. Program DIAK Program Listing (Continued)

	DO 9 K=1,NU	DIAK 395
	9 W(I,J)=W(I,J)+DD(I,K)*W(K,J)	DIAK 396
C	AN = F - G1*(D**Q*D):*D**Q*H	DIAK 397
C	AN = A OF EQUATION 0 = A*P + PA + Q - PEP	DIAK 398
	DO 10 I=1,NX	DIAK 399
	DO 10 J=1,NX	DIAK 400
	AN(I, I)=F(I,J)	DIAK 401
	DO 10 K=1,NU	DIAK 402
10	AN(I, I)=AN(I, J)-G1(I,K)*W(K, J)	DIAK 403
C	Q = -H**Q*D*(D**Q*D):*D**Q*H	DIAK 404
	DO 12 I=1,NX	DIAK 405
	DO 12 J=1,NX	DIAK 406
	Q(I, J)=0.	DIAK 407
	DO 12 K=1,NU	DIAK 408
12	Q(I, J)=Q(I, J)-W(K, I)*W(K, J)	DIAK 409
C	E = Q*H	DIAK 410
	DO 13 I=1,NR	DIAK 411
	DO 13 J=1,NX	DIAK 412
	E(I, J)=0.	DIAK 413
	DO 13 K=1,NR	DIAK 414
13	E(I, J)=E(I, J)+Q(I, K)*H(K, J)	DIAK 415
C	Q = H**Q*H - H**Q*D*(D**Q*D):*D**Q*H	DIAK 416
C	Q = Q OF EQUATION 0 = A*P + PA + Q - PEP	DIAK 417
	DO 14 I=1,NX	DIAK 418
	DO 14 J=1,NX	DIAK 419
	DO 15 K=1,NR	DIAK 420
15	Q(I, J)=Q(I, J)+H(K, J)*F(K, J)	DIAK 421
14	Q(J, I)=Q(I, J)	DIAK 422
C	W1 = (D**Q*D):*G1*	DIAK 423
	DO 16 I=1,NU	DIAK 424
	DO 16 J=1,NX	DIAK 425
	W1(I, J)=0	DIAK 426
	DO 16 K=1,NU	DIAK 427
16	W1(I, J)=W1(I, J)+D(I, K)*G1(J, K)	DIAK 428
C	E = G1*(D**Q*D):*G1*	DIAK 429
C	E = E OF EQUATION 0 = A*P + PA + Q - PEP	DIAK 430
	DO 17 I=1,NX	DIAK 431
	DO 17 J=1,NX	DIAK 432
	E(I, J)=0.	DIAK 433
	DO 18 K=1,NU	DIAK 434
18	E(I, J)=E(I, J)+G1(I, K)*W1(K, J)	DIAK 435
17	E(J, I)=E(I, J)	DIAK 436
C	PRINT AN, E, Q	DIAK 437
	WRITE(9,32)	DIAK 438
32	FORMAT(1H1/7X,36HSTARTING MATRICES FOR PA*A*P+Q-PEP=0//)	DIAK 439
	WRITE(9,25)	DIAK 440
	CALL MP(MX,MX,NX,NX,AN)	DIAK 441
	WRITE(9,26)	DIAK 442
	CALL MP(MX,MX,NX,NX,E)	DIAK 443
	WRITE(9,27)	DIAK 444
	CALL MP(MX,MX,NX,NX,Q)	DIAK 445
25	FORMAT(//7X,10H A MATRIX//)	DIAK 446
26	FORMAT(1H1/7X,10H E MATRIX//)	DIAK 447
27	FORMAT(1H1/7X,10H Q MATRIX//)	DIAK 448
C	DUMP F, Q, AND AN ON DISC TO CONSERVE STORAGE	DIAK 449
C	P, P1, AND EP USE STORAGE EQUIVALENT TO THESE MATRICES	DIAK 450
	REWIND 2	DIAK 451
	WRITE(2) F	DIAK 452
	WRITE(2) H, AN	DIAK 453
	ITERC=0	DIAK 454
C	CHECK GAINS INPUT OPTION	DIAK 455
	GO TO (3000,3000,2(50,3000),INPK	DIAK 456
C	FOR ALL OPTIONS EXCEPT INPK = 3 (USE AK IN STORAGE), AK = BK	DIAK 457
3000	DO 7010 I=1,NU	DIAK 458
	DO 7010 J=1,NX	DIAK 459
7010	AK(I, J)=BK(I, J)	DIAK 460

Figure 84. Program DIAK Program Listing (Continued)

```

2050 CONTINUE
C A = F * G1 * K
C A = CLOSED LOOP STABILITY MATRIX
DO 7011 I=1,NX
DO 7011 J=1,NX
A(I,J)=F(I,J)
DO 7011 K=1,NU
7011 A(I,J)=A(I,J)+G1(I,K)*AK(K,J)
C H = H * D * K
C H IS NOW CLOSED LOOP STATE-RESPONSE OUTPUT MATRIX
DO 7012 I=1,NR
DO 7012 J=1,NX
DO 7012 K=1,NU
7012 H(I,J)=H(I,J)+D(I,K)*AK(K,J)
C COMPUTE (H * D * K) * Q * (H * D * K)
C P = Q * (H * D * K)
DO 7013 I=1,NR
DO 7013 J=1,NX
P(I,J)=0.
DO 7013 K=1,NR
7013 P(I,J)=P(I,J)+QQ(I,K)*H(K,J)
C EP = (H * D * K) * P = (H * D * K) * Q * (H * D * K)
DO 7014 I=1,NX
DO 7014 J=1,NX
EP(I,J)=0.
DO 7014 K=1,NR
7014 EP(I,J)=EP(I,J)+H(K,I)*P(K,J)
C SOLVE FOR INITIAL RICCATI MATRIX P FROM
C 0 = A * P + P * A + (H * D * K) * Q * (H * D * K)
C
C VIA SUBROUTINE CAL
C P IS WORKING MATRIX HERE - RICCATI MATRIX RETURNS IN EP
CALL CAL1(A,EP,P,KWA,NX,MX,IMAX,1,IERR,EE)
IF(IERR.EQ.0) GO TO 875
C ERROR ENCOUNTERED IN CAL - GO TO NEXT RUN
WRITE(9,38)
38 FORMAT(1H1/7X,27H INITIAL GAINS ARE UNSTABLE//7X,19H CHECK NEXT PROIAK 497
10RLEM//)
READ(5,1215) IOUM
IF(IOUM.GT.0) STOP 77
READ(5,28) INPD,INPK
IF(INPK.EQ.1) GO TO 1216
IF(INPK.EQ.2.AND.(INPD.EQ.1.OR.INPD.EQ.4)) GO TO 1216
C NEXT RUN NOT SOLVABLE WITH PRESENT STARTING GAINS - SO STOP
WRITE(9,3008)
STOP 11
C SET P = EP, INITIALIZE PI = 0
875 DO 876 I=1,NX
DO 876 J=1,NX
PI(I,J)=0.
876 P(I,J)=EP(I,J)
C UPDATE A AND Q MATRICES FOR NEXT ITERATION
C A = AN - E * P
C Q = Q + P * E * P
C TO SOLVE FOR P FROM
C
C 0 = A * P + P * A + Q
C
C VIA SUBROUTINE CAL
C AFTER SOLVING FOR SECOND P, SOLVE FOR DIFFERENCES IN P BETWEEN ITERADIAK 520
C THUS INITIALIZE DIFFERENCES AND CONVERGENCE CRITERIA DIAK 521
C P IS DIFFERENCE AND PI IS THE TOTAL RICCATI MATRIX DIAK 522
C INITIALLY PI IS ZERO DIAK 523
DO 100 I=1,NX
DO 100 J=1,NX
EP(I,1)=0.
DIAK 461
DIAK 462
DIAK 463
DIAK 464
DIAK 465
DIAK 466
DIAK 467
DIAK 468
DIAK 469
DIAK 470
DIAK 471
DIAK 472
DIAK 473
DIAK 474
DIAK 475
DIAK 476
DIAK 477
DIAK 478
DIAK 479
DIAK 480
DIAK 481
DIAK 482
DIAK 483
DIAK 484
DIAK 485
DIAK 486
DIAK 487
DIAK 488
DIAK 489
DIAK 490
DIAK 491
DIAK 492
DIAK 493
DIAK 494
DIAK 495
DIAK 496
DIAK 497
DIAK 498
DIAK 499
DIAK 500
DIAK 501
DIAK 502
DIAK 503
DIAK 504
DIAK 505
DIAK 506
DIAK 507
DIAK 508
DIAK 509
DIAK 510
DIAK 511
DIAK 512
DIAK 513
DIAK 514
DIAK 515
DIAK 516
DIAK 517
DIAK 518
DIAK 519
DIAK 520
DIAK 521
DIAK 522
DIAK 523
DIAK 524
DIAK 525
DIAK 526

```

Figure 84. Program DIAK Program Listing (Continued)

DO 101 K=1,NX	DIAK 527
101 EP(I,I)=EP(I,J)*E(I,K)*P(K,J)	DIAK 528
100 A(I,J)=AN(I,J)-EP(I,J)	DIAK 529
DO 102 I=1,NX	DIAK 530
DO 102 J=1,NX	DIAK 531
DO 102 K=1,NX	DIAK 532
103 Q(I,J)=Q(I,J)+P(I,K)*EP(K,J)	DIAK 533
102 Q(J,I)=Q(I,J)	DIAK 534
EEE=EP	DIAK 535
1000 CONTINUE	DIAK 536
DO 2010 I=1,NX	DIAK 537
DO 2010 J=1,NX	DIAK 538
2010 PI(I,I)=P(I,J)+PI(I,J)	DIAK 539
CALL SECONO(TT)	DIAK 540
WRITE(9,3055) TT	DIAK 541
3055 FORMAT(//7X,6HTIME =,F10.5//)	DIAK 542
C CALL CAL -P IS AGAIN WORKING MATRIX - RICCATI MATRIX RETURNS IN Q	DIAK 543
CALL CAL(A,Q,P,KWA,NX,MA,IMAX,I,IERR,EEE)	DIAK 544
EEE=EP*10.	DIAK 545
CALL SECONO(TT)	DIAK 546
WRITE(9,3055) TT	DIAK 547
IF(IERR,EO,0) GO TO 874	DIAK 548
C ERROR ENCOUNTERED IN CAL - GO TO NEXT RUN	DIAK 549
WRITE(9,39)	DIAK 550
39 FORMAT(1H1//7X,30H RICCATI SOLUTION IS DIVERGING//7X,19H CHECK NEXT	DIAK 551
1 PROBLEM//)	DIAK 552
READ(5,1215) IDUM	DIAK 553
IF(IDUM,GT,0) STOP 77	DIAK 554
READ(5,28) INPD,INPK	DIAK 555
IF(INPK,EO,1) GO TO 1216	DIAK 556
IF((INPK,EO,2) AND ((INPD,EO,1) OR (INPD,EO,4))) GO TO 1216	DIAK 557
C NEXT RUN NOT SOLVABLE WITH PRESENT STARTING GAINS = SO STOP	DIAK 558
WRITE(9,3008)	DIAK 559
STOP 11	DIAK 560
C SET P = Q	DIAK 561
874 DO 877 I=1,NX	DIAK 562
DO 877 J=1,NX	DIAK 563
877 P(I,J)=Q(I,J)	DIAK 564
IF(ITERC,GT,0) GO TO 3057	DIAK 565
C ON SECOND ITERATION - SOLVE FOR DIFFERENCE P=P - PI	DIAK 566
DO 3058 I=1,NX	DIAK 567
DO 3058 J=1,NX	DIAK 568
3058 P(I,J)=P(I,J)-PI(I,J)	DIAK 569
3057 CONTINUE	DIAK 570
ITERC=ITERC+1	DIAK 571
C UPDATE A AND Q FOR NEXT ITERATION WHERE	DIAK 572
C A = AN - E*(P*PI) - (P*PI) IS TOTAL RICCATI MATRIX	DIAK 573
C Q = -P*E*P	DIAK 574
C TO SOLVE FOR THE DIFFERENCE P FROM	DIAK 575
C	DIAK 576
C $Q = A * P + P * A + Q$	DIAK 577
C	DIAK 578
DO 3050 I=1,NX	DIAK 579
DO 3050 J=1,NX	DIAK 580
EP(I,I)=Q.	DIAK 581
A(I,J)=AN(I,J)	DIAK 582
DO 3050 K=1,NX	DIAK 583
EP(I,I)=EP(I,J)*E(I,K)*P(K,J)	DIAK 584
3050 A(I,J)=A(I,J)-E(I,K)*(PI(K,J)+P(K,J))	DIAK 585
DO 3052 I=1,NX	DIAK 586
DO 3052 J=1,NX	DIAK 587
Q(I,J)=Q.	DIAK 588
DO 3051 K=1,NX	DIAK 589
3051 Q(I,J)=Q(I,J)+P(I,K)*EP(K,J)	DIAK 590
3052 Q(J,I)=Q(I,J)	DIAK 591
C BEFORE GOING TO THE NEXT ITERATION, CHECK FOR CONVERGENCE	DIAK 592

Figure 84. Program DIAK Program Listing (Continued)


```

C      CONVERGENCE IS WHEN THE ABSOLUTE CHANGE IN THE ELEMENTS OF THE RIDI 593
C      MATRIX BETWEEN ITERATIONS IS LESS THAN THE ABSOLUTE VALUE OF THE DIAK 594
C      TIMES EE 595
C      ONLY CHECK THE UPPER TRIANGULAR ELEMENTS 596
      ICT=0 597
      DO 105 I=1,NX 598
      DO 105 J=I,NX 599
      API=ABS(P(I,J)) 600
      IF(API.LT.1.E-20) GO TO 105 601
C      IF THE ELEMENTS ARE SMALL, CONSIDER THEM AS ZERO AND COUNT THEM AS CDIAK 602
      IF(API.LT.1.E+20) GO TO 888 603
C      IF THE ELEMENTS ARE LARGE, CONSIDER THEM AS DIVERGING AND GO TO NEXTDIAK 604
      WRITE(9,39) 605
      READ(5,1215) IDUM 606
      IF(IDUM.GT.0) STOP 77 607
      READ(5,28) INPD,INPK 608
      IF(INPK.EQ.1) GO TO 1216 609
      IF(INPK.EQ.2.AND.(INPD.EQ.1.OR.INPD.EQ.4)) GO TO 1216 610
C      NEXT RUN IS NOT SOLVABLE WITH PRESENT STARTING GAINS - SO STOP 611
      WRITE(9,3008) 612
      STOP 11 613
888  API=ABS(P(I,J)) 614
      IF(API.LT.1.E-20) GO TO 105 615
      106 RAT=P(I,J)/PI(I,J) 616
      RAT=ABS(RAT) 617
      IF(RAT.EE)105,105,107 618
C      COUNT CONVERGED ELEMENTS 619
      105 ICT=ICT+1 620
      107 CONTINUE 621
C      IF ICT DOES NOT EQUAL NC, THE NUMBER OF ELEMENTS, AND THE NUMBER OF DIAK 622
C      TIONS DOES NOT EQUAL ITER, GO TO NEXT ITERATION 623
      108 IF(NC-ICT)109,122,109 624
      109 IF((ITERC-ITER)/600.1001,100) 625
C      IF ITERC EQUALS ITER, NO CONVERGENCE - PRINT LAST TWO RICCATI MATDIAK 626
C      AND GO TO NEXT RUN 627
      1001 WRITE(9,122)ITER,ICT 628
      120 FORMAT(1H)17X,18H NOT CONVERGED IN 13.34H ITERATIONS-FIRST TERM TODIAK 629
      IFAIL WAS 14/) 630
      ITERM=ITER-1 631
      DO 3054 I=1,NX 632
      DO 3054 J=1,NX 633
      3054 P(I,J)=P(I,J)+PI(I,J) 634
      WRITE(9,121)ITER 635
      121 FORMAT(///23H P MATRIX AT ITERATION 13//) 636
      CALL MP(MX,MX,NX,NX,P) 637
      WRITE(9,121)ITERM 638
      CALL MP(MX,MX,11X,NX,P) 639
      WRITE(9,39) 640
C *** MODIFICATIONS 641
CR  READ(5,1215) IDUM 642
CR  IF(IDUM.GT.0) STOP 77 643
CR  READ(5,28) INPD,INPK 644
CR  IF(INPK.EQ.1) GO TO 1216 645
CR  IF(INPK.EQ.2.AND.(INPD.EQ.1.OR.INPD.EQ.4)) GO TO 1216 646
C      NEXT RUN IS NOT SOLVABLE WITH PRESENT STARTING GAINS - SO STOP 647
CR  WRITE(9,3008) 648
CR  STOP 11 649
C *** MODIFICATIONS 650
      122 CONTINUE 651
C      COMPUTE OPTIMAL GAINS 652
      K = -(D**Q*D):*D**Q*H - (D**Q*D):*G1** 653
      DO 3056 I=1,NX 654
      DO 3056 J=1,NX 655
      3056 P(I,J)=P(I,J)+PI(I,J) 656
      DO 125 I=1,NU 657
      DO 125 J=1,NX 658
      DIAK 659

```

Figure 84. Program DIAK Program Listing (Continued)

AK(I,J)=-W(I,J)	DIAK 659
DO 125 K=1,NX	DIAK 660
125 AK(I,J)=AK(I,J)-W(I,K)*P(K,J)	DIAK 661
C SET COMMAND FEEDFORWARD GAINS TO ZERO	DIAK 662
NXMNC=NX-NCS+1	DIAK 663
DO 86 I=1,NU	DIAK 664
II=I+1F-NU	DIAK 665
DO 86 J=NXMNC,NX	DIAK 666
C *** MODIFICATIONS	DIAK 667
C A(II,J)=0.	DIAK 668
C *** MODIFICATIONS	DIAK 669
86 AK(I,J)=0.	DIAK 670
C *** MODIFICATIONS	DIAK 671
C RECOMPUTE A - CLOSED LOOP STABILITY MATRIX	DIAK 672
REWIND 2	DIAK 673
READ(?) A	DIAK 674
DO 88 I=1,NX	DIAK 675
DO 88 J=1,NX	DIAK 676
DO 88 K=1,NU	DIAK 677
88 A(I,J)=A(I,J)+G(I,K)*AK(K,J)	DIAK 678
C *** MODIFICATIONS	DIAK 679
C PRINT GAINS MATRIX AND RICCATI MATRIX	DIAK 680
4004 FORMAT(1H1/7X,13H GAINS MATRIX//)	DIAK 681
4010 WRITE(9,4005)	DIAK 682
4005 FORMAT(1H1/7X,15H RICCATI MATRIX//)	DIAK 683
CALL MP(MX,MX,NX,NX,P)	DIAK 684
WRITE(9,4004)	DIAK 685
CALL MP(MU,MX,NU,NX,AK)	DIAK 686
C RE-READ H AND M MATRICES FROM DISC	DIAK 687
REWIND 2	DIAK 688
READ(?)	DIAK 689
READ(?) H*AM	DIAK 690
IF (NCONT.LT.2) GO TO 82	DIAK 691
C RECOMPUTE QUADRATIC WEIGHTS ON CONTROL RATES	DIAK 692
NSCSS=NF-NU+1	DIAK 693
DO 80 I=NSCSS,NF	DIAK 694
II=I+NSCRR-NSCSS	DIAK 695
IJ=I-NSCSS+1	DIAK 696
DO 80 J=NSCSS,NF	DIAK 697
JJ=J+NSCRR-NSCSS	DIAK 698
80 QD(II,JJ)=-P(I,J)*G(I,IJ)/(H(JJ,J)*D(II,IJ))	DIAK 699
REWIND 2	DIAK 700
READ(?) F	DIAK 701
NUU=N(I)*NU	DIAK 702
NCU=0	DIAK 703
DO 84 I=1,NU	DIAK 704
DO 84 J=NSCSS,NF	DIAK 705
IF (ABS(AK(I,J)),GT,.05) GO TO 84	DIAK 706
NCU=NCU+1	DIAK 707
84 CONTINUE	DIAK 708
IF (NCU.EQ.NUU) GO TO 85	DIAK 709
IF (NQ.GT.ITERQ) GO TO 85	DIAK 710
NQ=NQ+1	DIAK 711
INPK=3	DIAK 712
INPD=2	DIAK 713
GO TO 81	DIAK 714
85 CONTINUE	DIAK 715
WRITE(6,83) IRUN	DIAK 716
83 FORMAT(17H0 MATRIX FOR CASE,13)	DIAK 717
CALL OUTP(MR,MR,NR,NR,QQ,6)	DIAK 718
82 CONTINUE	DIAK 719
WRITE(6,7776)	DIAK 720
7776 FORMAT(20(4H))	DIAK 721
C PUNCH IDENTIFICATION	DIAK 722
WRITE(6,9010) IRUN	DIAK 723
9010 FORMAT(21HGAINS MATRIX FOR CASE,13)	DIAK 724

Figure 84. Program DIAK Program Listing (Continued)

C	PUNCH OPTIMAL GAINS	DIAK 725
	CALL OUTP(MX,MX,NU,NX,AK,6)	DIAK 726
	WRITE(6,7776)	DIAK 727
C	INVERT M MATRIX (IN P) FOR COMPUTATION OF KSTAR = K*M:	DIAK 728
	DO 892 I=1,NX	DIAK 729
	DO 892 J=1,NX	DIAK 730
892	P(I,J)=AM(I,J)	DIAK 731
	CALL TDINVR(ISOI,IDSOL,NX,NX,P,MX,KWA,DET)	DIAK 732
	IF((ISOI+IDSOL)-2) 889,889,890	DIAK 733
C	IF M MATRIX DOESN'T INVERT, FORGET COMPUTATION OF KSTAR - SKIP TO	DIAK 734
C	RESPONSE CALCULATIONS (STATEMENT 894)	DIAK 735
890	WRITE(9,40)	DIAK 736
40	FORMAT(1H1/7X,32H M MATRIX INVERSE DOES NOT EXIST//7X,10H IGNORE	DIAK 737
	!T//)	DIAK 738
	GO TO 894	DIAK 739
C	COMPUTE KSTAR (IN AN)	DIAK 740
889	DO 1280 I=1,NU	DIAK 741
	DO 1280 J=1,NX	DIAK 742
	AN(I,I)=0.	DIAK 743
	DO 1280 K=1,NX	DIAK 744
1280	AN(I,J)=AN(I,J)+AK(I,K)*P(K,J)	DIAK 745
C	STORE KSTAR IN W1	DIAK 746
	DO 58 I=1,NU	DIAK 747
	DO 58 J=1,NX	DIAK 748
58	W1(I,I)=AN(I,J)	DIAK 749
C	PRINT AND PUNCH KSTAR	DIAK 750
	WRITE(9,1281)	DIAK 751
1281	FORMAT(1H1/7X,13H KSTAR MATRIX//)	DIAK 752
	CALL OP(MX,MX,NU,NX,AN)	DIAK 753
	WRITE(6,9011) TRUN	DIAK 754
9011	FORMAT(,21HKSTAR MATRIX FOR CASE,13)	DIAK 755
	CALL OUTP(MX,MX,NU,NX,AN,6)	DIAK 756
	WRITE(6,7776)	DIAK 757
C	GO TO RESPONSE CALCULATIONS	DIAK 758
	GO TO 894	DIAK 759
893	DO 894 I=1,NU	DIAK 760
	DO 894 J=1,NX	DIAK 761
	W1(I,I)=HK(I,J)	DIAK 762
	AK(I,I)=0.	DIAK 763
	DO 894 K=1,NX	DIAK 764
894	AK(I,J)=AK(I,J)+AK(I,K)*AM(K,J)	DIAK 765
	DO 895 I=1,NX	DIAK 766
	DO 895 J=1,NX	DIAK 767
	A(I,J)=F(I,J)	DIAK 768
	DO 895 K=1,NU	DIAK 769
895	A(I,J)=A(I,J)+G1(I,K)*AK(K,J)	DIAK 770
	WRITE(9,42)	DIAK 771
42	FORMAT(1H1/7X,41H AIRCRAFT RESPONSES WITH PRESCRIBED GAINS//)	DIAK 772
894	DO 4052 I=1,NR	DIAK 773
	DO 4052 J=1,NX	DIAK 774
	DO 4052 K=1,NU	DIAK 775
4052	H(I,J)=H(I,J)+D(I,K)*AK(K,J)	DIAK 776
	GO TO (850,851,851),NOCOV	DIAK 777
851	CONTINUE	DIAK 778
	DO 6080 I=1,NR	DIAK 779
	DO 6080 J=1,NR	DIAK 780
6080	OR(I,I)=0.	DIAK 781
	AJ=0.	DIAK 782
	KCOM=	DIAK 783
6076	KCOM=KCOM+1	DIAK 784
	WRITE(9,41) KCOM	DIAK 785
41	FORMAT(1H1/7X,36H COVARIANCE ANALYSIS FOR DISTURBANCE,13//)	DIAK 786
	DO 4020 I=1,NX	DIAK 787
	DO 4020 J=1,NX	DIAK 788
4020	E(I,J)=G2(I,KCOM)*G2(J,KCOM)	DIAK 789
	DO 4075 I=1,NX	DIAK 790

Figure 84. Program DIAK Program Listing (Continued)

DO 6075 J=1,NX	DIAK 791
6075 P(I,J)=A(I,J)	DIAK 792
CALL CAL1(P,E,0,MWA,NX,MX,IMAX,2,IERR,FE)	DIAK 793
IF(IERR.EQ.0) GO TO 896	DIAK 794
WRITE(9,43)	DIAK 795
43 FORMAT(1H1/7X,26H COVARIANCE MATRIX UNDEFINED//7X,27H IGNORE COVARIANCE ANALYSIS//)	DIAK 796
GO TO 855	DIAK 797
896 WRITE(9,4051)	DIAK 798
4051 FORMAT(/7X,18H COVARIANCE MATRIX//)	DIAK 799
CALL MP(MX,MX,NX,NX,E)	DIAK 800
DO 4053 I=1,NR	DIAK 801
DO 4053 J=1,NX	DIAK 802
AN(I,J)=0.	DIAK 803
DO 4053 K=1,NX	DIAK 804
4053 AN(I,J)=AN(I,J)+H(I,K)*E(K,J)	DIAK 805
DO 4054 I=1,NR	DIAK 806
DO 4054 J=1,NR	DIAK 807
WR(I,J)=0.	DIAK 808
DO 4054 K=1,NX	DIAK 809
4054 WR(I,J)=WR(I,J)+AN(I,K)*H(J,K)	DIAK 810
WRITE(9,4055)	DIAK 811
4055 FORMAT(1H1/7X,27H RESPONSE COVARIANCE MATRIX//)	DIAK 812
CALL MP(MR,MR,NR,NR,WR)	DIAK 813
DO 6077 I=1,NR	DIAK 814
DO 6077 J=1,NR	DIAK 815
OR(I,J)=OR(I,J)+WR(I,J)	DIAK 816
6077 AJ=AJ+WR(I,J)*OO(I,J)	DIAK 817
DO 7015 I=1,NX	DIAK 818
DO 7015 J=1,NX	DIAK 819
P(I,J)=0.	DIAK 820
DO 7015 K=1,NX	DIAK 821
7015 P(I,J)=P(I,J)+E(I,K)*AM(J,K)	DIAK 822
DO 7016 I=1,NX	DIAK 823
DO 7016 J=1,NX	DIAK 824
Q(I,J)=0.	DIAK 825
DO 7016 K=1,NX	DIAK 826
7016 Q(I,J)=Q(I,J)+AM(I,K)*P(K,J)	DIAK 827
WRITE(9,44)	DIAK 828
44 FORMAT(1H1/7X,26H MEASUREMENT COVARIANCE MATRIX//)	DIAK 829
CALL MP(MX,MX,NX,NX,0)	DIAK 830
DO 1112 I=1,NU	DIAK 831
DO 1112 L=1,NX	DIAK 832
W(I,L)=0.	DIAK 833
DO 1112 K=1,NX	DIAK 834
1112 W(I,L)=W(I,L)+AK(I,K)*E(K,L)	DIAK 835
DO 6085 I=1,NU	DIAK 836
DO 6085 J=1,NU	DIAK 837
DDD(I,J)=0.	DIAK 838
DO 6085 K=1,NX	DIAK 839
6085 DDD(I,J)=DDD(I,J)+W(I,K)*AK(J,K)	DIAK 840
WRITE(9,45)	DIAK 841
45 FORMAT(1H1/7X,26H CONTROL COVARIANCE MATRIX//)	DIAK 842
CALL MP(MU,MU,NU,NU,DDD)	DIAK 843
IF(NOCOV.GT.2) GO TO 2	DIAK 844
DO 1111 I=1,NX	DIAK 845
DO 1111 J=1,NX	DIAK 846
P(I,J)=0.	DIAK 847
IF(E(I,I).LT.1.E-20) GO TO 1111	DIAK 848
IF(E(I,J).LT.1.E-20) GO TO 1111	DIAK 849
P(I,J)=E(I,J)/SQRT(E(I,I)*E(J,J))	DIAK 850
1111 CONTINUE	DIAK 851
WRITE(9,46)	DIAK 852
46 FORMAT(1H1/7X,31H STATE CROSS-CORRELATION MATRIX//)	DIAK 853
CALL MP(MX,MX,NX,NX,P)	DIAK 854
DO 1113 I=1,NU	DIAK 855
	DIAK 856

Figure 84. Program DIAK Program Listing (Continued)

DO 1113 J=1,NX	DIAK 857
P(I,J)=0.	DIAK 858
P(I,J)=W(I,J)*AK(I,J)	DIAK 859
1113 CONTINUE	DIAK 860
WRITE(9,47)	DIAK 861
47 FORMAT(1H1/7X,41H CONTROL-STATE ROW-SUM CORRELATION MATRIX//)	DIAK 862
CALL MP(MX,MX,NU,NX,P)	DIAK 863
DO 1114 I=1,NR	DIAK 864
DO 1114 J=1,NX	DIAK 865
P(I,J)=0.	DIAK 866
P(I,J)=AN(I,J)*H(I,J)	DIAK 867
1114 CONTINUE	DIAK 868
WRITE(9,48)	DIAK 869
48 FORMAT(1H1/7X,42H RESPONSE-STATE ROW-SUM CORRELATION MATRIX//)	DIAK 870
CALL MP(MX,MX,NR,NX,P)	DIAK 871
DO 1115 I=1,NX	DIAK 872
DO 1115 J=1,NX	DIAK 873
P(I,J)=0.	DIAK 874
IF(Q(I,I).LT.1.E-27) GO TO 1115	DIAK 875
IF(Q(J,J).LT.1.E-27) GO TO 1115	DIAK 876
P(I,J)=Q(I,J)/SQRT(Q(I,I)*Q(J,J))	DIAK 877
1115 CONTINUE	DIAK 878
WRITE(9,49)	DIAK 879
49 FORMAT(1H1/7X,37H MEASUREMENT CROSS-CORRELATION MATRIX//)	DIAK 880
CALL MP(MX,MX,NX,NV,P)	DIAK 881
DO 1300 I=1,NX	DIAK 882
DO 1300 J=1,NX	DIAK 883
P(I,J)=0.	DIAK 884
DO 1300 K=1,NX	DIAK 885
1301 P(I,J)=P(I,J)+AM(I,K)*F(K,J)	DIAK 886
P(I,J)=P(I,J)*AM(I,J)	DIAK 887
1300 CONTINUE	DIAK 888
WRITE(9,13(2))	DIAK 889
1302 FORMAT(1H1/7X,45H MEASUREMENT-STATE ROW-SUM CORRELATION MATRIX//)	DIAK 890
CALL MP(MX,MX,NX,NX,P)	DIAK 891
DO 1116 I=1,NU	DIAK 892
DO 1116 J=1,NX	DIAK 893
P(I,J)=0.	DIAK 894
DO 1117 K=1,NX	DIAK 895
1117 P(I,J)=P(I,J)+W(I,K)*AM(J,K)	DIAK 896
P(I,J)=P(I,J)*W1(I,J)	DIAK 897
1116 CONTINUE	DIAK 898
WRITE(9,50)	DIAK 899
50 FORMAT(1H1/7X,47H CONTROL-MEASUREMENT ROW-SUM CORRELATION MATRIX//)	DIAK 900
1)	DIAK 901
CALL MP(MX,MX,NU,NX,P)	DIAK 902
CONTINUE	DIAK 903
DO 63 I=1,NX	DIAK 904
63 Q(I,I)=SQRT(Q(I,I))	DIAK 905
DO 64 I=1,NU	DIAK 906
64 QDD(I,I)=SQRT(QDD(I,I))	DIAK 907
WRITE(9,65) ((I,QDD(I,I)),I=1,NU)	DIAK 908
65 FORMAT(1H1//20X,16H R.M.S. CONTROLS/(5X,13,13X,E15.8))	DIAK 909
WRITE(9,66) ((I,Q(I,I)),I=1,NX)	DIAK 910
66 FORMAT(1H1//20X,20H R.M.S. MEASUREMENTS/(5X,13,13X,E15.8))	DIAK 911
DO 4056 I=1,NP	DIAK 912
4056 WR(I,I)=SQRT(WR(I,I))	DIAK 913
WRITE(9,4057) ((I,WR(I,I)),I=1,NP)	DIAK 914
4057 FORMAT(1H1//20X,17H R.M.S. RESPONSES/(5X,13,13X,E15.8))	DIAK 915
IF(KCON.LT.NN) GO TO 6076	DIAK 916
WRITE(9,63)	DIAK 917
66 FORMAT(1H1/7X,37H TOTAL RESPONSE COVARIANCE MATRIX//)	DIAK 918
CALL MP(MR,MR,NR,NP,OP)	DIAK 919
DO 61 I=1,NR	DIAK 920
DO 61 J=1,NR	DIAK 921
P(I,J)=0.	DIAK 922

Figure 84. Program DIAK Program Listing (Continued)

	IF (OR(I,I).LT.1.F-20) GO TO 61	DIAK 923
	IF (OR(J,J).LT.1.F-20) GO TO 61	DIAK 924
	P(I,J)=QR(I,J)/SQRT(QR(I,I)*Q(J,J))	DIAK 925
61	CONTINUE	DIAK 926
	WRITE(9,62)	DIAK 927
62	FORMAT(1H1/7X,40H TOTAL RESPONSE CROSS-CORRELATION MATRIX//)	DIAK 928
	CALL MP(MX,MX,NR,ND,P)	DIAK 929
	DO 6082 I=1,NR	DIAK 930
6082	QR(I,1)=SQRT(QR(I,1))	DIAK 931
	WRITE(9,6081) ((I,QR(I,I)),I=1,NR)	DIAK 932
6081	FORMAT(1H1/7X,22HTOTAL R.M.S. RESPONSES/(7X,13,3X,E15.8))	DIAK 933
	WRITE(9,6078) AJ	DIAK 934
6078	FORMAT(1H1/20X,17HQUADRATIC COST = ,E15.8)	DIAK 935
850	CONTINUE	DIAK 936
	IF(NOP.EQ.1) GO TO 897	DIAK 937
	IF(NSTEP.EQ.0.AND.NRAND.EQ.1) GO TO 897	DIAK 938
	CALL TIMER(A,G2,H,X,X1,DX,DX1,XL,DL,GN,GS,R,IPLR,ITITL,IUNIT,CL,	DIAK 939
	IT,DT,NT,YMAX,YMIN,IFLT,IRUN,ICATE,NSTEP,NRAND,NPLOT,NPRIN,NN,NX,NF	DIAK 940
	2,NG,NDS,NR,MXR,MN,IX,XPOIN,NOP,NAME1,NAME2,SCAL,NEWY,T1,T2,VGLG)	DIAK 941
897	CONTINUE	DIAK 942
	CALL POLES(NX,A,MX,RR,M)	DIAK 943
	ITEPC=0	DIAK 944
1200	READ(4,1215) IDUM	DIAK 945
	IF(IDUM.GT.0) GO TO 7777	DIAK 946
1215	FORMAT(2I1)	DIAK 947
	READ(4,28) INPD,INPK	DIAK 948
1216	IRUN=IRUN+1	DIAK 949
	REWIND 2	DIAK 950
	READ(2) F	DIAK 951
	READ(2) H,AM	DIAK 952
	READ(4,28) NCONT	DIAK 953
	READ(4,28) NOCOV,NSTEP,NRAND,NPRIN,NPLOT	DIAK 954
	WRITE(9,37) INPD,INPK,NCONT,NCOCV,NSTEP,NRAND,NPRIN,NPLOT	DIAK 955
	IF(INPD.EQ.1) GO TO 1216	DIAK 956
	WRITE(9,1271) IFLT,IRUN	DIAK 957
	GO TO (121,122,123,124,125),INPD	DIAK 958
7777	CONTINUE	DIAK 959
	ENDIF 6	DIAK 960
	END	DIAK 961

Figure 84. Program DIAK Program Listing (Concluded)

```

OVERL Y(KON2,P..) FFOC 2
PROGRAM FFOC FFOC 3
DIMENSION F(40,40),G(40,50),H(40,40),I(40,40) FFOC 4
* D(40,6),AM(4,40),Q(40,40),K(06,40),JK(06,40),DJDK(06,40) FFOC 5
* AKG(6,40),XI(9,9),HK(6,40),A(40,40),X(4,40),C(40,40) FFOC 6
* P(40,40),MDK(4,40),S(40,4),PR(608),DELK(06,40),AKP(06,40) FFOC 7
DIMENSION U(40,40),V(9,9),E(9,9),FS(40,40),DQ(6,40) FFOC 8
* DDD(66,6),KKA(50,5),T(50,5),RR(50,40),DUV(50),DUVT(50),IF(50) FFOC 9
* JF(5),AMT(50,40),Y(40,40),Z(40,40),NORD(40) FFOC 10
C FFOC 11
C ARRAY DIMENSIONS FFOC 12
C FFOC 13
MX=40 FFOC 14
MR=40 FFOC 15
MNN=2 FFOC 16
MU=6 FFOC 17
MM=40 FFOC 18
MFF=9 FFOC 19
MFR=4 FFOC 20
MF=50 FFOC 21
C INPUT INTEGER PARAMETERS FFOC 22
READ(5,1) IMAX,NITM,NOPR,NOCOV,NBEGIN ,NDIAK FFOC 23
READ(5,1) NX,NR,NU,NN,NFF,NF FFOC 24
1 FORMAT(4,I2) FFOC 25
READ(5,1) (NORD(I),I=1,NX) FFOC 26
ITER= FFOC 27
NM=NX FFOC 28
WRITE(9,9) IMAX,NITM FFOC 29
9 FORMAT(1H1/7X,29H MAXIMUM NO. OF INNER LOOP ITERATIONS =,I3//7X, FFOC 30
139H MAXIMUM NO. OF OUTER LOOP ITERATIONS =,I3//) FFOC 31
WRITE(9,11) NOCOV,NBEGIN,NOPR FFOC 32
11 FORMAT(//7X,8H NOCOV =,I3,5X,9H NBEGIN =,I3,5X,7H NOPR =,I3//) FFOC 33
WRITE(9,13) NX,NR,NU,NN,NFF,NF FFOC 34
13 FORMAT(//7X,16H NO. OF STATES =,I3,5X,19H NO. OF RESPONSES =,I3// FFOC 35
17X,18H NO. OF CONTROLS =,I3,5X,22H NO. OF DISTURBANCES =,I3// FFOC 36
27X,29H NO. OF FEEDFORWARD STATES =,I3//7X,26H NO. OF FIXED-FORM GFFOC 37
3AINS =,I3//) FFOC 38
C FFOC 39
C INITIAL STEP SIZE FFOC 40
60 READ(5,1277) EPSI FFOC 41
C FFOC 42
C INPUT REAL PARAMETERS FFOC 43
READ(5,1277) AJSTAR FFOC 44
AJT=AJSTAR FFOC 45
WRITE(9,5035) FFOC 46
5035 FORMAT( //7X,29H LOWEST COST EXPECTED(AJSTAR)/) FFOC 47
WRITE(9,5033) AJSTAR FFOC 48
5033 FORMAT(//7X,(6G10,4)) FFOC 49
WRITE(9,2) (NORD(I),I=1,NX) FFOC 50
2 FORMAT(//7X,27H STATES ARE ORDERED AS SUCH// (7X,30I3)) FFOC 51
READ(5,1277) DROC FFOC 52
READ(5,1277) ALAM,DELT,ALAMD FFOC 53
1277 FORMAT(6G10,4) FFOC 54
3 NG=1 FFOC 55
NC=1 FFOC 56
NUNST=0 FFOC 57
C FFOC 58
C ROWS AND COLUMNS OF FIXED GAINS -- K1 FFOC 59
READ(5,1) (JF(I),JF(I),I=1,NF) FFOC 60
WRITE(9,20) FFOC 61
20 FORMAT(//7X,29H FIXED GAINS ROW COLUMN//) FFOC 62
WRITE(9,201) (JF(I),JF(I),I=1,NF) FFOC 63
201 FORMAT(21X,216) FFOC 64

```

Figure 85. Program FFOC Program Listing

C		FFOC 65
C	INPUT SYSTEM MATRICES -- F,G1,G2,H,D,M,Q	FFOC 66
	CALL ZERO (F,MX,MX)	FFOC 67
	CALL ZERO (G1,MX,MU)	FFOC 68
	CALL ZERO (G2,MX,MNN)	FFOC 69
	CALL ZERO (H,MR,MX)	FFOC 70
	CALL ZERO (D,MR,MU)	FFOC 71
	CALL ZERO (AM,MM,MX)	FFOC 72
	CALL ZERO (Q,MR,MR)	FFOC 73
	CALL INPT (F,MX,MX)	FFOC 74
	CALL INPT (G1,MX,MU)	FFOC 75
	CALL INPT (G2,MX,MNN)	FFOC 76
	CALL INPT (H,MR,MX)	FFOC 77
	CALL INPT (D,MR,MU)	FFOC 78
	CALL INPT (AM,MM,MX)	FFOC 79
	CALL INPT (Q,MR,MR)	FFOC 80
C	INPUT OPTIMAL GAINS	FFOC 81
	CALL ZERO (AKG,MU,MX)	FFOC 82
	CALL INPT (AKG,MU,MX)	FFOC 83
	CALL SHUF (F,G1,G2,H,AM,AKG,Y,NORD,MX,NX,MR,VR,VM,NU,MU,NU,MVN,NN)	FFOC 84
	WRITE (9,1010)	FFOC 85
1010	FORMAT (1H1/7X,10H F MATRIX//)	FFOC 86
	CALL MP (MX,MX,NX,NX,F)	FFOC 87
	WRITE (9,1011)	FFOC 88
1011	FORMAT (1H1/7X,10H G1 MATRIX//)	FFOC 89
	CALL MP (MX,MN,NX,NU,G1)	FFOC 90
	WRITE (9,1012)	FFOC 91
1012	FORMAT (1H1/7X,10H G2 MATRIX//)	FFOC 92
	CALL MP (MX,MNN,NX,NU,G2)	FFOC 93
	WRITE (9,1013)	FFOC 94
1013	FORMAT (1H1/7X,10H H MATRIX//)	FFOC 95
	CALL MP (MR,MX,NR,NX,H)	FFOC 96
	WRITE (9,1014)	FFOC 97
1014	FORMAT (1H1/7X,10H D MATRIX//)	FFOC 98
	CALL MP (MR,MN,NR,NU,D)	FFOC 99
	WRITE (9,1015)	FFOC 100
1015	FORMAT (1H1/7X,10H M MATRIX//)	FFOC 101
	CALL MP (MM,MX,NM,NX,AM)	FFOC 102
	WRITE (9,1016)	FFOC 103
1016	FORMAT (1H1/7X,10H Q MATRIX//)	FFOC 104
	CALL MP (MR,MR,NR,NR,Q)	FFOC 105
	CALL ZERO (AKP,MU,MM)	FFOC 106
	L=1	FFOC 107
	DO 1750 I=1,NU	FFOC 108
	DO 1750 J=1,NM	FFOC 109
	IF (L.GT.NF) GO TO 1750	FFOC 110
	IF (I.NE.IF(L)) GO TO 1750	FFOC 111
	IF (J.NE.JF(L)) GO TO 1750	FFOC 112
	AKP(I,J)=1.	FFOC 113
	DO 1751 N=1,NX	FFOC 114
1751	AMT(L,N)=AM(J,N)	FFOC 115
	L=L+1	FFOC 116
1750	CONTINUE	FFOC 117
	WRITE (9,1017)	FFOC 118
1017	FORMAT (1H1/7X,40H MEASUREMENT MATRIX FOR FIXED FORM GAINS//)	FFOC 119
	CALL MP (MF,MX,NF,NX,AMT)	FFOC 120
	WRITE (9,1500)	FFOC 121
1500	FORMAT (1H1/7X,22H OPTIMAL RICCATI GAINS//)	FFOC 122
	CALL MP (MU,MX,NU,NM,AKG)	FFOC 123
	DO 1501 I=1,NM	FFOC 124
	DO 1501 J=1,NX	FFOC 125
1501	S(I,J)=AM(I,J)	FFOC 126
	CALL TOINVR (ISOL,INSOL,NX,NX,S,MX,KWA,DE1)	FFOC 127
	IF ((ISOL+IDSOL).LE.?) GO TO 1502	FFOC 128
	WRITE (9,1403) ISOL,IDSOL	FFOC 129
1403	FORMAT (//7X,51H MEASUREMENT MATRIX IS NOT INVERTIBLE, ERROR CODE =	FFOC 130

Figure 85. Program FFOC Program Listing (Continued)

1.213//)		FFOC 131
STOP 13		FFOC 132
1502 DO 14 5 I=1,NI		FFOC 133
DO 14 5 J=1,NX		FFOC 134
AK(I, J)=0.		FFOC 135
DO 14 5 K=1,NX		FFOC 136
1405 AK(I, J)=AK(I, J)+AKG(I, K)*S(K, J)		FFOC 137
C		FFOC 138
C DEFINE K2		FFOC 139
CALL INPT(AK,MI,MM)		FFOC 140
DO 14 4 I=1,NI		FFOC 141
DO 14 4 J=1,NX		FFOC 142
RK(I, J)=AK(I, J)*(1.-AKP(I, J))		FFOC 143
1404 AK(I, J)=AK(I, J)*AKP(I, J)		FFOC 144
CALL INPT(RK,MI,MM)		FFOC 145
WRITE(9,1018)		FFOC 146
1018 FORMAT(1H1//7X,23H INITIAL GAINS -- K1(LAMBDA)//)		FFOC 147
CALL MP(MU, MX, NI, NI, AK)		FFOC 148
WRITE(9,1019)		FFOC 149
1019 FORMAT(//7X,10H K2 MATRIX//)		FFOC 150
CALL MP(MI, MM, NI, NI, RK)		FFOC 151
IF(ALAM.LT..99) GO TO 1406		FFOC 152
DO 14 7 I=1,NI		FFOC 153
DO 14 7 J=1,NM		FFOC 154
1407 DELK(I, J)=0.		FFOC 155
GO TO 1408		FFOC 156
C		FFOC 157
C INPUT PRESENT FIXED GAINS -- K1(LAMBDA)		FFOC 158
1406 CALL ZERO(AK,MI,MM)		FFOC 159
CALL INPT(AK,MI,MM)		FFOC 160
WRITE(9,1020)		FFOC 161
1020 FORMAT(//7X,34H PRESENT FIXED GAINS -- K1(LAMBDA)//)		FFOC 162
CALL MP(MU, MM, NI, NI, AK)		FFOC 163
C		FFOC 164
C INPUT FIXED PREDICTOR -- DELK1(LAMBDA)		FFOC 165
CALL ZERO(DELK, MU, MM)		FFOC 166
CALL INPT(DELK, MU, MM)		FFOC 167
1408 WRITE(9,1021)		FFOC 168
1021 FORMAT(//7X,35H PRESENT PREDICTOR -- DELK1(LAMBDA)//)		FFOC 169
CALL MP(MU, MM, NI, NI, DELK)		FFOC 170
C		FFOC 171
C TAKE STEP IN LAMBDA		FFOC 172
172 ALAM=ALAM-DELT		FFOC 173
WRITE(9,173) ALAM		FFOC 174
173 FORMAT(//7X,9HLAMBDA = .F9.3)		FFOC 175
C		FFOC 176
C PREDICT GAINS FOR NEW LAMBDA		FFOC 177
DO 31 I=1,NI		FFOC 178
DO 31 J=1,NM		FFOC 179
DK(I, J)=DELK(I, J)		FFOC 180
DELK(I, J)=AK(I, J)		FFOC 181
310 AK(I, J)=AK(I, J)+DK(I, J)		FFOC 182
C		FFOC 183
C INITIAL CONDITIONS		FFOC 184
C		FFOC 185
NL=1		FFOC 186
LAST=0		FFOC 187
NCHK=		FFOC 188
NGD=0		FFOC 189
NIT=0		FFOC 190
EPS=EPSI		FFOC 191
AJJ=1.*AJT		FFOC 192
7 AJL=AJT		FFOC 193
WRITE(9,4051) NIT		FFOC 194
4051 FORMAT(//7X,9HITERATION,13)		FFOC 195
WRITE(9,59) EPS		FFOC 196

Figure 85. Program FFOC Program Listing (Continued)

C		FFOC 197
C	PRINT GAINS	FFOC 198
	WRITE(9,4004)	FFOC 199
4004	FORMAT(//7X,13H GAINS MATRIX//)	FFOC 200
	CALL MP(MU,MM,NU,NM,AK)	FFOC 201
C		FFOC 202
C	INITIALIZE GRADIENT PROJECTION	FFOC 203
	IF(NG,NE,1) GO TO 5	FFOC 204
	DO 16 I=1,NU	FFOC 205
	DO 16 J=1,NM	FFOC 206
16	AKP(I,J)=AK(I,J)	FFOC 207
C		FFOC 208
C	INITIALIZE ARRAYS	FFOC 209
C		FFOC 210
C	COMPUTE F+G)*K*(LAMBDA)*M=A	FFOC 211
C		FFOC 212
	5 DO 12 J=1,NX	FFOC 213
	DO 12 I=1,NU	FFOC 214
	C(I,J)=0.	FFOC 215
	DO 12 K=1,NM	FFOC 216
	C(I,J)=C(I,J)+(AK(I,K)*PK(I,K)*ALAM)*AM(K,J)	FFOC 217
12	CONTINUE	FFOC 218
	IF(LAST,NE,1) GO TO 66	FFOC 219
	DO 15 I=1,NU	FFOC 220
	DO 15 J=1,NM	FFOC 221
151	DJDK(I,J)=AK(I,J)*ALAM*PK(I,J)	FFOC 222
	WRITE(9,1022)	FFOC 223
1022	FORMAT(//7X,37H K*(LAMBDA) FOR RESPONSE CALCULATIONS//)	FFOC 224
	CALL MP(MU,MM,NU,NM,DJDK)	FFOC 225
66	CONTINUE	FFOC 226
	DO 8 I=1,NX	FFOC 227
	DO 8 J=1,NX	FFOC 228
	A(I,J)=F(I,J)	FFOC 229
	DO 8 K=1,NU	FFOC 230
	A(I,J)=A(I,J)+G(I,K)*C(K,J)	FFOC 231
	IF(NL,NE,1) GO TO 144	FFOC 232
C		FFOC 233
C	CHECK FOR STABILITY OF A	FFOC 234
	CALL POLES(NX,A,MX,RR,M)	FFOC 235
	KK=0	FFOC 236
	II=1	FFOC 237
921	IF(RR(II),LT,0.) GO TO 185	FFOC 238
C		FFOC 239
C	IF UNSTABLE -- HALVE DELTA LAMBDA AND PREDICTOR	FFOC 240
69	I=NUNST+1	FFOC 241
	WRITE(9,188) I	FFOC 242
188	FORMAT(//7X,24H UNSTABLE -- CHANGE GAINS//7X,13,14H INSTABILITY//)	FFOC 243
	IF(NUNST,EQ,3) GO TO 5071	FFOC 244
	ALAM=ALAM*DELT	FFOC 245
	IF(NUNST,EQ,1) GO TO 5071	FFOC 246
C		FFOC 247
C	FIRST OR THIRD INSTABILITY -- HALVE PREDICTOR	FFOC 248
	DO 5040 I=1,NU	FFOC 249
	DO 5040 J=1,NM	FFOC 250
	AK(I,J)=AK(I,J)-OK(I,J)	FFOC 251
5040	DELK(I,J)=OK(I,J)/2.	FFOC 252
	AJT=A/STAR	FFOC 253
	NUNST=NUNST+1	FFOC 254
	GO TO 172	FFOC 255
C		FFOC 256
C	SECOND OR FOURTH INSTABILITY -- HALVE DELTA LAMBDA	FFOC 257
5071	DELT=DELT*.5	FFOC 258
C	*** MODIFICATIONS	FFOC 259
	IF(DELT,LE,1.0E-06)WRITE(9,7740)DELT	FFOC 260
7740	FORMAT(1H1//,1X,51H*** EXIT ON DETECTING VERY SMALL VALUE FOR DELT	FFOC 261
	IT ***.G10.4)	FFOC 262

Figure 85. Program FFOC Program Listing (Continued)

```

      IF (DELT.LE.1.0E-06) GO TO 1730
C *** MODIFICATIONS
      DO 5072 I=1,NU
      DO 5072 J=1,NM
      DELK(I,J)=DK(I,J)
5072 AK(I,J)=AK(I,J)-DK(I,J)
C
C STOP AND PUNCH GAINS AND PREDICTOR ON FOURTH INSTABILITY
      IF (NUNST.EQ.3) GO TO 5041
C *** MODIFICATIONS
      ALAMD=ALAMD*DELT
C *** MODIFICATIONS
      AJT=A*STAR
      NUNST=2
      GO TO 177
185 IF (RR((I+1).EQ.0. ) GO TO 922
      KK=KK*2
      GO TO 923
922 KK=KK*1
923 IF (KK.EQ.N ) GO TO 924
      II=II*2
      GO TO 921
924 CONTINUE
C
C RECOMPUTE A
      DO 187 I=1,NX
      DO 187 J=1,NX
      A(I,J)=F(I,J)
      DO 187 K=1,NU
187 A(I,J)=A(I,J)+G(I,K)*C(K,J)
C
C COMPUTE H*DKM. R=(H*DKM)*Q(H*DKM). DO
184 DO 7012 I=1,NR
      DO 7012 J=1,NX
      HDK(I,J)=H(I,J)
      DO 7012 K=1,NU
7012 HDK(I,J)=HDK(I,J)+D(I,K)*C(K,J)
      IF (LAST.EQ.1) GO TO 126
      DO 7013 I=1,NR
      DO 7013 J=1,NX
      C(I,J)=0.
      DO 7013 K=1,NR
7013 C(I,J)=C(I,J)+D(I,K)*HDK(K,J)
      DO 7014 I=1,NX
      DO 7014 J=1,NX
      R(I,J)=0.
      DO 7014 K=1,NR
7014 R(I,J)=R(I,J)+HDK(K,I)*C(K,J)
      DO 18 I=1,NU
      DO 18 J=1,NR
      DD(I,J)=0.
      DO 18 K=1,NR
18 DD(I,J)=DD(I,J)+D(K,I)*C(K,J)
      DO 434 I=1,NU
      DO 434 J=1,NU
      DDD(I,J)=0.
      DO 434 K=1,NR
434 DDD(I,J)=DDD(I,J)+DD(I,K)*D(K,J)
C
C COMPUTE COVARIANCE MATRIX
      CALL COVAR(XI,A,C,x,G2,S,E,FS,V,U,NX,NFF,VN,MX,NFF,MFR,MNN,IMAX,
      ITER,IERR,KWA)
      IF (IEPR.EQ.0) GO TO 17
      IF (NL.NE.1) GO TO 4
      DO 6 I=1,NU
      DO 6 J=1,NX

```

Figure 85. Program FFOC Program Listing (Continued)

	C(I,J)=0.	FFOC 329
	DO 6 K=1,NM	FFOC 330
6	C(I,J)=C(I,J)+(AK(I,K)+BK(I,K)*ALAM)*AM(K,J)	FFOC 331
	GO TO 69	FFOC 332
4	CONTINUE	FFOC 333
	AJT=1.F*20	FFOC 334
	IF(NG.NE.1) GO TO 43	FFOC 335
	GO TO 1756	FFOC 336
	17 ITER=	FFOC 337
C		FFOC 338
C	CALCULATE COST	FFOC 339
C		FFOC 340
	AJT=0.	FFOC 341
	DO 34 I=1,NX	FFOC 342
	DO 34 J=1,NX	FFOC 343
34	AJT=AJT+P(I,J)*X(I,J)	FFOC 344
	WRITE(9,122) AJT	FFOC 345
122	FORMAT(//7X,4H COST = ,E15.3)	FFOC 346
	IF(NM.EQ.0) GO TO 10	FFOC 347
	IF(AJT.LT.0.) GO TO 1510	FFOC 348
	IF(AJT.LT.AJT) GO TO 10	FFOC 349
1510	WRITE(9,1402)	FFOC 350
1402	FORMAT(//7X,43H COST EXCEEDS 10 TIMES LOWEST COST EXPECTED//)	FFOC 351
	WRITE(9,14)	FFOC 352
14	FORMAT(1H1/7X,18H COVARIANCE MATRIX//)	FFOC 353
	CALL MP(MX,MX,NX,NX,X)	FFOC 354
	STOP 11	FFOC 355
	19 IF(NG.NE.1) GO TO 450	FFOC 356
130	IF(NG1.EQ.2) GO TO 67	FFOC 357
C		FFOC 358
C	COMPUTE XM, MX INVERSE	FFOC 359
	DO 125 I=1,NX	FFOC 360
	DO 125 J=1,NM	FFOC 361
	C(I,J)=0.	FFOC 362
	DO 125 K=1,NX	FFOC 363
125	C(I,J)=C(I,J)+X(I,K)*AM(J,K)	FFOC 364
	CALL TRANS(AMT,X,T,DD,NX,MX,NF,MU,MF,RR,IF)	FFOC 365
C		FFOC 366
C	COSTATE CALCULATION	FFOC 367
C		FFOC 368
	CALL COSTAT (R,A,S,X,ES,Y,Z,E,U,V,KWA,MX,MFR,MFF,NX,NFF,IMAX,IERR)	FFOC 369
	IF(IERR.EQ.1) GO TO 1400	FFOC 370
	WRITE(9,1401)	FFOC 371
1401	FORMAT(//7X,45H COSTATE MATRIX UNSTABLE WHEN STATE MATRIX IS//)	FFOC 372
	STOP 11	FFOC 373
C		FFOC 374
C	GRADIENT CALCULATION	FFOC 375
C		FFOC 376
C		FFOC 377
C	COMPUTE DD(H+DKM)XM(MXM)-1 -- 2G1*SKM(MXM)-1	FFOC 378
1400	DO 19 I=1,NU	FFOC 379
	DO 19 J=1,NX	FFOC 380
	R(I,J)=0.	FFOC 381
	DO 19 K=1,NR	FFOC 382
19	R(I,J)=R(I,J)+DD(I,K)*HDK(K,J)	FFOC 383
	DO 30 I=1,NU	FFOC 384
	DO 30 J=1,NX	FFOC 385
	X(I,J)=0.	FFOC 386
	DO 30 K=1,NX	FFOC 387
30	X(I,J)=X(I,J)+G1(K,I)*S(K,J)	FFOC 388
	DO 126 I=1,NU	FFOC 389
	DO 126 J=1,NM	FFOC 390
	DJDK(I,J)=0.	FFOC 391
	DO 126 K=1,NX	FFOC 392
126	DJDK(I,J)=DJDK(I,J)+(R(I,K)+X(I,K))*C(K,J)*2.	FFOC 393
	GO TO 850	FFOC 394

Figure 85. Program FFOC Program Listing (Continued)

67	CONTINUE	FFOC 395
C		FFOC 396
C	PROJECTED GRADIENT	FFOC 397
	DO 104 I=1,NU	FFOC 398
	DO 104 J=1,NM	FFOC 399
104	DJDK(I,J)=AKP(I,J)-AK(I,J)	FFOC 400
	GO TO 850	FFOC 401
C		FFOC 402
C	CALCULATE RESPONSES	FFOC 403
C		FFOC 404
120	CONTINUE	FFOC 405
	GO TO (853,851,851),NOCOV	FFOC 406
851	CALL PESPIC(A,G2,AM,DJDK,Y,Y,Z,S,R,ES,E,U,V,XI,DDQ,AKG,DDQ,HDK,KWA,	FFOC 407
	INX,NFF,NN,NM,NU,NR,MX,MFF,MFB,MNN,MM,MU,MR,(ITER,IMAX,ITER,NOCOV)	FFOC 408
	ITER=	FFOC 409
853	CALL POLES(INX,A,MX,RR,M)	FFOC 410
850	CONTINUE	FFOC 411
	IF(LAST.EQ.1) GO TO 57	FFOC 412
	IF(NL.EQ.1) AJLAT=AJT	FFOC 413
	NREG1 = 0	FFOC 414
	IF(NG.NE.1) GO TO 43	FFOC 415
	IF(NG0.EQ.2) GO TO 1756	FFOC 416
	CALL TDINVR(ISOL,IDSOL,NF,NF,T,MF,KWA,DET)	FFOC 417
	IF((ISOL+IDSOL).LE.2) GO TO 852	FFOC 418
	WRITE(9,1023)	FFOC 419
1023	FORMAT(//7X,39H GRADIENT TRANSFORMATION NOT INVERTABLE//)	FFOC 420
	STOP **	FFOC 421
852	WRITE(9,1024)	FFOC 422
1024	FORMAT(//7X,31H GRADIENT TRANSFORMATION MATRIX//)	FFOC 423
	CALL INSCR(T,DJDK,DJV,DJVT,(F,JF,NF,VJ,NM,MU,MM,MF)	FFOC 424
	CALL POLES(NF,T,MF,RR,M)	FFOC 425
1756	IF(NL.GT.0) GO TO 500	FFOC 426
	IF(NG4R.GT.0) GO TO 500	FFOC 427
C		FFOC 428
C	CORRECT FOR INSTABILITY WHILE COMPUTING GRADIENT	FFOC 429
	NGH=1	FFOC 430
	IF(AJT.GT.AJM) NGH=0	FFOC 431
	IF(AJT.LT.0.) NGH=0	FFOC 432
	IF(NG0.NE.0) GO TO 500	FFOC 433
	NGBR=1	FFOC 434
	EPS=EPS0	FFOC 435
	DO 501 I=1,NU	FFOC 436
	DO 501 J=1,NM	FFOC 437
	AK(I,J)=AK(I,J)-DK(I,J)	FFOC 438
	DK(I,J)=EPS0*DK(I,J)/FPS	FFOC 439
	IF(NG0.EQ.2) AK(I,J)=AK(I,J)+2.*DK(I,J)	FFOC 440
	IF(NG0.EQ.3) AK(I,J)=AK(I,J)+.5*DK(I,J)	FFOC 441
	IF(NG0.EQ.5) AK(I,J)=AK(I,J)+DK(I,J)	FFOC 442
501	CONTINUE	FFOC 443
	GO TO 7	FFOC 444
500	IF(NCHK.EQ.0) GO TO 5006	FFOC 445
C		FFOC 446
C	COMPUTE RATIO OF COSTS	FFOC 447
	ROC=AJT/AJLAT	FFOC 448
	WRITE(9,5030) ROC	FFOC 449
5030	FORMAT(//7X,17H RATIO OF COSTS = .F10.4)	FFOC 450
	IF(ROC.GT.DROC) LAST=1	FFOC 451
	AJLAT=AJT	FFOC 452
	IF(NIT.GE.NITM) LAST=1	FFOC 453
C		FFOC 454
C	NORMALIZE GRADIENT AND COMPUTE DELTA GAINS	FFOC 455
C		FFOC 456
5006	SUM=0.	FFOC 457
5008	DO 40 I=1,NU	FFOC 458
	DO 40 J=1,NM	FFOC 459
	40 SUM=SUM+DJDK(I,J)*DJDK(I,J)	FFOC 460

Figure 85. Program FFOC Program Listing (Continued)

	SUM=SQRT(SUM)	FFOC 461
	WRITE(9,5031) SUM	FFOC 462
5031	FORMAT(//,7X,16HGRADIENT NORM = ,F15.3)	FFOC 463
5009	DO 103 I=1,NU	FFOC 464
	DO 103 J=1,NM	FFOC 465
103	DJDK(I,J)=DJDK(I,J)/SUM	FFOC 466
121	WRITE(9,39)	FFOC 467
39	FORMAT(//7X,19HNORMALIZED GRADIENT//)	FFOC 468
38	CALL IP(MU,MM,NU,NI,NDJDK)	FFOC 469
5010	NC=0	FFOC 470
	IF(LAST.EQ.1) GO TO 5	FFOC 471
C		FFOC 472
C	COUNT GRADIENT DIRECTIONS	FFOC 473
	NGRB=	FFOC 474
	NGD=NGD+1	FFOC 475
	NCHK=	FFOC 476
	IF(NOPR.GI.0) GO TO 102	FFOC 477
	IF(NGD.EQ.3) NCHK=1	FFOC 478
	IF(NGD.EQ.3) NGD=0	FFOC 479
	GO TO 110	FFOC 480
102	IF(NGD.EQ.2) NCHK=1	FFOC 481
	IF(NGD.EQ.2) NGD=0	FFOC 482
110	AJOG=AJT	FFOC 483
	AJOGI=AJL	FFOC 484
5011	DO 42 I=1,NU	FFOC 485
	DO 42 J=1,NM	FFOC 486
	AKG(I,J)=AK(I,J)	FFOC 487
42	DK(I,J)=-EPS *DJDK(I,J)	FFOC 488
	NG=0	FFOC 489
	NC=1	FFOC 490
	NCO=0	FFOC 491
	GO TO 44	FFOC 492
C		FFOC 493
C	STEP SIZE LOGIC	FFOC 494
C		FFOC 495
43	IF(AJT.LT.0.) GO TO 301	FFOC 496
	IF(AJT.LT.AJL) GO TO 41	FFOC 497
C		FFOC 498
C	UNSTABLE -- HALVE STEP SIZE	FFOC 499
301	NCO=NC	FFOC 500
	NC=1	FFOC 501
	NIT=NIT-1	FFOC 502
	IF(NCO.GT.1) NIT=NIT-1	FFOC 503
	AJT=AJOG	FFOC 504
	AJL=AJOGL	FFOC 505
	EPS=EPS/?.	FFOC 506
C ***	MODIFICATIONS	FFOC 507
	IF(EPS.LE.1.0E-06)WRITE(9,7720)EPS	FFOC 508
7720	FORMAT(1H1,//.1X,50H*** EXIT ON DETECTING VERY SMALL VALUE FOR EPS	FFOC 509
	1 ***.G10.4)	FFOC 510
	IF(EPS.LE.1.0E-06)GO TO 173:	FFOC 511
C ***	MODIFICATIONS	FFOC 512
5012	DO 123 I=1,NU	FFOC 513
	DO 123 J=1,NM	FFOC 514
	AK(I,J)=AKG(I,J)	FFOC 515
123	DK(I,J)=.5*DK(I,J)	FFOC 516
	GO TO 44	FFOC 517
41	IF(NC.GT.1) GO TO 45	FFOC 518
	IF(AJT.GT.AJL) GO TO 47	FFOC 519
	IF(NC).EQ.1) GO TO 47	FFOC 520
C		FFOC 521
C	DOUBLE STEP SIZE	FFOC 522
5013	DO 46 I=1,NU	FFOC 523
	DO 46 J=1,NM	FFOC 524
46	AK(I,J)=AK(I,J)+DK(I,J)	FFOC 525
	NIT=NIT+1	FFOC 526

Figure 85. Program FFOC Program Listing (Continued)

	NC=2	FFOC 527
	AJLL=AJL	FFOC 528
	GO TO 7	FFOC 529
C		FFOC 530
C	HALVE STEP SIZE	FFOC 531
47	DO 48 I=1,NU	FFOC 532
	DO 48 J=1,NM	FFOC 533
48	AK(I, J)=AK(I, J)-.5*DK(I, J)	FFOC 534
5014	NIT=NIT+1	FFOC 535
	NC=3	FFOC 536
	AJLL=AJL	FFOC 537
	GO TO 7	FFOC 538
C		FFOC 539
C	COMPUTE NEW STEP SIZE	FFOC 540
45	IF(NC.EQ.3) GO TO 49	FFOC 541
	AJD=AJL-AJT	FFOC 542
	IF(AJD.LT.0.) NC=5	FFOC 543
	AJDD=AJLL-AJL	FFOC 544
	IF(AJD.LT.AJDD) GO TO 431	FFOC 545
	EPSS=.5*EPS	FFOC 546
	GO TO 432	FFOC 547
431	R=(AJT-AJL*2.+AJLL)/(2.*EPS*EPS)	FFOC 548
	GO TO 50	FFOC 549
49	AJD=AJT-AJL	FFOC 550
	AJDD=AJLL-AJT	FFOC 551
	IF(AJDD.LT.0.) NC=4	FFOC 552
	IF(AJD.LT.0.) GO TO 433	FFOC 553
	IF(AJD.LT.AJDD) GO TO 433	FFOC 554
	EPSS=EPS	FFOC 555
	GO TO 432	FFOC 556
433	R=(-4.*AJT-2.*AJL-2.*AJLL)/(EPS*EPS)	FFOC 557
50	AB=(AJL-AJLL-EPS*EPS*R)/EPS	FFOC 558
	EPSS=-AB/(2.*R)	FFOC 559
432	EP=EPSS	FFOC 560
	AJM=AMINI(AJT, AJL, AJLL)	FFOC 561
	EPSO=EPS	FFOC 562
	NCO=NC	FFOC 563
	IF(NC.EQ.3) GO TO 51	FFOC 564
	IF(NC.EQ.4) GO TO 51	FFOC 565
5015	DO 52 I=1,NU	FFOC 566
	DO 52 J=1,NM	FFOC 567
52	AK(I, J)=AK(I, J)-2.*DK(I, J)	FFOC 568
	GO TO 53	FFOC 569
51	DO 54 I=1,NU	FFOC 570
	DO 54 J=1,NM	FFOC 571
54	AK(I, J)=AK(I, J)-.5*DK(I, J)	FFOC 572
53	NC=1	FFOC 573
	NG=1	FFOC 574
	DO 55 I=1,NU	FFOC 575
	DO 55 J=1,NM	FFOC 576
	DK(I, J)=EPSS*DK(I, J)/FPS	FFOC 577
55	AK(I, J)=AK(I, J)+DK(I, J)	FFOC 578
5016	NIT=NIT+1	FFOC 579
	IF(EP.GT.0.) GO TO 175	FFOC 580
	WRITE(9,59) EP	FFOC 581
	EPS=-EPSS	FFOC 582
	NCO=4	FFOC 583
	GO TO 7	FFOC 584
175	EPS=EPSS	FFOC 585
	GO TO 7	FFOC 586
44	DO 65 I=1,NU	FFOC 587
	DO 65 J=1,NM	FFOC 588
65	AK(I, J)=AK(I, J)+DK(I, J)	FFOC 589
5017	NIT=NIT+1	FFOC 590
	GO TO 7	FFOC 591
57	NC=1	FFOC 592

Figure 85. Program FFOC Program Listing (Continued)

59	FORMAT(//7X.12HSTEP SIZE = .E15.A)	FFOC 593
	NR=1	FFOC 594
	AJSTEP=AJT	FFOC 595
	NUNST=C	FFOC 596
C		FFOC 597
C	INITIALIZE NEW PREDICTOR	FFOC 598
	DO 311 I=1,NM	FFOC 599
	DO 311 J=1,NM	FFOC 600
	311 DELK(I,J)=AK(I,J)-DELK(I,J)	FFOC 601
C		FFOC 602
C	PRINT PREDICTOR	FFOC 603
5019	WRITE(9,313)	FFOC 604
	313 FORMAT(//7X.14H NEW PREDICTOR//)	FFOC 605
	CALL IP(MU,MM,NU,NM,DELK)	FFOC 606
	IF(ALAM.GT.ALAMD) GO TO 172	FFOC 607
	GO TO 173A	FFOC 608
	86 FORMAT(3G10.4,30X.16H ALAM DELT ALAMD)	FFOC 609
C		FFOC 610
C	STOP PROGRAM -- PRINT PREDICTOR, PUNCH OUTPUT	FFOC 611
5041	WRITE(9,313)	FFOC 612
	CALL IP(MU,MM,NU,NM,DELK)	FFOC 613
C ***	MODIFICATIONS	FFOC 614
1730	CONTINUE	FFOC 615
	WRITE(1,7701)	FFOC 616
7701	FORMAT(9HFFOC DATA)	FFOC 617
	WRITE(1,7702)	FFOC 618
7702	FORMAT(15HALAM,DELT,ALAMD)	FFOC 619
	WRITE(1,7703)ALAM,DELT,ALAMD	FFOC 620
7703	FORMAT(3G10.4)	FFOC 621
	WRITE(1,7704)	FFOC 622
7704	FORMAT(11HGAIN MATRIX)	FFOC 623
	CALL OUTP(MU,MM,NU,NM,AK,1)	FFOC 624
	WRITE(1,7705)	FFOC 625
7705	FORMAT(20(4H))	FFOC 626
	WRITE(1,7706)	FFOC 627
7706	FORMAT(15HPREDICTOR GAINS)	FFOC 628
	CALL OUTP(MU,MM,NU,NM,DELK,1)	FFOC 629
	WRITE(1,7705)	FFOC 630
	ENDIF 1	FFOC 631
5025	CONTINUE	FFOC 632
C ***	MODIFICATIONS	FFOC 633
	END	FFOC 634

Figure 85. Program FFOC Program Listing (Concluded)

	OVERLAY(KONP,3,0)	DATAK 2
	PROGRAM DATAK	DATAK 3
C		DATAK 4
C	PURPOSE - TO SET UP DIMENSIONS AND CALL DATA PREPARATION PROGRAMS	DATAK 5
C	ANALYSIS-A F KONAP/J K MAHESH-THE HONEYWELL INC.	DATAK 6
C	DATE WRITTEN - 1975	DATAK 7
C		DATAK 8
C	SUBPROGRAMS CALLED	DATAK 9
C	DEPRM	DATAK 10
C	ERRM	DATAK 11
C	DDIAK	DATAK 12
C	OFFOC	DATAK 13
C	DLSA	DATAK 14
C	FINK	DATAK 15
C		DATAK 16
	COMMON/INOUT/IR, IW, IPRINT, INSR, LOCATF, NULL, MARK(20),	DATAK 17
	IJO, JS, JSD, JF, JO	DATAK 18
	COMMON /INF/ NXM, NPM, NUM, CODE, MS1, MS2, MS3, MS4	DATAK 19
	DIMENSION S1(0R500)	DATAK 20
C	DIMENSION A(NXM, NXM), X(NXM, NUM), C(NRM, NXM), D(NPM, NUM)	DATAK 21
	DIMENSION S2(17000)	DATAK 22
C	DIMENSION R1(NXM, NUM), R2(NXM, NUM)	DATAK 23
C	DIMENSION C1(NRM, NXM), C2(NRM, NXM)	DATAK 24
C	DIMENSION D11(NRM, NUM), RK(NUM, NRM), HKC3(NUM, NXM)	DATAK 25
C		DATAK 26
C	DIMENSION CC(NXPM, NXRUM), NAME(NXRUM)	DATAK 27
	DIMENSION S3(00001)	DATAK 28
	DIMENSION S4(00001)	DATAK 29
	DATA HODIA, HOFFO, HDLSA/4HSDTA, 4HSFFO, 4HSLSA/	DATAK 30
	DATA HPRIN/4HPPIN/	DATAK 31
C		DATAK 32
C	COMPUTE ARRAY START INDEXES	DATAK 33
C		DATAK 34
	NXRM=NXM+NRM	DATAK 35
	NXRUM=NXRM+NUM	DATAK 36
	M1=1 * M2=M1+NXM*NXM * M3=M2+NXM*NUM * M4=M3+NPM*NXM	DATAK 37
	M5=M4+NRM*NUM	DATAK 38
	N1=1 * N2=N1+NXM*NUM * N3=N2+NXM*NUM * N4=N3+NPM*NXM	DATAK 39
	N5=N4+NRM*NXM * N6=N5+NPM*NUM * N7=N6+NUM*NRM * N8=N7+NUM*NXM	DATAK 40
	K1=1 * K2=K1+NXRUM*NXRUM	DATAK 41
	K3=K2+NXRUM	DATAK 42
C		DATAK 43
C	CHECK IF SCRATCH ARRAY SIZES ARE SUFFICIENT	DATAK 44
C		DATAK 45
	IF(K3.GT.NR)N8=K3	DATAK 46
	IF((M5.GT.MS1).OR.(N8.GT.MS2))	DATAK 47
	1CALL DEPRM(M5, NR, MS3, MS4, MS1, MS2, MS3, MS4, 3, 0, 4HDATA, 4HK , IW)	DATAK 48
C		DATAK 49
C	CALL DATA PREPARATION PROGRAMS	DATAK 50
C		DATAK 51
	IF(CODE.EQ.HODIA)GO TO 160	DATAK 52
	IF(CODE.EQ.HOFFO)GO TO 260	DATAK 53
	IF(CODE.EQ.HDLSA)GO TO 360	DATAK 54
	CALL DEPRM(1, 4HDATA, 4HK , 3, 0, IW)	DATAK 55
160	CONTINUE	DATAK 56
	CALL DDIAK(S1(M1), S1(M2), S1(M3), S1(M4), S2(N1), S2(N2), S2(N3)	DATAK 57
	1, S2(N4), S2(N5), S2(N6), NXM, NPM, NUM)	DATAK 58
	GO TO 460	DATAK 59
260	CONTINUE	DATAK 60
	CALL OFFOC(S1(M1), S1(M2), S1(M3), S1(M4), S2(N1), S2(N2), S2(N3)	DATAK 61
	1, S2(N4), S2(N5), S2(N6), NXM, NPM, NUM)	DATAK 62
	GO TO 460	DATAK 63
360	CONTINUE	DATAK 64

Figure 86. Program DATAK Program Listing

CALL LSA(S1(M1),S1(M2),S1(M3),S1(M4),S2(N1),S2(N2),S2(N3)	DATAK 65
1,S2(N4),S2(N5),S2(N6),S2(N7),MX,NP,NU,NXM,NRM,NUM)	DATAK 66
CALL F[NK(S1(M1),S1(M2),S1(M3),S1(M4),S2(K1),S2(K2),	DATAK 67
INX,NP,NU,NXM,NRM,NUM,NXR,NXRM)	DATAK 68
460 CONTINUE	DATAK 69
C	DATAK 70
C RETURN TO MAIN OVERLAY	DATAK 71
C	DATAK 72
END	DATAK 73

Figure 86. Program DATAK Program Listing (Concluded)

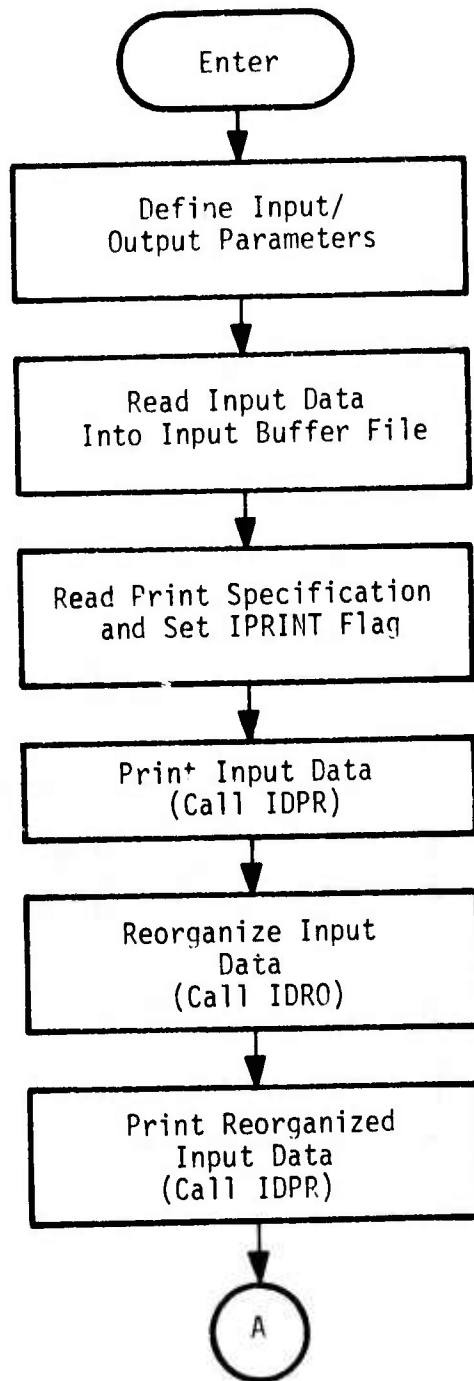


Figure 87. Subroutine KOR2 Flow Chart

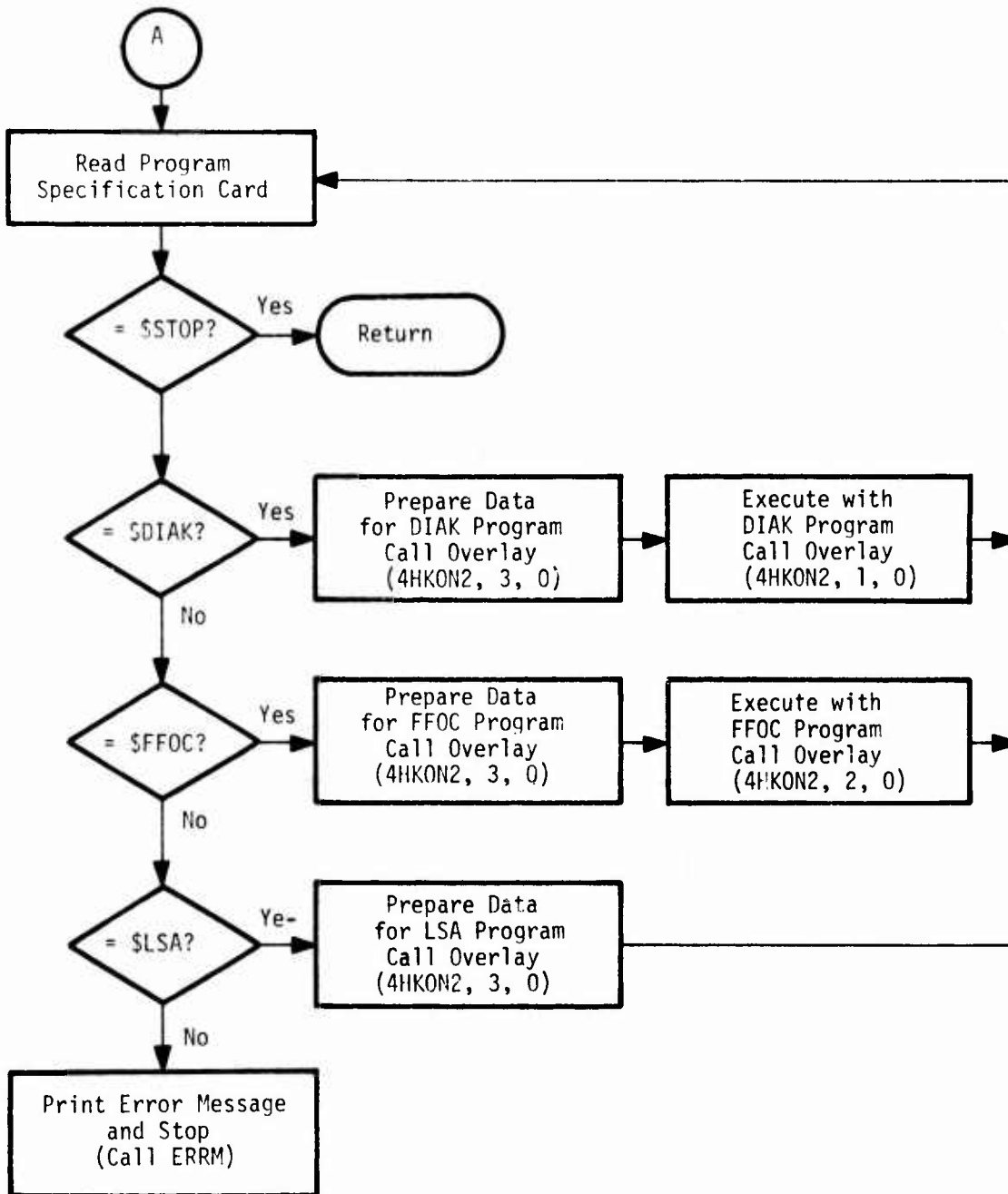


Figure 87. Subroutine KOR2 Flow Chart (Concluded)

```

C          SUBROUTINE KOR62
C
C          ANALYSIS - A F KONAR / J K KAMESH - THE HONEYWELL INC
C          PURPOSE - TO ORGANIZE EXECUTION OF KONPACT-2 PROGRAMS
C          DATE WRITTEN - JULY 1975
C
C          SUBPROGRAMS CALLED
C          IDPO
C          IDPB
C          ERJN
C
C          LABELLED COMMON LIST
C          IR          FILE NO FOR INPUT DATA BUFFER
C          IW          FILE NO FOR LINE PRINTED
C          IPRINT     PRINT CONTROL FLAG
C          INSERT     HOLLERITH INSE
C          LOCATE     HOLLERITH LOCA
C          NULL       HOLLERITH NULL
C          MARK       HOLLERITH MARK
C          JQ         FILE NO FOR QUADRUPLER DATA FILE
C          JS         FILE NO FOR SCRATCH FILE
C          JSD        FILE NO FOR SDSTP FILE
C          JF         FILE NO FOR FDATA FILE
C          JD         FILE NO FOR ODATA FILE
C
C          DIMENSION CARD(20)
C          COMMON /INOUT/IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),
C          1JQ,JS,JSD,JF,JD
C          COMMON /INF/ NXM,NPM,NUN,CONE
C          INTEGER HINSE,HLOCA,HNULL,HDLR
C          DATA HINSE,HLOCA,HNULL,HDLR/4HINSE,4HLOCA,4HNULL,4HDLR/
C          DATA HPRINT,HTHIN,HERYT/4HPRINT,4HTHIN,4HERYT/
C          DATA HPUT,HNAL,HPUT/4HPUT,4HNAL,4HPUT /
C          DATA HDIA,HDFFO,HOLSA/4HSDIA,4HDFFO,4HOLSA/
C          DATA HSTG/4HSTG/
C          DATA /C/HC/
C
C          DEFINE INPUT/OUTPUT PARAMETERS
C
C          IP=7 & IW=9 & IPRINT=4 & JQ=8 & JS=5 & JSD=2 & JF=1 & JD=6
C          INSERT=HINSE & LOCATE=HLOCA & NULL=HNULL
C          DO 10 I=1,20
C          MARK(I)=HDLR
C          100 CONTINUE
C
C          READ INPUT DATA INTO INPUT DATA BUFFER FILE
C
C          110 CONTINUE
C          READ(4,12)CARD
C          IF(EOF(4))140,115
C          115 CONTINUE
C          WRITE(IR,120)CARD
C          120 FORMAT(20A4)
C          130 FORMAT(A1,A3)
C          GO TO 110
C          140 CONTINUE
C          ENDFILE IR
C          REWIND IR
C
C          READ PRINT SPECIFICATION AND SET IPRINT
C
C          142 CONTINUE
C          READ(IR,121) CARD
C          KOR62 2
C          KOR62 3
C          KOR62 4
C          KOR62 5
C          KOR62 6
C          KOR62 7
C          KOR62 8
C          KOR62 9
C          KOR62 10
C          KOR62 11
C          KOR62 12
C          KOR62 13
C          KOR62 14
C          KOR62 15
C          KOR62 16
C          KOR62 17
C          KOR62 18
C          KOR62 19
C          KOR62 20
C          KOR62 21
C          KOR62 22
C          KOR62 23
C          KOR62 24
C          KOR62 25
C          KOR62 26
C          KOR62 27
C          KOR62 28
C          KOR62 29
C          KOR62 30
C          KOR62 31
C          KOR62 32
C          KOR62 33
C          KOR62 34
C          KOR62 35
C          KOR62 36
C          KOR62 37
C          KOR62 38
C          KOR62 39
C          KOR62 40
C          KOR62 41
C          KOR62 42
C          KOR62 43
C          KOR62 44
C          KOR62 45
C          KOR62 46
C          KOR62 47
C          KOR62 48
C          KOR62 49
C          KOR62 50
C          KOR62 51
C          KOR62 52
C          KOR62 53
C          KOR62 54
C          KOR62 55
C          KOR62 56
C          KOR62 57
C          KOR62 58
C          KOR62 59
C          KOR62 60
C          KOR62 61
C          KOR62 62
C          KOR62 63
C          KOR62 64

```

Figure 88. Subroutine KOR62 Program Listing

DECODE(4,13,CARD(1))CC,DUMMY	KORG2 65
IF(CC.EQ.HC)GO TO 142	KORG2 66
IF(CARD(1).NE.HPRINT)GO TO 152	KORG2 67
IF(CARD(3).EQ.HTHIN)IPRINT=	KORG2 68
IF(CARD(3).EQ.HTHIN)GO TO 142	KORG2 69
IF(CARD(3).EQ.HERYT)IPRINT=6	KORG2 70
IF(CARD(3).EQ.HERYT)GO TO 142	KORG2 71
IF(CARD(3).NE.HPRINT)GO TO 144	KORG2 72
IF(IPRINT.EQ.1)IPRINT=5	KORG2 73
IF(IPRINT.EQ.5)GO TO 142	KORG2 74
IPRINT=3	KORG2 75
GO TO 142	KORG2 76
144 CONTINUE	KORG2 77
IF(CARD(3).NE.HNAL)GO TO 146	KORG2 78
IF(IPRINT.EQ.1)IPRINT=4	KORG2 79
IF(IPRINT.EQ.4)GO TO 142	KORG2 80
IPRINT=2	KORG2 81
GO TO 142	KORG2 82
146 CONTINUE	KORG2 83
IF(CARD(3).NE.HPUT)GO TO 148	KORG2 84
IF(IPRINT.EQ.2)IPRINT=4	KORG2 85
IF(IPRINT.EQ.4)GO TO 142	KORG2 86
IF(IPRINT.EQ.3)IPRINT=5	KORG2 87
IF(IPRINT.EQ.5)GO TO 142	KORG2 88
IPRINT=1	KORG2 89
GO TO 142	KORG2 90
148 CONTINUE	KORG2 91
C	KORG2 92
C PRINT ERROR MESSAGE	KORG2 93
C	KORG2 94
WRITE(IW,150)	KORG2 95
150 FORMAT(1H,///.1X,3 HPRINT CARD SPECIFICATION ERROR,///.1X,	KORG2 96
143HINPUT AND FINAL OUTPUT DATA WILL BE PRINTED)	KORG2 97
152 CONTINUE	KORG2 98
REWIND IR	KORG2 99
C	KORG2100
C PRINT INPUT DATA	KORG2101
C	KORG2102
IF((IPRINT.NE.1).AND.(IPRINT.LT.4))GO TO 154	KORG2103
WRITE(IW,154)	KORG2104
154 FORMAT(1H,///.1X,24H*** INPUT DATA CARDS ***,//)	KORG2105
CALL IDPR(IR,IW)	KORG2106
WRITE(IW,156)	KORG2107
156 FORMAT(///.1X,31H*** END OF INPUT DATA CARDS ***,//)	KORG2108
REWIND IR	KORG2109
158 CONTINUE	KORG2110
C	KORG2111
C REORGANIZE INPUT DATA	KORG2112
C	KORG2113
CALL IDRO(IR,IW,JS)	KORG2114
C	KORG2115
C PRINT REORGANIZED INPUT DATA	KORG2116
C	KORG2117
IF(IPRINT.LT.6)GO TO 164	KORG2118
WRITE(IW,160)	KORG2119
160 FORMAT(1H,///.1X,37H*** REORGANIZED INPUT DATA ***,//)	KORG2120
CALL IDPR(IR,IW)	KORG2121
WRITE(IW,162)	KORG2122
162 FORMAT(///.1X,37H*** END OF REORGANIZED INPUT DATA ***,//)	KORG2123
164 CONTINUE	KORG2124
C	KORG2125
C READ INPUT DATA CARDS	KORG2126
C	KORG2127
159 CONTINUE	KORG2128
READ(IR,120) CARD	KORG2129
IF(CARD(1).EQ.HPRINT)GO TO 154	KORG2130

Figure 88. Subroutine KORG2 Program Listing (Continued)

	IF (CARD(1).EQ.HDST0) RETURN	KORG2131
	IF (CARD(1).EQ.HDD1A) GO TO 180	KORG2132
	IF (CARD(1).EQ.HOFF0) GO TO 200	KORG2133
	IF (CARD(1).EQ.HHLSA) GO TO 360	KORG2134
	CALL FPRM(1.4HKORG.4M2 .0.).1W)	KORG2135
C		KORG2136
C	CALL OVERLAY LOADER TO LOAD REQUIRED PROGRAMS FOR EXECUTION	KORG2137
C		KORG2138
180	CONTINUE	KORG2139
	CODE=CARD(1)	KORG2140
	CALL OVERLAY(4HKONP.3.0)	KORG2141
	CALL OVERLAY(4HKONP.1.0)	KORG2142
	GO TO 159	KORG2143
260	CONTINUE	KORG2144
	CODE=CARD(1)	KORG2145
	CALL OVERLAY(4HKONP.3.0)	KORG2146
	CALL OVERLAY(4HKONP.2.0)	KORG2147
	GO TO 159	KORG2148
360	CONTINUE	KORG2149
	CODE=CARD(1)	KORG2150
	CALL OVERLAY(4HKONP.3.0)	KORG2151
	GO TO 159	KORG2152
	END	KORG2153

Figure 88. Subroutine KORG2 Program Listing (Concluded)

```

SUBROUTINE TIMER(A,G2,HDK,X,XT,DX,DX1,XL,GN,GS,R,IPLR,ITITL, TIMER 2
IUUNIT,CL,T,DT,ST,YMAX,YMIN,IFLT,IRUN,IDATE,NSTEP,NRAND,NPLOT,TIMER 3
2NPRIN,NN,NX,NF,NG,NC,NR,MXR,MN,MA,MARSP,NOP,NAME1,NAME2,SCAL,NEWY,TIMER 4
3T1,T2,NGLG)
DIMENSION A(MX,MX),G2(MX,MN),HDK(MX,MA),X(MX),DX(MX),DX1(MX) TIMER 5
DIMENSION XT(MX,MN),XL(MX,2),GN(MX,MN),GS(MA,MN),R(MARSP) TIMER 6
DIMENSION IPLR(MXR),ITITL(MXN),IUUNIT(MXR),IRUF(15),LW(10) TIMER 7
DIMENSION YMAX(MAR),YMIN(MAR),CL(MN,1) TIMER 8
DIMENSION SCAL(MAR) TIMER 9
DIMENSION NEWY(4R) TIMER 10
INTEGER BLANK TIMER 11
RUN=IRUN TIMER 12
IF(NOP.GT.1) GO TO 22 TIMER 13
WRITE(9,2)3) TIMER 14
203 FORMAT(1H1/7X,44HNO. OF PLOTS IS ZERO - IGNORE TIME RESPONSES) TIMER 15
RETURN TIMER 16
22 WRITE(9,102) TIMER 17
102 FORMAT(1H1/7X,14HTIME RESPONSES//) TIMER 18
BLANK=1:H TIMER 19
LW(1)=1:H* TIMER 20
LW(2)=1:H* TIMER 21
LW(3)=1:H* TIMER 22
LW(4)=1:H* TIMER 23
LW(5)=1:H* TIMER 24
LW(6)=1:H* TIMER 25
LW(7)=1:H* TIMER 26
LW(8)=1:H* TIMER 27
LW(9)=1:H* TIMER 28
LW(10)=1:H* TIMER 29
C NSTEP=0 NO STEP INPUTS TIMER 30
C =1 STEP COMMANDS TIMER 31
C =2 STEP GUSTS TIMER 32
C =3 BOTH TIMER 33
C =4 NO STEP INPUTS - TRANSIENTS ONLY TIMER 34
C NRAND=0 NO RANDOM INPUTS TIMER 35
C NRAND=1 GUSTS TIMER 36
C NPRIN=0 DON'T PRINT RESPONSES TIMER 37
C NPRIN=1 PRINT RESPONSES TIMER 38
C NPLOT=0 NO PLOTS TIMER 39
C NPLOT=1 CALCOMP PLOTS TIMER 40
C NPLOT=2 LINE PRINTER PLOTS TIMER 41
C NPLOT=3 BOTH TIMER 42
IRUF(1)=IDATE TIMER 43
IRUF(2)=BLANK TIMER 44
IRUF(3)=NAME1 TIMER 45
IRUF(4)=NAME2 TIMER 46
IRUF(5)=1:H FLIGHT TIMER 47
IRUF(6)=1:H CONDITION TIMER 48
IF(NPLOT.EQ.0.OR.NPLOT.EQ.2) GO TO 2 TIMER 49
CALL FACTOR(2,1) TIMER 50
CALL PLOT(0,0,-13,0,-3) TIMER 51
CALL PLOT(0,0,5,-3) TIMER 52
CALL SYMBOL(0,0,0,14,IRUF(1),90,0,40) TIMER 53
CALL SYMBOL(0,5,0,0,14,IRUF(5),90,0,20) TIMER 54
CALL SYMBOL(0,5,3,0,14,IFLT,90,0,10) TIMER 55
CALL SYMBOL(1,0,0,0,14,IRUN,9,0,3) TIMER 56
CALL NUMBER(1,0,0,0,14,RUN,9,0,-1) TIMER 57
CALL PLOT(1,5,0,0,-3) TIMER 58
IF(NPLOT.EQ.1) GO TO 1 TIMER 59
2 WRITE(9,100) (IRUF(I),I=1,6),IFLT,IRUN TIMER 60
100 FORMAT(2X,6A10,2X,A10,2X,3HRUN,13//) TIMER 61
1 CONTINUE TIMER 62
DO 3 I=1,NX TIMER 63
TIMER 64

```

Figure 89. Subroutine TIMER Program Listing

	DO 3 J=1,NN	TIMER 65
	GN(I, J)=0.	TIMER 66
3	GS(I, J)=0.	TIMER 67
	NFG=NX	TIMER 68
	IF(INSTEP.EQ.4) GO TO 7	TIMER 69
	IF(NRAND.EQ.0) GO TO 4	TIMER 70
	N=NF+1	TIMER 71
	NFG=NX-NC	TIMER 72
	ETA=G*AN(1)	TIMER 73
	DO 5 I=N,NFG	TIMER 74
	DO 5 J=1,NN	TIMER 75
5	GN(I, J)=G2(I, J)/SQRT(OT)	TIMER 76
	GO TO 1A	TIMER 77
4	NFG=NF	TIMER 78
10	IF(INSTEP.EQ.2) GO TO 4	TIMER 79
	IF(INSTEP.EQ.0) GO TO 7	TIMER 80
	DO 8 I=1,NF	TIMER 81
	DO 8 J=1,NC	TIMER 82
	JG=J+JG	TIMER 83
	JJ=J+JX-NC	TIMER 84
8	GS(I, JG)=A(I, JJ)*CL(JG, 1)	TIMER 85
	IF(INSTEP.NE.3) GO TO 7	TIMER 86
6	NFG=NF+NGLG	TIMER 87
7	CONTINUE	TIMER 88
	WRITE(9,101) NOP	TIMER 89
101	FORMAT(7X, 9HTHERE ARE .13,21H RESPONSES TO COMPUTE//)	TIMER 90
	NT=T/DT	TIMER 91
	S=NT	TIMER 92
	S=S*DT	TIMER 93
	IF(S.LT.T) NT=NT+1	TIMER 94
	NT=NT+1	TIMER 95
	NTP=ST/DT	TIMER 96
	S=NTP	TIMER 97
	S=S*DT	TIMER 98
	IF(S.LT.ST) NTP=NTP+1	TIMER 99
	IF(NTP.EQ.0) NTP=1	TIMER100
	NTP=NTP/NTP	TIMER101
	NTP=NTP+1	TIMER102
	NPTOT=0	TIMER103
	DO 12 J=1,NN	TIMER104
	IF(INSTEP.EQ.4) GO TO 51	TIMER105
	IF(J.GT.NG) GO TO 41	TIMER106
	IF(NRAND.EQ.0.AND.NSTEP.EQ.1) GO TO 12	TIMER107
	GO TO 51	TIMER108
41	IF(INSTEP.EQ.0.OR.NSTEP.EQ.2) GO TO 12	TIMER109
51	DO 11 I=1,NX	TIMER110
	X(I)=X1(I, J)	TIMER111
11	DX1(I)=0.	TIMER112
	IF(J.LE.NG) GO TO 56	TIMER113
	IF(INSTEP.EQ.0) GO TO 56	TIMER114
	IF(INSTEP.EQ.2) GO TO 56	TIMER115
	IF(INSTEP.EQ.4) GO TO 56	TIMER116
	JJ=J+JX-NC-NG	TIMER117
	X(JJ)=CL(J, 1)	TIMER118
56	CONTINUE	TIMER119
	IF(NPRIN.EQ.0) GO TO 24	TIMER120
	WRITE(9,103) J	TIMER121
103	FORMAT(1H1/7X, 20HTIME RESPONSES FOR DISTURBANCE .13//)	TIMER122
24	DO 17 IT=1,NT	TIMER123
	IF(J.GT.NG) GO TO 13	TIMER124
	IF(INSTEP.LE.1) GO TO 55	TIMER125
	IF(INSTEP.GT.3) GO TO 55	TIMER126
	CALL GUST(A, GS, CL, X, DT, T1, T2, J, NFG, NG, IT, NX, M1)	TIMER127
55	CONTINUE	TIMER128
	IF(NRAND.EQ.0) GO TO 13	TIMER129
	ETA=G*AN(1)	TIMER130

Figure 89. Subroutine TIMER Program Listing (Continued)

GO TO 14	TIMER131
13 ETA=0.	TIMER132
14 DO 15 I=1,NFG	TIMER133
DX(I)=GN(I,J)*ETA+GS(I,J)	TIMER134
DO 52 K=1,NFG	TIMER135
52 DX(I)=DX(I)+A(I,K)*X(K)	TIMER136
IF(XL(I,2).LE.0.) GO TO 15	TIMER137
IF(ABS(DX(I)).GT.XL(I,2))DX(I)=SIGN(XL(I,2),DX(I))	TIMER138
15 CONTINUE	TIMER139
DO 16 I=1,NFG	TIMER140
X(I)=X(I)+DT*(2.*DX(I)-DX(I))/2.	TIMER141
DX(I)=DX(I)	TIMER142
IF(XL(I,1).LE.0.) GO TO 16	TIMER143
IF(ABS(X(I)).GT.XL(I,1)) X(I)=SIGN(XL(I,1),X(I))	TIMER144
16 CONTINUE	TIMER145
IF(NTP.EQ.1) GO TO 18	TIMER146
ITM=MOD(IT,NTP)	TIMER147
IF(ITM.NE.1) GO TO 17	TIMER148
ITT=(IT/NTP)+1	TIMER149
18 IF(NTP.EQ.1) ITT=IT	TIMER150
II=ITT-NTP-2	TIMER151
IP=1	TIMER152
DO 19 I=1,NR	TIMER153
IF(IP.GT.NOP) GO TO 23	TIMER154
IF(IPR(IP).NE.1) GO TO 19	TIMER155
II=II+NTP+2	TIMER156
R(II)=.	TIMER157
DO 20 K=1,NX	TIMER158
20 R(II)=R(II)+HDK(I,K)*X(K)	TIMER159
R(II)=R(II)*SCAL(IP)	TIMER160
IP=IP+1	TIMER161
19 CONTINUE	TIMER162
DO 21 I=1,NFG	TIMER163
IX=I+NR	TIMER164
IF(IP.GT.NOP) GO TO 23	TIMER165
IF(IPR(IP).NE.IX) GO TO 21	TIMER166
II=II+NTP+2	TIMER167
R(II)=X(II)*SCAL(IP)	TIMER168
IP=IP+1	TIMER169
21 CONTINUE	TIMER170
23 II=II+NTP+2	TIMER171
TIME=ITT-1	TIMER172
TIME=TIME*ST	TIMER173
R(II)=TIME	TIMER174
IF(INPIN.EQ.0) GO TO 17	TIMER175
WRITE(9,202) TIME	TIMER176
202 FORMAT(5X,6HTIME =,F10.3)	TIMER177
200 FORMAT(5X,4(A)C.2H= ,G8.2,2X))	TIMER178
IPR=ITT	TIMER179
K=-3	TIMER180
DO 26 INP=1,NOP+4	TIMER181
K=K+4	TIMER182
IPR1=IPR+NTP+2	TIMER183
IPR2=IPR1+NTP+2	TIMER184
IPR3=IPR2+NTP+2	TIMER185
KK=K+2-NOP	TIMER186
IF(KK.LE.0) GO TO 54	TIMER187
IF(KK.EQ.3) WRITE(9,200) ITITL(K),R(IPR)	TIMER188
IF(KK.EQ.2) WRITE(9,200) ITITL(K),R(IPR),ITITL(K+1),R(IPR1)	TIMER189
IF(KK.EQ.1) WRITE(9,200) ITITL(K),R(IPR),ITITL(K+1),R(IPR1)	TIMER190
1,ITITL(K+2),R(IPR2)	TIMER191
GO TO 26	TIMER192
54 WRITE(9,200) ITITL(K), R(IPR),ITITL(K+1),R(IPR),ITITL(K+2),	TIMER193
IR(IPR2),ITITL(K+3),R(IPR3)	TIMER194
26 IPR=IPR3+NTP+2	TIMER195
WRITE(9,201)	TIMER196

Figure 89. Subroutine TIMER Program Listing (Continued)

201	FORMAT(1H)	TIMER197
17	CONTINUE	TIMER198
	IF(NPLOT.EQ.0) GO TO 12	TIMER199
	IF(NPLOT.EQ.1) GO TO 30	TIMER200
	WRITE(9,103) J	TIMER201
	DO 32 K=1,NOP	TIMER202
	DO 31 I=1,15	TIMER203
31	IRUF(I)=BLANK	TIMER204
	IF(INE.Y(K).EQ.0) GO TO 33	TIMER205
	YMAX(K)=-1.E+20	TIMER206
	YMIN(K)= 1.E+20	TIMER207
	DO 34 L=1,NTP	TIMER208
	I=(K-1)*(NTP+2)+L	TIMER209
	YMAX(K)=AMAX(YMAX(K),R(I))	TIMER210
34	YMIN(K)=AMIN(YMIN(K),R(I))	TIMER211
33	CONTINUE	TIMER212
	IF(YMIN(K).EQ.YMAX(K)) GO TO 32	TIMER213
	RANGE=YMAX(K)-YMIN(K)	TIMER214
	IRUF(3)=ITITL(K)	TIMER215
	IRUF(4)=IUNIT(K)	TIMER216
	WRITE(9,104) (IRUF(I),I=1,5)	TIMER217
104	FORMAT(1H1/5A10//)	TIMER218
	IRUF(3)=BLANK	TIMER219
	IRUF(4)=BLANK	TIMER220
	X(I)=YMIN(K)	TIMER221
	DO 36 I=1,5	TIMER222
36	X(I+1)=X(I)+RANGE/5.	TIMER223
	WRITE(9,105) (X(I),I=1,6)	TIMER224
105	FORMAT(6F14,3)	TIMER225
	WRITE(9,106)	TIMER226
106	FORMAT(7X,2H 1.5(14H-----[])	TIMER227
	DO 37 L=1,NTP	TIMER228
	I=(K-1)*(NTP+2)+L	TIMER229
	IF(RANGE.EQ.0.) GO TO 42	TIMER230
	LL=(R(I)-YMIN(K))*70./RANGE	TIMER231
	IF(LL.EQ.0) GO TO 42	TIMER232
	INL=L/10	TIMER233
	IF(INL.LE.0) INL=0	TIMER234
	IF(INL.GT.11) INL=11	TIMER235
	IW=MOD(LL,10)	TIMER236
	IF(IW.LE.0) GO TO 40	TIMER237
	IF(INL.EQ.0) GO TO 53	TIMER238
	DO 38 I=1,INL	TIMER239
38	IRUF(I)=BLANK	TIMER240
53	IRUF(INL+1)=LW(IW)	TIMER241
	IB=INL+2	TIMER242
	DO 39 I=IB,15	TIMER243
39	IRUF(I)=BLANK	TIMER244
	GO TO 45	TIMER245
40	IW=10	TIMER246
	IW=10	TIMER247
	IE=INL-1	TIMER248
	DO 43 I=1,IE	TIMER249
43	IRUF(I)=BLANK	TIMER250
	IRUF(IE+1)=LW(IW)	TIMER251
	IB=IE+2	TIMER252
	DO 44 I=IB,15	TIMER253
44	IRUF(I)=BLANK	TIMER254
	GO TO 45	TIMER255
42	IRUF(I)=LW(I)	TIMER256
	DO 46 I=2,15	TIMER257
46	IRUF(I)=BLANK	TIMER258
45	TIME=L-1	TIMER259
	TIME=TIME*ST	TIMER260
	WRITE(9,107) TIME,(IRUF(I),I=1,12)	TIMER261
107	FORMAT(F6.2,7MS 1.12A10)	TIMER262

Figure 89. Subroutine TIMER Program Listing (Continued)

DO 47 I=1,15	TIMER263
47 IBUF(1)=BLANK	TIMER264
37 CONTINUE	TIMER265
32 CONTINUE	TIMER266
30 CONTINUE	TIMER267
IF(NPLOT.EQ.2) GO TO 12	TIMER268
FJ=J	TIMER269
CALL SYMBOL(0.,0.,14,30)TIME RESPONSES FOR DISTURBANCE(90.,30)	TIMER270
CALL NUMBER(0.,5.,14,FJ,90.,-1)	TIMER271
CALL PLOT(1.,0.,-3)	TIMER272
IARX=NOP *(NTP+2)+1	TIMER273
DO 48 K=1,NOP	TIMER274
IF(NEWY(K).EQ.0) GO TO 49	TIMER275
IAR=(K-1)*(NTP+2)+1	TIMER276
CALL SCALE(R(IAR),R.,NTP,1)	TIMER277
CALL SCALE(R(IARX),10.,NTP,1)	TIMER278
GO TO 50	TIMER279
49 IAR=K*(NTP+2)-1	TIMER280
IF(YMIN(K).EQ.YMAX(K)) GO TO 48	TIMER281
R(IAR)=YMIN(K)	TIMER282
R(IAR+1)=(YMAX(K)-YMIN(K))/R.	TIMER283
IAR=IARX+NTP	TIMER284
R(IAR)=0.	TIMER285
R(IAR+1)=T/10.	TIMER286
50 IBUF(1)=ITITL(K)	TIMER287
IBUF(2)=BLANK	TIMER288
IBUF(3)=IUNIT(K)	TIMER289
IBUF(4)=10)TIME IN SE	TIMER290
IBUF(5)=10)CONDS	TIMER291
IAR=K*(NTP+2)-1	TIMER292
IARP=IAR+1	TIMER293
CALL AXIS(0.,0.,IBUF(1),30,8.,90.,R(IAR),R(IARP))	TIMER294
IAR=IARX+NTP	TIMER295
IARP=IAR+1	TIMER296
CALL AXIS(0.,0.,IBUF(4),-20,10.,0.,R(IAR),R(IARP))	TIMER297
IAR=(K-1)*(NTP+2)+1	TIMER298
CALL LINE(R(IARX),R(IAR),NTP,1,0,0)	TIMER299
CALL PLOT(13.,0.,-3)	TIMER300
NPTOT=NPTOT+1	TIMER301
IF(NPTOT.LT.5) GO TO 48	TIMER302
NPTOT=0	TIMER303
CALL DSP(2)	TIMER304
CALL PLOT(0.,-13.,-3)	TIMER305
CALL PLOT(0.,0.5,-3)	TIMER306
IBUF(1)=IDATE	TIMER307
IBUF(2)=BLANK	TIMER308
IBUF(3)=NAME1	TIMER309
IBUF(4)=NAME2	TIMER310
IBUF(5)=10H FLIGHT	TIMER311
IBUF(6)=10H CONDITION	TIMER312
CALL SYMBOL(0.,0.,14,IBUF(1),90.,40)	TIMER313
CALL SYMBOL(.5,0.,14,IBUF(5),90.,20)	TIMER314
CALL SYMBOL(.5,3.,14,IFLT,90.,10)	TIMER315
CALL SYMBOL(1.,0.,14,3)RUN,90.,3)	TIMER316
CALL NUMBER(1.,.6.,14, RUN,90.,-1)	TIMER317
CALL PLOT(1.5,0.,-3)	TIMER318
48 CONTINUE	TIMER319
12 CONTINUE	TIMER320
RETURN	TIMER321
END	TIMER322

Figure 89. Subroutine TIMER Program Listing (Concluded)

	SUBROUTINE SGUST(A,GS,CL,X,DT,T1,T2,J,NF,NG,IT,MX,MN)	SGUST 2
	DIMENSION A(MX,MX),GS(MX,MN),CL(MN,1),X(MX)	SGUST 3
	IF(IT.GT.1) GO TO 7	SGUST 4
	JJ=J*IF	SGUST 5
	DO 5 I=1,NF	SGUST 6
5	GS(I,1)=A(I,JJ)*CL(J,1)	SGUST 7
	X(JJ)=CL(J,1)	SGUST 8
	ND1=T1/DT	SGUST 9
	ND2=T2/DT	SGUST 10
	S=ND1*DT	SGUST 11
	IF(S.LT.T1) ND1=ND1+1	SGUST 12
	S=ND2*DT	SGUST 13
	IF(S.LT.T2) ND2=ND2+1	SGUST 14
	ND1=ND1+1	SGUST 15
	ND2=ND2+1	SGUST 16
3	IF(IT.LT.ND1) RETURN	SGUST 17
	IF(IT.GT.ND1) GO TO 2	SGUST 18
	JJ=J+IG*NF	SGUST 19
	DO 1 I=1,NF	SGUST 20
1	GS(I,1)=GS(I,J)+A(I,JJ)*CL(J,1)	SGUST 21
	X(JJ)=CL(J,1)	SGUST 22
2	IF(IT.NE.ND2) RETURN	SGUST 23
	JJ=J+2*NG*NF	SGUST 24
	DO 4 I=1,NF	SGUST 25
4	GS(I,1)=GS(I,J)+A(I,JJ)*CL(J,1)	SGUST 26
	X(JJ)=CL(J,1)	SGUST 27
	RETURN	SGUST 28
	END	SGUST 29

Figure 90. Subroutine SGUST Program Listing

SUBROUTINE CAL1 (A,XN,P,KWA,N,NR,IMAX,IT,IERR,EF)	CAL1	2
DIMENSION A(NR,1),XN(NR,1),P(NR,1),KWA(NR)	CAL1	3
IERR=	CAL1	4
TR=0.	CAL1	5
DO 300 I=1,N	CAL1	6
300 TR=TR+A(I,I)	CAL1	7
FN=N	CAL1	8
TR=AMAX1(TR,-FN)	CAL1	9
IF (TR) 301,2,2	CAL1	10
2 IERR=	CAL1	11
GO TO 601	CAL1	12
301 ALF=ABS(TR)/FN	CAL1	13
NC=N*(N+1)	CAL1	14
NC=NC/2	CAL1	15
DO 60 I=1,N	CAL1	16
DO 63 J=1,N	CAL1	17
GOTO(61,62),IT	CAL1	18
61 P(I,J)=A(I,J)	CAL1	19
GOTO 63	CAL1	20
62 P(I,J)=A(J,I)	CAL1	21
63 CONTINUE	CAL1	22
P(I,I)=P(I,I)-ALF	CAL1	23
60 CONTINUE	CAL1	24
CALL TDINVR (ISOL,IDSOL,N,N,P,NR,KWA,DET)	CAL1	25
IF ((ISOL+IDSOL).LE.2) GO TO 22	CAL1	26
IERR=	CAL1	27
GO TO 601	CAL1	28
22 DO 4 I=1,N	CAL1	29
DO 4 J=1,N	CAL1	30
A(I,J)=0.	CAL1	31
DO 4 K=1,N	CAL1	32
4 A(I,J)=A(I,J)+P(K,I)*XN(K,J)*2.*ALF	CAL1	33
DO 5 I=1,N	CAL1	34
DO 5 J=1,N	CAL1	35
XN(I,I)=0.	CAL1	36
DO 5 K=1,N	CAL1	37
5 XN(I,I)=XN(I,I)+A(I,K)*P(K,J)	CAL1	38
DO 7 I=1,N	CAL1	39
DO 8 J=1,N	CAL1	40
8 P(I,J)=P(I,J)*2.*ALF	CAL1	41
7 P(I,I)=P(I,I)+1.	CAL1	42
ITER=	CAL1	43
100 CONTINUE	CAL1	44
DO 9 I=1,N	CAL1	45
DO 9 J=1,N	CAL1	46
A(I,J)=0.	CAL1	47
DO 9 K=1,N	CAL1	48
9 A(I,J)=A(I,J)+P(K,I)*XN(K,J)	CAL1	49
ICOT=	CAL1	50
DO 10 I=1,N	CAL1	51
DO 10 J=1,N	CAL1	52
DXIJ=	CAL1	53
DO 11 K=1,N	CAL1	54
11 DXIJ=DXIJ+A(I,K)*P(K,J)	CAL1	55
XN(I,I)=XN(I,I)+DXIJ	CAL1	56
XN(J,I)=XN(I,J)	CAL1	57
AXN=AMAX1(XN(I,I))	CAL1	58
IF (AXN.LT.1.E-20) GO TO 14	CAL1	59
IF (AXN.LT.1.E-20) GO TO 201	CAL1	60
IERR=	CAL1	61
GO TO 601	CAL1	62
201 RAT=ABS(DXIJ/XN(I,I))	CAL1	63
IF (RAT-EE) 14,14,70	CAL1	64

Figure 91. Subroutine CAL1 Program Listing

14 ICOT=ICOT+1	CAL1 65
70 CONTINUE	CAL1 66
10 CONTINUE	CAL1 67
18 ITER=ITER+1	CAL1 68
IF (ICOT-NC) 15,50,15	CAL1 69
15 CONTINUE	CAL1 70
DO 20 I=1,N	CAL1 71
DO 20 J=1,N	CAL1 72
20 A(I,J)=P(I,J)	CAL1 73
16 DO 17 I=1,N	CAL1 74
DO 17 J=1,N	CAL1 75
P(I,J)=0.	CAL1 76
DO 17 K=1,N	CAL1 77
17 P(I,J)=P(I,J)+A(I,K)*A(K,J)	CAL1 78
40 IF (ITER-IMAX) 100,50,50	CAL1 79
50 CONTINUE	CAL1 80
WRITE (9,600) ITER	CAL1 81
600 FORMAT (/7X,6H ITER=I2)	CAL1 82
RETURN	CAL1 83
601 WRITE (9,602) IERR	CAL1 84
602 FORMAT (/7X,6H IERR=I2)	CAL1 85
RETURN	CAL1 86
END	CAL1 87

Figure 91. Subroutine CAL1 Program Listing (Concluded)

SUBROUTINE STRIC(A,R,PS,W,S,TF,NX,NU,NXM,MU)	STRIC 2
DIMENSION A(NXM,NXM),R(NXM,MU),W(NXM,NXM),S(NXM,NXM)	STRIC 3
DIMENSION TPF(NXM,NXM),PS(NXM,NXM)	STRIC 4
DT=.01	STRIC 5
DO 1 I=1,NX	STRIC 6
DO 1 J=1,NX	STRIC 7
W(I,J)=0.	STRIC 8
DO 1 K=1,NU	STRIC 9
1 W(I,J)=W(I,J)+R(I,K)*R(J,K)*DT	STRIC 10
NT=10	STRIC 11
KT=10	STRIC 12
T=0.	STRIC 13
DO 20 L=1,KT	STRIC 14
T=T+DT	STRIC 15
DO 3 I=1,NX	STRIC 16
DO 4 J=1,NX	STRIC 17
S(I,J)=0.	STRIC 18
4 TPF(I,J)=0.	STRIC 19
S(I,I)=1.	STRIC 20
3 TPF(I,I)=1.	STRIC 21
DO 10 M=2,NT	STRIC 22
FAC=M-1	STRIC 23
FAC=1./FAC	STRIC 24
DO 5 I=1,NX	STRIC 25
DO 5 J=1,NX	STRIC 26
PS(I,I)=0.	STRIC 27
DO 6 K=1,NX	STRIC 28
6 PS(I,I)=PS(I,I)-TPF(I,K)*A(K,J)*FAC*T	STRIC 29
5 S(I,J)=S(I,J)+PS(I,J)	STRIC 30
DO 7 I=1,NX	STRIC 31
DO 7 J=1,NX	STRIC 32
7 TPF(I,J)=PS(I,J)	STRIC 33
100 CONTINUE	STRIC 34
DO 8 I=1,NX	STRIC 35
DO 8 J=1,NU	STRIC 36
TPF(I,J)=0.	STRIC 37
DO 8 K=1,NX	STRIC 38
8 TPF(I,J)=TPF(I,J)+S(I,K)*R(K,J)	STRIC 39
DO 9 I=1,NX	STRIC 40
DO 9 J=1,NX	STRIC 41
DO 9 K=1,NU	STRIC 42
9 W(I,J)=W(I,J)+TPF(I,K)*TPF(I,K)*DT	STRIC 43
200 CONTINUE	STRIC 44
WRITE(9,300)	STRIC 45
300 FORMAT(1H1/7X,12H W(T) MATRIX/)	STRIC 46
CALL WP(NXM,NXM,NX,NX,W)	STRIC 47
RETURN	STRIC 48
END	STRIC 49

Figure 9%. Subroutine STRIC Program Listing

	SUBROUTINE SHUFL (A,MM,NN,M,N,MC,NC,NORD,B,MX)	SHUFL 2
	DIMENSION A(MM,NN),NORD(MX),B(MX,MX)	SHUFL 3
	IF(MC,EQ,0) GO TO 1	SHUFL 4
	DO 2 I=1,M	SHUFL 5
	II=NORD(I)	SHUFL 6
2	DO 2 I=1,N	SHUFL 7
	B(I,J)=A(II,J)	SHUFL 8
	DO 3 I=1,M	SHUFL 9
	DO 3 I=1,N	SHUFL 10
3	A(I,J)=B(I,J)	SHUFL 11
1	CONTINUE	SHUFL 12
	IF(NC,EQ,0) RETURN	SHUFL 13
	DO 4 I=1,N	SHUFL 14
	JJ=NORD(J)	SHUFL 15
	DO 4 I=1,M	SHUFL 16
4	B(I,J)=A(I,JJ)	SHUFL 17
	DO 5 I=1,M	SHUFL 18
	DO 5 I=1,N	SHUFL 19
5	A(I,J)=B(I,J)	SHUFL 20
	RETURN	SHUFL 21
	END	SHUFL 22

Figure 93. Subroutine SHUFL Program Listing

	SUBROUTINE SHUF(F,G1,G2,H,AM,AKG,Y,NORD,MX,NX,MR,NR,MM,NM,MU,NU,	SHUF	2
	IMNN,NN)	SHUF	3
	DIMENSION F(MX,MX),G1(MX,MU),G2(MX,MNN),H(MR,MX),AM(MM,MX),	SHUF	4
	AKG(MU,MX),NORD(MX)	SHUF	5
	DIMENSION Y(MX,MX)	SHUF	6
	DO 1 I=1,NX	SHUF	7
	II=NORD(I)	SHUF	8
	DO 1 J=1,NX	SHUF	9
	JJ=NORD(J)	SHUF	10
1	Y(I,J)=F(II,JJ)	SHUF	11
	DO 2 I=1,NX	SHUF	12
	DO 2 J=1,NX	SHUF	13
2	F(I,J)=Y(I,J)	SHUF	14
	DO 3 I=1,NX	SHUF	15
	II=NORD(I)	SHUF	16
	DO 4 J=1,NU	SHUF	17
4	Y(I,J)=G1(II,J)	SHUF	18
	DO 3 J=1,NN	SHUF	19
	JJ=J+NU	SHUF	20
3	Y(I,J)=G2(II,J)	SHUF	21
	DO 5 I=1,NX	SHUF	22
	DO 6 J=1,NU	SHUF	23
6	G1(I,J)=Y(I,J)	SHUF	24
	DO 5 J=1,NN	SHUF	25
	JJ=J+NU	SHUF	26
5	G2(I,J)=Y(I,JJ)	SHUF	27
	DO 7 J=1,NX	SHUF	28
	JJ=NORD(J)	SHUF	29
	DO 7 I=1,NR	SHUF	30
7	Y(I,J)=H(I,JJ)	SHUF	31
	DO 8 J=1,NX	SHUF	32
	DO 8 I=1,NR	SHUF	33
8	H(I,J)=Y(I,J)	SHUF	34
	DO 9 J=1,NX	SHUF	35
	JJ=NORD(J)	SHUF	36
	DO 9 I=1,NM	SHUF	37
9	Y(I,J)=AM(I,JJ)	SHUF	38
	DO 10 J=1,NX	SHUF	39
	DO 10 I=1,NM	SHUF	40
10	AM(I,J)=Y(I,J)	SHUF	41
	DO 11 J=1,NX	SHUF	42
	JJ=NORD(J)	SHUF	43
	DO 11 I=1,NU	SHUF	44
11	Y(I,J)=AKG(I,JJ)	SHUF	45
	DO 12 J=1,NX	SHUF	46
	DO 12 I=1,NU	SHUF	47
12	AKG(I,J)=Y(I,J)	SHUF	48
	RETURN	SHUF	49
	END	SHUF	50

Figure 94. Subroutine SHUF Program Listing

```

SUBROUTINE RESP(C,A,G2,AM,AK,X,Y,Z,S,R,ES,E,U,V,XI,DQ,AKG,DDQ,HDK,RESP 2
1KWA,NX,NFF,NN,NM,NI,NR,MX,MFF,MFB,MN,MM,MU,MR,ITER,IMAX,IERR,NCOV)RESP 3
DIMENSION XI(MFF,MFF),X(MX,MX),R(MX,MM),HDK(MP,MX),C(MX,MX),RESP 4
IG2(MX,MN),A(MX,MX),AM(MM,MX),AK(MU,MM),Y(MX,MX),Z(MX,MX),S(MX,MX),RESP 5
2ES(MX,MX),U(MFB,MFR),V(MFF,MFF),E(MFF,MFF),DQ(MU,MM),RESP 6
3KWA(MX),AKG(MU,MX),DDQ(MU,MU)RESP 7
NFB=NX-NFFRESP 8
CRESP 9
C COMPUTE COVARIANCE MATRIX FOR DISTURBANCE KCOMRESP 10
DO 6080 I=1,NRRESP 11
DO 6080 J=1,NRRESP 12
6080 R(I,J)=0.RESP 13
KCOM=0RESP 14
6076 KCOM=KCOM+1RESP 15
DO 4020 I=1,NFFRESP 16
DO 4020 J=1,NFFRESP 17
II=I+NFBRESP 18
JJ=J+NFBRESP 19
4020 C(I,J)=G2(II,KCOM)+G2(JJ,KCOM)RESP 20
WRITE(9,41) KCOMRESP 21
41 FORMAT(1H1/7X,36H COVARIANCE ANALYSIS FOR DISTURBANCE,13//)RESP 22
ITER=0RESP 23
CALL COVAR(XI,A,C,X,G2,S,E,FS,V,U,NX,NFF,NN,MX,MFF,MFB,MN,IMAX,RESP 24
1ITER,2,IERR,KWA)RESP 25
IF(IERR.EQ.0) GO TO 896RESP 26
WRITE(9,43)RESP 27
43 FORMAT(1H1/7X,28H COVARIANCE MATRIX UNDEFINED//7X,27H IGNORE COVARRESP 28
1ANCE ANALYSIS//)RESP 29
RETURNRESP 30
896 WRITE(9,4051)RESP 31
4051 FORMAT(//7X,18H COVARIANCE MATRIX//)RESP 32
CALL MP(MX,MX,NX,NX,XI)RESP 33
CRESP 34
C COMPUTE (H+DKM)X(H+DKM)RESP 35
DO 4053 I=1,NRRESP 36
DO 4053 J=1,NXRESP 37
C(I,J)=0.RESP 38
DO 4053 K=1,NXRESP 39
4053 C(I,J)=C(I,J)+HDK(I,K)*X(K,J)RESP 40
DO 4054 I=1,NRRESP 41
DO 4054 J=1,NRRESP 42
S(I,J)=0.RESP 43
DO 4054 K=1,NXRESP 44
4054 S(I,J)=S(I,J)+C(I,K)*HDK(J,K)RESP 45
IF(NCOV.GT.2) GO TO 2RESP 46
WRITE(9,42)RESP 47
42 FORMAT(1H1/7X,27H RESPONSE COVARIANCE MATRIX//)RESP 48
CALL MP(MX,MX,NR,NR,S)RESP 49
DO 7015 I=1,NXRESP 50
DO 7015 J=1,NMRESP 51
ES(J,I)=0.RESP 52
DO 7015 K=1,NXRESP 53
7015 ES(J,I)=ES(J,I)+X(I,K)*AM(J,K)RESP 54
DO 7016 I=1,NMRESP 55
DO 7016 J=1,NMRESP 56
Y(I,J)=0.RESP 57
DO 7016 K=1,NXRESP 58
7016 Y(I,J)=Y(I,J)+AM(I,K)*ES(J,K)RESP 59
WRITE(9,44)RESP 60
44 FORMAT(1H1/7X,30H MEASUREMENT COVARIANCE MATRIX//)RESP 61
CALL MP(MX,MX,NM,NM,Y)RESP 62
DO 1112 I=1,NURESP 63
DO 1112 J=1,NMRESP 64

```

Figure 95. Subroutine RESP Program Listing

	DQ(I,J)=0.	HESP 65
	DO 1112 K=1,NM	RESP 66
1112	DQ(I,J)=DQ(I,J)+AK(I,K)*Y(K,J)	RESP 67
	DO 6085 I=1,NU	RESP 68
	DO 6085 J=1,NU	RESP 69
	DOD(I,J)=0.	RESP 70
	DO 6085 K=1,NM	RESP 71
6085	DOD(I,J)=DOD(I,J)+DQ(I,K)*AK(J,K)	RESP 72
	WRITE(9,45)	RESP 73
	45 FORMAT(1H1/7X,26H CONTROL COVARIANCE MATRIX//)	HESP 74
	CALL MP(MU,MU,NU,NU,DOD)	HESP 75
	DO 1111 I=1,NX	RESP 76
	DO 1111 J=1,NX	RESP 77
	Z(I,J)=0.	RESP 78
	IF(X(I,I).LT.1.E-20) GO TO 1111	RESP 79
	IF(X(J,J).LT.1.E-20) GO TO 1111	RESP 80
	Z(I,J)=X(I,J)/SQRT(X(I,I)*X(J,J))	RESP 81
1111	CONTINUE	RESP 82
	WRITE(9,46)	RESP 83
	46 FORMAT(1H1/7X,31H STATE CROSS-CORRELATION MATR(X//)	RESP 84
	CALL MP(MX,MX,NX,NX,Z)	HESP 85
	DO 1122 I=1,NU	RESP 86
	DO 1122 J=1,NX	RESP 87
	AKG(I,J)=0.	RESP 88
	DO 1122 K=1,NM	RESP 89
1122	AKG(I,J)=AKG(I,J)+AK(I,K)*AM(K,J)	RESP 90
	DO 1113 I=1,NU	RESP 91
	DO 1113 J=1,NX	RESP 92
	Z(I,J)=0.	RESP 93
	IF(DOD(I,I).LT.1.E-20) GO TO 1113	RESP 94
	IF(X(I,J).LT.1.E-20) GO TO 1113	RESP 95
	DO 1123 K=1,NM	RESP 96
1123	Z(I,J)=Z(I,J)+AKG(I,K)*X(K,J)	RESP 97
	Z(I,J)=Z(I,J)/SQRT(DOD(I,I)*X(J,J))	RESP 98
1113	CONTINUE	RESP 99
	WRITE(9,47)	RESP 100
	47 FORMAT(1H1/7X,39H CONTROL-STATE CROSS-CORRELATION MATRIX//)	RESP 101
	CALL MP(MX,MX,NU,NX,Z)	RESP 102
	DO 1114 I=1,NR	RESP 103
	DO 1114 J=1,NX	RESP 104
	Z(I,J)=0.	RESP 105
	IF(S(I,I).LT.1.E-20) GO TO 1114	RESP 106
	IF(X(J,J).LT.1.E-20) GO TO 1114	RESP 107
	Z(I,J)=C(I,J)/SQRT(S(I,I)*X(J,J))	RESP 108
1114	CONTINUE	RESP 109
	WRITE(9,48)	RESP 110
	48 FORMAT(1H1/7X,40H RESPONSE-STATE CROSS-CORRELATION MATRIX//)	RESP 111
	CALL MP(MX,MX,NR,NX,Z)	HESP 112
	DO 1115 I=1,NM	RESP 113
	DO 1115 J=1,NM	RESP 114
	Z(I,J)=0.	RESP 115
	IF(Y(I,I).LT.1.E-20) GO TO 1115	RESP 116
	IF(Y(J,J).LT.1.E-20) GO TO 1115	RESP 117
	Z(I,J)=Y(I,J)/SQRT(Y(I,I)*Y(J,J))	RESP 118
1115	CONTINUE	RESP 119
	WRITE(9,49)	RESP 120
	49 FORMAT(1H1/7X,37H MEASUREMENT CROSS-CORRELATION MATRIX//)	RESP 121
	CALL MP(MX,MX,NM,NM,Z)	RESP 122
	DO 1300 I=1,NM	RESP 123
	DO 1300 J=1,NX	RESP 124
	Z(I,J)=0.	RESP 125
	IF(Y(I,I).LT.1.E-20) GO TO 1300	RESP 126
	IF(X(J,J).LT.1.E-20) GO TO 1300	RESP 127
	DO 1301 K=1,NM	RESP 128
1301	Z(I,J)=Z(I,J)+AM(I,K)*X(K,J)	RESP 129
	Z(I,J)=Z(I,J)/SQRT(Y(I,I)*X(J,J))	RESP 130

Figure 95. Subroutine RESP Program Listing (Continued)

1300	CONTINUE	RESP 131
	WRITE(9,1302)	RESP 132
1302	FORMAT(1H1/7X,43H MEASUREMENT-STATE CROSS-CORRELATION MATRIX//)	RESP 133
	CALL MP(MX,MX,NX,NX,Z)	RESP 134
	DO 1116 I=1,NU	RESP 135
	DO 1116 J=1,NM	RESP 136
	Z(I,J)=0.	RESP 137
	IF(DQN(I,I).LT.1.E-20) GO TO 1116	RESP 138
	IF(Y(I,J).LT.1.E-20) GO TO 1116	RESP 139
	Z(I,J)=DQ(I,J)/SQRT(DQN(I,I)*Y(J,J))	RESP 140
1116	CONTINUE	RESP 141
	WRITE(9,50)	RESP 142
50	FORMAT(1H1/7X,45H CONTROL-MEASUREMENT CROSS-CORRELATION MATRIX//)	RESP 143
	CALL MP(MX,MX,NI,NM,Z)	RESP 144
	DO 1118 I=1,NR	RESP 145
	DO 1118 J=1,NM	RESP 146
	Z(I,J)=0.	RESP 147
	IF(S(I,I).LT.1.E-20) GO TO 1118	RESP 148
	IF(Y(I,J).LT.1.E-20) GO TO 1118	RESP 149
	DO 1119 K=1,NX	RESP 150
1119	Z(I,J)=Z(I,J)+C(I,K)*AM(J,K)	RESP 151
	Z(I,J)=Z(I,J)/SQRT(S(I,I)*Y(J,J))	RESP 152
1118	CONTINUE	RESP 153
	WRITE(9,51)	RESP 154
51	FORMAT(1H1/7X,46H RESPONSE-MEASUREMENT CROSS-CORRELATION MATRIX//)	RESP 155
	CALL MP(MX,MX,NR,NM,Z)	RESP 156
	DO 1120 I=1,NU	RESP 157
	DO 1120 J=1,NR	RESP 158
	Z(I,J)=0.	RESP 159
	IF(DQN(I,I).LT.1.E-20) GO TO 1120	RESP 160
	IF(S(J,J).LT.1.E-20) GO TO 1120	RESP 161
	DO 1121 K=1,NX	RESP 162
1121	Z(I,J)=Z(I,J)+AKG(I,K)*C(J,K)	RESP 163
	Z(I,J)=Z(I,J)/SQRT(DQN(I,I)*S(J,J))	RESP 164
1120	CONTINUE	RESP 165
	WRITE(9,52)	RESP 166
52	FORMAT(1H1/7X,42H CONTROL-RESPONSE CROSS-CORRELATION MATRIX//)	RESP 167
	CALL MP(MX,MX,NU,NP,Z)	RESP 168
2	DO 4056 I=1,NR	RESP 169
	IF(S(I,I).LT.0.) S(I,I)=0.	RESP 170
	R(I,I)=R(I,I)+S(I,I)	RESP 171
4056	S(I,I)=SQRT(S(I,I))	RESP 172
	WRITE(9,4057)((I,S(I,I)),I=1,NR)	RESP 173
4057	FORMAT(//20X,17H R.M.S. RESPONSES/(18X,13,E16,A))	RESP 174
	IF(KCON.LT.NN) GO TO 6076	RESP 175
	DO 6082 I=1,NR	RESP 176
6082	R(I,I)=SQRT(R(I,I))	RESP 177
	WRITE(9,6081)((I,R(I,I)),I=1,NR)	RESP 178
6081	FORMAT(//7X,22HTOTAL R.M.S. RESPONSES/(18X,13,E16.8))	RESP 179
	RETURN	RESP 180
	END	RESP 181

Figure 95. Subroutine RESP Program Listing (Concluded)

```

SUBROUTINE COVAR(XI,A,C,X,G2,S,E,ES,V,U,NX,NFF,NN,MX,MFF,MFB,MN, COVAR 2
IMAX,ITER,IR,IERR,KWA) COVAR 3
DIMENSION A(MX,MX),C(MX,MX),G2(MX,MN),X(MX,MX),S(MX,MX),XI(MFF,MFF COVAR 4
),KWA(MX),E(MFF,MFF),FS(MX,MX),U(MFB,MFB),V(MFF,MFF) COVAR 5
NFR=NX-NFF COVAR 6
IF(ITER.NE.0) GO TO 150 COVAR 7
C COVAR 8
C COVAR 9
C COVAR 10
C COVAR 11
C COMPUTE X22 FROM G=A22*X22+X22*A22+G22*G22 COVAR 12
9 DO 11 I=1,NFF COVAR 13
DO 11 J=1,NFF COVAR 14
II=I+NFR COVAR 15
JJ=J+NFR COVAR 16
S(I,J)=A(II,JJ) COVAR 17
IF(IR.EQ.2) GO TO 11 COVAR 18
C(I,J)=0. COVAR 19
DO 12 K=1,NN COVAR 20
12 C(I,J)=C(I,J)+G2(II,K)*G2(JJ,K) COVAR 21
11 CONTINUE COVAR 22
CALL CAL(S,C,X,KWA,NFF,MX,IMAX,2,IERR) COVAR 23
IF(IERR.GT.0) RETURN COVAR 24
DO 151 I=1,NFF COVAR 25
DO 151 J=1,NFF COVAR 26
151 XI(I,J)=C(I,J) COVAR 27
C COVAR 28
C COMPUTE X12 FROM G=A11*X12+X12*A22+A12*X22 COVAR 29
150 DO 152 I=1,NFR COVAR 30
DO 152 L=1,NFR COVAR 31
152 X(I,L)=A(I,L) COVAR 32
DO 152 J=1,NFF COVAR 33
C(I,J)=0. COVAR 34
DO 152 K=1,NFF COVAR 35
KK=K+NFR COVAR 36
152 C(I,J)=C(I,J)+A(I,KK)*X(K,J) COVAR 37
DO 152 I=1,NFF COVAR 38
DO 152 J=1,NFF COVAR 39
II=I+NFR COVAR 40
JJ=J+NFR COVAR 41
153 S(I,J)=A(JJ,II) COVAR 42
CALL GCAL(S,X,C,NFF,NFR,MFF,MFB,IMAX,E,ES,V,U,KWA,MX,IERR) COVAR 43
IF(IERR.GT.0) RETURN COVAR 44
C COVAR 45
C COMPUTE X11 FROM G=A11*X11+X11*A11+A12*X12+X12*A12 COVAR 46
DO 154 I=1,NFR COVAR 47
DO 154 J=1,NFR COVAR 48
X(I,J)=A(I,J) COVAR 49
S(I,J)=0. COVAR 50
DO 154 K=1,NFF COVAR 51
KK=K+NFR COVAR 52
154 S(I,J)=S(I,J)+A(I,KK)*C(J,K)+C(I,K)*A(J,KK) COVAR 53
CALL CAL(X,S,ES,KWA,NFR,MX,IMAX,2,IERR) COVAR 54
IF(IERR.GT.0) RETURN COVAR 55
DO 155 I=1,NFR COVAR 56
DO 155 J=1,NFF COVAR 57
JJ=J+NFR COVAR 58
X(I,J)=C(I,J) COVAR 59
155 X(JJ,K)=X(I,JJ) COVAR 60
DO 154 I=1,NFF COVAR 61
DO 154 J=1,NFF COVAR 62
II=I+NFR COVAR 63
JJ=J+NFR COVAR 64

```

Figure 96. Subroutine COVAR Program Listing

```
156 X(I,J)=A(I,J)
    DO 157 I=1,NFR
      DO 157 J=1,NFR
157 X(I,J)=S(I,J)
    RETURN
    END
```

```
COVAR 65
COVAR 66
COVAR 67
COVAR 68
COVAR 69
COVAR 70
```

Figure 96. Subroutine COVAR Program Listing (Concluded)

```

SURROUTINE COSTAT(R,A,S,X,ES,Y,Z,E,U,V,KWA,MX,MFH,MFF,NX,NFF,IMAX,COSTAT 2
IERR) COSTAT 3
DIMENSION Y(MX,MX),A(MX,MX),S(MX,MX),Z(MX,MX),FS(MX,MX),X(MX,MX), COSTAT 4
IU(MFB,MFB),R(MX,MX),V(MFF,MFF),KWA(MX),E(MFF,MFF) COSTAT 5
NFR=NX-NFF COSTAT 6
DO 1 I=1,NX COSTAT 7
DO 1 J=1,NX COSTAT 8
S(I,J)=R(I,J) COSTAT 9
1 X(I,J)=A(I,J) COSTAT10
C COSTAT11
C COMPUTE S11 FROM 0=S11*A11+A11*S11+R11 COSTAT12
CALL CAL(X,S,ES,KWA,NFR,MX,IMAX,1,IERR) COSTAT13
IF(IERR.GT.0) RETURN COSTAT14
C COSTAT15
C COMPUTE S12 FROM 0=S12*A11+A22*S12+S11*A12+R12 COSTAT16
DO 15A I=1,NFR COSTAT17
DO 15A J=1,NFF COSTAT18
JJ=J+NFB COSTAT19
ES(I,J)=R(I,JJ) COSTAT20
DO 15A K=1,NFR COSTAT21
158 ES(I,J)=ES(I,J)+S(I,K)*A(K,JJ) COSTAT22
DO 15A I=1,NFR COSTAT23
DO 15A J=1,NFR COSTAT24
159 X(I,J)=A(J,I) COSTAT25
DO 160 I=1,NFF COSTAT26
DO 160 J=1,NFF COSTAT27
II=I+NFB COSTAT28
JJ=J+NFB COSTAT29
160 Y(I,J)=A(II,JJ) COSTAT30
CALL GCAL(Y,X,ES,NFF,NFR,MFF,MFB,IMAX,E,Z,V,U,KWA,MX,IERR) COSTAT31
IF(IERR.GT.0) RETURN COSTAT32
DO 162 I=1,NFR COSTAT33
DO 162 J=1,NFF COSTAT34
JJ=J+NFB COSTAT35
S(JJ,I)=ES(I,J) COSTAT36
162 S(I,JJ)=ES(I,J) COSTAT37
C COSTAT38
C COMPUTE S22 FROM 0=S22*A22+A22*S22+A12*S12+S12*A12+R22 COSTAT39
DO 163 I=1,NFF COSTAT40
DO 163 J=1,NFF COSTAT41
II=I+NFB COSTAT42
JJ=J+NFB COSTAT43
Y(I,J)=A(II,JJ) COSTAT44
X(I,J)=R(II,JJ) COSTAT45
DO 163 K=1,NFR COSTAT46
163 X(I,J)=X(I,J)+A(K,II)*S(K,JJ)+S(K,II)*A(K,JJ) COSTAT47
CALL CAL(Y,X,R,KWA,NFF,MX,IMAX,1,IERR) COSTAT48
IF(IERR.GT.0) RETURN COSTAT49
DO 164 I=1,NFF COSTAT50
DO 164 J=1,NFF COSTAT51
II=I+NFB COSTAT52
JJ=J+NFB COSTAT53
164 S(II,JJ)=X(I,J) COSTAT54
RETURN COSTAT55
END COSTAT56

```

Figure 97. Subroutine COSTAT Program Listing

SUBROUTINE TRANS(AMT,X,T,DDO,NX,MX,NF,MU,MF,9,IF)	TRANS 2
DIMENSION AMT(MF,MX),X(MX,MX),T(MF,MF),DDO(MU,MU),R(MF,MX),IF(MF)	TRANS 3
DO 1 I=1,NF	TRANS 4
DO 1 J=1,NX	TRANS 5
R(I,J)=0.	TRANS 6
DO 1 K=1,NX	TRANS 7
1 R(I,J)=R(I,J)+AMT(I,K)*X(K,J)	TRANS 8
DO 2 I=1,NF	TRANS 9
DO 2 J=1,NF	TRANS 10
T(I,J)=0.	TRANS 11
L=IF(I)	TRANS 12
M=IF(J)	TRANS 13
DO 2 K=1,NX	TRANS 14
2 T(I,J)=T(I,J)+DDO(L,M)*R(I,K)*AMT(J,K)	TRANS 15
RETURN	TRANS 16
END	TRANS 17

Figure 98. Subroutine TRANS Program Listing

	SUBROUTINE UNSCR(T,DJDK,DJVV,DJVT,IF,JF,NF,NU,NM,MI,MM,MF)	UNSCR 2
	DIMENSION T(MF,MF),DJDK(MU,MM),DJV(MF),DJVT(MF)	UNSCR 3
	DIMENSION IF(MF),JF(MF)	UNSCR 4
	L=1	UNSCR 5
	DO 1 I=1,NU	UNSCR 6
	DO 1 J=1,NM	UNSCR 7
	IF(L.GT.NF) GO TO 1	UNSCR 8
	IF(I.NF.IF(L)) GO TO 1	UNSCR 9
	IF(J.NF.JF(L)) GO TO 1	UNSCR 10
	DJVV(L)=DJDK(I,J)	UNSCR 11
	L=L+1	UNSCR 12
1	CONTINUE	UNSCR 13
	DO 3 I=1,NF	UNSCR 14
	DJVT(I)=0.	UNSCR 15
	DO 3 K=1,NF	UNSCR 16
3	DJVT(I)=DJVT(I)+T(I,K)*DJV(K)	UNSCR 17
	L=1	UNSCR 18
	DO 4 I=1,NU	UNSCR 19
	DO 4 J=1,NM	UNSCR 20
	IF(L.GT.NF) GO TO 5	UNSCR 21
	IF(I.NF.IF(L)) GO TO 5	UNSCR 22
	IF(J.NF.JF(L)) GO TO 5	UNSCR 23
	DJDK(I,J)=DJVT(L)	UNSCR 24
	L=L+1	UNSCR 25
	GO TO 4	UNSCR 26
5	DJDK(I,J)=0.	UNSCR 27
4	CONTINUE	UNSCR 28
	RETURN	UNSCR 29
	END	UNSCR 30

Figure 99. Subroutine UNSCR Program Listing

```

SUBROUTINE GCAL(A,R,X,N,M,NN,MM,IMAX,E,ES,V,U,KWA,MX,IERR)
C
C THIS SUBROUTINE SOLVES THE GENERAL MATRIX EQUATION  $XA+BA=C$ 
C FORM  $(I-A)(I+R)$ ,  $(I-H)$ , AND  $(I+B)$  THEN INVERT  $(I-A)$  AND  $(I-H)$ 
DIMENSION A(MX,MX),H(MX,MX),X(MX,MX),E(NN,NN)
DIMENSION ES(MX,MX),V(NN,NN),U(MM,MM),KWA(MX)
FF=.01
IFRR=
DO 1 I=1,N
DO 2 J=1,N
2 E(I,J) = -A(I,J)
1 E(I,I)=E(I,I)+1.
DO 3 I=1,M
DO 4 J=1,M
4 ES(I,J) = -H(I,J)
3 ES(I,I)=ES(I,I)+1.
NRR=N
NCC=N
NR=NN
CALL TDINVR(ISOL,IDSOL,NRR,NCC,E,NR,KWA,DET)
IF((ISOL+IDSOL).LE.?) GO TO 5
IERR=
GO TO 6
5 NR=MX
CALL TDINVR(ISOL,IDSOL,M,M,ES,NR,KWA,DET)
IF((ISOL+IDSOL).LE.?) GO TO 6
IERR=
GO TO 6
6 NC=N*
C
C FORM U,V, AND W
C
DO 11 I=1,N
DO 11 J=1,N
V(I,J) = E(I,J)
DO 11 K=1,N
11 V(I,J) = V(I,J) + A(I,K)*E(K,J)
DO 12 I=1,M
DO 12 J=1,M
U(I,J) = ES(I,J)
DO 12 K=1,M
12 U(I,J) = U(I,J)+ES(I,K)*B(K,J)
DO 13 I=1,M
DO 13 J=1,N
B(I,J)=0.
DO 14 K=1,M
14 B(I,J) = B(I,J) + ES(I,K) * A(K,J)
13 B(I,J)=2.*B(I,J)
DO 15 I=1,M
DO 15 J=1,N
X(I,J) = 0.
DO 15 K=1,N
15 X(I,J)= X(I,J) + H(I,K) * E(K,J)
ITER=
100 CONTINUE
DO 30 I=1,M
DO 30 J=1,N
B(I,J)=0.
DO 30 K=1,M
30 B(I,J) = B(I,J) + H(I,K)*X(K,J)
C CONVERGENCE CHECK
C
ICOT=

```

```

GCAL 2
GCAL 3
GCAL 4
GCAL 5
GCAL 6
GCAL 7
GCAL 8
GCAL 9
GCAL 10
GCAL 11
GCAL 12
GCAL 13
GCAL 14
GCAL 15
GCAL 16
GCAL 17
GCAL 18
GCAL 19
GCAL 20
GCAL 21
GCAL 22
GCAL 23
GCAL 24
GCAL 25
GCAL 26
GCAL 27
GCAL 28
GCAL 29
GCAL 30
GCAL 31
GCAL 32
GCAL 33
GCAL 34
GCAL 35
GCAL 36
GCAL 37
GCAL 38
GCAL 39
GCAL 40
GCAL 41
GCAL 42
GCAL 43
GCAL 44
GCAL 45
GCAL 46
GCAL 47
GCAL 48
GCAL 49
GCAL 50
GCAL 51
GCAL 52
GCAL 53
GCAL 54
GCAL 55
GCAL 56
GCAL 57
GCAL 58
GCAL 59
GCAL 60
GCAL 61
GCAL 62
GCAL 63
GCAL 64

```

Figure 100. Subroutine GCAL Program Listing

DO 31 I=1,M	GCAL 65
DO 31 J=1,N	GCAL 66
DX=0.	GCAL 67
DO 32 K=1,N	GCAL 68
32 DX = DX + R(I,K)*V(K,J)	GCAL 69
X(I,J) = X(I,J) + DX	GCAL 70
AX=AR*(X(I,J))	GCAL 71
IF(AX.LT.1.E-20) GO TO 42	GCAL 72
IF(AX.LT.1.E 20) GO TO 41	GCAL 73
IFRR=.	GCAL 74
GO TO 6.1	GCAL 75
41 RAT=ARS(DX/X(I,J))	GCAL 76
IF(RAT-EE)42,42,43	GCAL 77
42 ICOT=ICOT+1	GCAL 78
43 CONTINUE	GCAL 79
31 CONTINUE	GCAL 80
ITER=ITER+1	GCAL 81
IF(ICOT-NC)44,5)44	GCAL 82
44 CONTINUE	GCAL 83
DO 33 I=1,N	GCAL 84
DO 33 J=1,N	GCAL 85
33 E(I,J)=V(I,J)	GCAL 86
DO 34 I=1,M	GCAL 87
DO 34 J=1,M	GCAL 88
34 ES(I,J)=U(I,J)	GCAL 89
DO 45 I=1,N	GCAL 90
DO 45 J=1,N	GCAL 91
V(I,J)=0.	GCAL 92
DO 45 K=1,N	GCAL 93
45 V(I,J) = V(I,J) + F(I,K)*E(K,J)	GCAL 94
DO 46 I=1,M	GCAL 95
DO 46 J=1,M	GCAL 96
U(I,J)=0.	GCAL 97
DO 46 K=1,M	GCAL 98
46 U(I,J) = U(I,J) + ES(I,K)*ES(K,J)	GCAL 99
IF(ITER.LT.IMAX) GO TO 100	GCAL 100
WRITE(9,600)	GCAL 101
600 FORMAT(/7X,12H ITER = IMAX)	GCAL 102
50 CONTINUE	GCAL 103
RETURN	GCAL 104
601 WRITE(9,602) IFRR	GCAL 105
602 FORMAT(/7X,7H IEPR =.I2)	GCAL 106
RETURN	GCAL 107
END	GCAL 108

Figure 100. Subroutine GCAL Program Listing (Concluded)

SUBROUTINE CAL (A,XN,P,KWA,N,NP,IMAX,IT,IERR)	CAL	2
DIMENSION A(NR,1),XN(NR,1),P(NR,1),KWA(NR)	CAL	3
IERR=	CAL	4
TR=0.	CAL	5
DO 30 I=1,N	CAL	6
300 TR=TR+A(I,I)	CAL	7
FN=N	CAL	8
IF (TR) 301,2,2	CAL	9
2 IERR=	CAL	10
GO TO 601	CAL	11
301 ALF=ABS(TR)/FN	CAL	12
EE=.01	CAL	13
NC=N*(N+1)	CAL	14
NC=NC/2	CAL	15
DO 66 I=1,N	CAL	16
DO 63 J=1,N	CAL	17
GOTO(4,62),IT	CAL	18
61 P(I,J)=A(I,J)	CAL	19
GOTO 43	CAL	20
62 P(I,J)=A(J,I)	CAL	21
63 CONTINUE	CAL	22
P(I,I)=P(I,I)-ALF	CAL	23
60 CONTINUE	CAL	24
CALL TDINVR (ISOL,IDSOL,N,NP,NR,KWA,DET)	CAL	25
IF ((ISOL+IDSOL).LE.2) GO TO 22	CAL	26
IERR=	CAL	27
GO TO 401	CAL	28
22 DO 4 I=1,N	CAL	29
DO 4 I=1,N	CAL	30
A(I,J)=0.	CAL	31
DO 4 K=1,N	CAL	32
4 A(I,J)=A(I,J)+P(K,I)*XN(K,J)*2.*ALF	CAL	33
DO 5 I=1,N	CAL	34
DO 5 J=1,N	CAL	35
XN(I,J)=0.	CAL	36
DO 5 K=1,N	CAL	37
5 XN(I,J)=XN(I,J)+A(I,K)*P(K,J)	CAL	38
DO 7 I=1,N	CAL	39
DO 8 J=1,N	CAL	40
8 P(I,J)=P(I,J)*2.*ALF	CAL	41
7 P(I,I)=P(I,I)+1.	CAL	42
ITER=	CAL	43
100 CONTINUE	CAL	44
DO 9 I=1,N	CAL	45
DO 9 J=1,N	CAL	46
A(I,J)=0.	CAL	47
DO 9 K=1,N	CAL	48
9 A(I,J)=A(I,J)+P(K,I)*XN(K,J)	CAL	49
ICOT=	CAL	50
DO 10 I=1,N	CAL	51
DO 10 J=1,N	CAL	52
DXIJ=.	CAL	53
DO 11 K=1,N	CAL	54
11 DXIJ=DXIJ+A(I,K)*P(K,J)	CAL	55
XN(I,J)=XN(I,J)+DXIJ	CAL	56
XN(J,I)=XN(I,J)	CAL	57
AXN=ABS(XN(I,J))	CAL	58
IF (AXN.LT.1.E-20) GO TO 14	CAL	59
IF (AXN.LT.1.E-20) GO TO 201	CAL	60
IERR=	CAL	61
GO TO 661	CAL	62
201 RAT=ABS(DXIJ/XN(I,J))	CAL	63
IF (RAT-EE)14,14,79	CAL	64

Figure 101. Subroutine CAL Program Listing

14 ICOT=ICOT+1	CAL 65
70 CONTINUE	CAL 66
10 CONTINUE	CAL 67
10 ITER=ITER+1	CAL 68
IF(ICOT-NC)15,50,15	CAL 69
15 CONTINUE	CAL 70
DO 20 I=1,N	CAL 71
DO 20 J=1,N	CAL 72
20 A(I,J)=P(I,J)	CAL 73
16 DO 17 I=1,N	CAL 74
DO 17 J=1,N	CAL 75
P(I,J)=0.	CAL 76
DO 17 K=1,N	CAL 77
17 P(I,J)=P(I,J)+A(I,K)*A(K,J)	CAL 78
40 IF(ITER.LT.IMAX) GO TO 100	CAL 79
WRITE(9,600)	CAL 80
600 FORMAT(/7X,12H ITER = IMAX)	CAL 81
50 CONTINUE	CAL 82
RETURN	CAL 83
601 WRITE(9,602) IERR	CAL 84
602 FORMAT(/7X,6H IERR=12)	CAL 85
RETURN	CAL 86
END	CAL 87

Figure 101. Subroutine CAL Program Listing (Concluded)

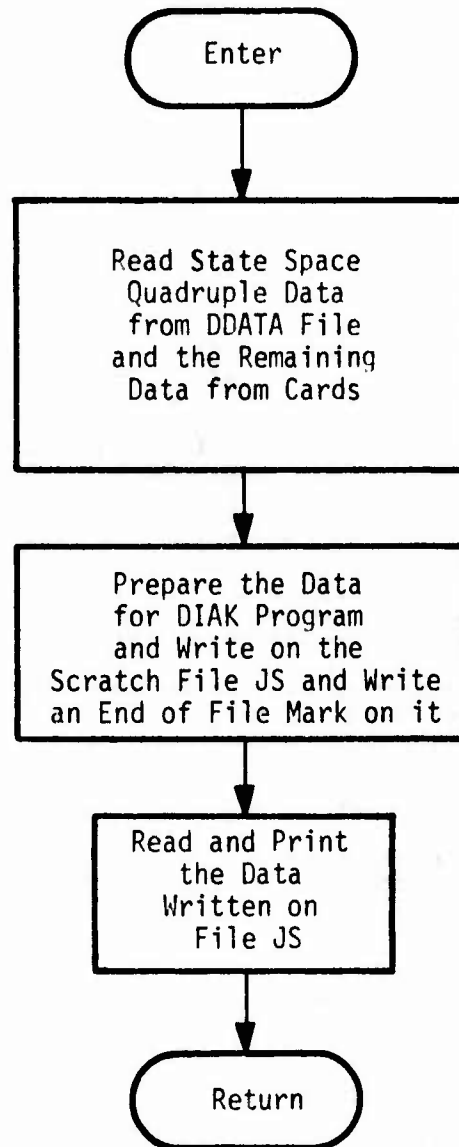


Figure 102. Subroutine DDLIAK Flow Chart

```

SUBROUTINE DDIAK(A,B,C,D,R1,B2,C1,C3,D11,RK,NXM,NRM,NUM)          DDIAK  2
C                                                                    DDIAK  3
C PURPOSE - TO PREPARE DATA FOR DIAK PROGRAM                     DDIAK  4
C ANALYSIS - A F KONAR / J K MAMESH - THE HONEYWELL INC         DDIAK  5
C DATE WRITTEN - 1975                                           DDIAK  6
C                                                                    DDIAK  7
C SUBPROGRAMS CALLED                                             DDIAK  8
C   ZERO                                                         DDIAK  9
C   FILE                                                         DDIAK 10
C   MPDS                                                         DDIAK 11
C   ERDM                                                         DDIAK 12
C   WTP                                                         DDIAK 13
C   INDTM                                                       DDIAK 14
C                                                                    DDIAK 15
C ARGUMENTS LIST                                               DDIAK 16
C   A                   STATE TRANSITION MATRIX                DDIAK 17
C   H                   CONTROL INPUT MATRIX                   DDIAK 18
C   C                   STATE OUTPUT MATRIX                    DDIAK 19
C   D                   CONTROL OUTPUT MATRIX                   DDIAK 20
C   B1                  INPUT MATRIX FOR CONTROL INPUTS - G1   DDIAK 21
C   B2                  INPUT MATRIX FOR GUST INPUTS - G2       DDIAK 22
C   C1                  STATE OUTPUT MATRIX FOR DESIGN OUTPUTS - M DDIAK 23
C   C3                  STATE OUTPUT MATRIX FOR MEASUREMENTS - M DDIAK 24
C   D11                OUTPUT MATRIX FOR DESIGN OUTPUTS - D    DDIAK 25
C   RK                  FEEDBACK GAIN MATRIX                    DDIAK 26
C   NXM                INPUT  MAXIMUM NO OF STATES             DDIAK 27
C   NRM                INPUT  MAXIMUM NO OF OUTPUTS            DDIAK 28
C   NUM                INPUT  MAXIMUM NO OF INPUTS             DDIAK 29
C                                                                    DDIAK 30
C   DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM)     DDIAK 31
C   DIMENSION R1(NXM,NUM),R2(NXM,NUM)                          DDIAK 32
C   DIMENSION C1(NRM,NXM),C3(NRM,NXM)                          DDIAK 33
C   DIMENSION D11(NRM,NUM),RK(NUM,NRM)                         DDIAK 34
C   DIMENSION HEAD(20),CARD(20)                                DDIAK 35
C   COMMON/INOUT/IR,IV,IWRITE,INSERT,LOCATE,NULL,MARK(20)    DDIAK 36
C   I,JQ,JS,JSD,JF,JD                                          DDIAK 37
C   DATA HRFH,HRGR,HRGZR,HRHRR/4H F ,4H G1 ,4H G2 ,4H H /    DDIAK 38
C   DATA HRDBR,HRBMR,HRCAR,HRREAD/4H D ,4H A4 ,1HC,4H CAR,4HREAD/ DDIAK 39
C   DATA HRTAP,HEND/4H TAP,4HEND /                             DDIAK 40
C   DATA HRHRH,HPRR/4H ,4HPE /                                 DDIAK 41
C   DATA HRBKB/4H RK /                                         DDIAK 42
C                                                                    DDIAK 43
C   READ IF DATA IS ON CARDS ONLY                             DDIAK 44
C                                                                    DDIAK 45
C   READ(IR,20)CARD                                             DDIAK 46
C 20 FORMAT(20A4)                                              DDIAK 47
C   IF(CARD(6).EQ.HHRH)GO TO 80                                  DDIAK 48
C   IF(CARD(6).NE.HPRR)GO TO 162                                DDIAK 49
C   CALL ZERO(A,NXM,NXM)                                         DDIAK 50
C   CALL ZERO(H,NXM,NUM)                                          DDIAK 51
C   CALL ZERO(C,NRM,NUM)                                          DDIAK 52
C   CALL ZERO(D,NRM,NUM)                                          DDIAK 53
C   CALL ZERO(RK,NUM,NRM)                                         DDIAK 54
C   READ(IR,20)HEAD                                             DDIAK 55
C   CALL FILE(JQ,LOCATE,HEAD)                                    DDIAK 56
C   READ(10)T,NX,NR,NU,((A(I,J),I=1,NX),J=1,NX),              DDIAK 57
C   1((R(I,J),I=1,NX),J=1,NU),((C(I,J),I=1,NR),J=1,NX),        DDIAK 58
C   2((D(I,J),I=1,NR),J=1,NU),NXA,NRA,NUA,VR1,NR2,NR3,NU1,NU2,NU3 DDIAK 59
C                                                                    DDIAK 60
C   PARTITION MATRICES B,C,D                                    DDIAK 61
C                                                                    DDIAK 62
C   IF(NU1.LE.0)STOP 111                                         DDIAK 63
C   IF(NU2.LE.0)STOP 111                                         DDIAK 64

```

Figure 103. Subroutine DDIAK Program Listing

IF(NR1.LE.1)STOP 111	DDIAK 65
IF(NR1.LE.2)STOP 111	DDIAK 66
DO 24 J=1,NX	DDIAK 67
DO 24 J=1,NU1	DDIAK 68
24 R1(I,J)=R(I,J)	DDIAK 69
DO 28 J=1,NU2	DDIAK 70
JJ=NU1+J	DDIAK 71
28 R2(I,J)=H(I,JJ)	DDIAK 72
DO 32 J=1,NX	DDIAK 73
DO 34 J=1,NR1	DDIAK 74
34 C1(I,J)=C(I,J)	DDIAK 75
DO 40 J=1,NR3	DDIAK 76
IT=NR1+NR2+J	DDIAK 77
40 C3(I,J)=C(I,J)	DDIAK 78
DO 44 J=1,NR1	DDIAK 79
DO 44 J=1,NU1	DDIAK 80
44 D1(I,J)=D(I,J)	DDIAK 81
IF(IP-INT,LT,6) GO TO R0	DDIAK 82
CALL PRS(A,NXM,NXM,NX,NX,T,4HA)	DDIAK 83
CALL PRS(R,NXM,NUM,NX,NU1,T,4HB)	DDIAK 84
CALL PRS(C,NRM,NXM,NR1,NX,T,4HC)	DDIAK 85
CALL PRS(D,NRM,NUM,NR1,NU1,T,4HD)	DDIAK 86
CALL PRS(R1,NXM,NUM,NX,NU1,T,4HB1)	DDIAK 87
CALL PRS(R2,NXM,NUM,NX,NU2,T,4HB2)	DDIAK 88
CALL PRS(C1,NRM,NXM,NR1,NX,T,4HC1)	DDIAK 89
CALL PRS(C3,NRM,NXM,NR3,NX,T,4HC3)	DDIAK 90
CALL PRS(D1,NRM,NUM,NR1,NU1,T,4HD1)	DDIAK 91
R0 CONTINUE	DDIAK 92
C ORGANIZE CARD AND TAPE DATA ON TAPE	DDIAK 93
C	DDIAK 94
C	DDIAK 95
100 READ(19,120) CARD	DDIAK 96
120 FORMAT(20A4)	DDIAK 97
IF((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HRTAP))GO TO 160	DDIAK 98
IF((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HRCAR))GO TO 100	DDIAK 99
IF(CARD(1).EQ.HEND) GO TO 3 3	DDIAK100
WRITE(JS,120) CARD	DDIAK101
GO TO 100	DDIAK102
160 CONTINUE	DDIAK103
IF(CARD(6).EQ.HRFHR) GO TO 180	DDIAK104
IF(CARD(6).EQ.HHGP) GO TO 200	DDIAK105
IF(CARD(6).EQ.HRG2R) GO TO 220	DDIAK106
IF(CARD(6).EQ.HRHRR) GO TO 240	DDIAK107
IF(CARD(6).EQ.HRDRR) GO TO 260	DDIAK108
IF(CARD(6).EQ.HRAMR) GO TO 280	DDIAK109
IF(CARD(6).EQ.HRRKR)GO TO 295	DDIAK110
162 CONTINUE	DDIAK111
WRITE(IW,165)	DDIAK112
165 FORMAT(//IX,24HINPUT CONTROL CARD ERROR)	DDIAK113
CALL FRM(1,4HDDIA,4HK .3,0,IW)	DDIAK114
C WRITE MATRIX DATA ON SCRATCH FILE FOR DIAK PROGRAM	DDIAK115
C	DDIAK116
C	DDIAK117
180 CONTINUE	DDIAK118
CALL WTP(A,NX,NX,NXM,NXM,JS)	DDIAK119
GO TO 100	DDIAK120
200 CONTINUE	DDIAK121
CALL WTP(R1,NX,NU1,NXM,NUM,JS)	DDIAK122
GO TO 100	DDIAK123
220 CONTINUE	DDIAK124
CALL WTP(R2,NX,NU2,NXM,NUM,JS)	DDIAK125
GO TO 100	DDIAK126
240 CONTINUE	DDIAK127
CALL WTP(C1,NR1,NX,NRM,NXM,JS)	DDIAK128
GO TO 100	DDIAK129
260 CONTINUE	DDIAK130

Figure 103. Subroutine DDIK Program Listing (Continued)

CALL WTP(DI1,NR1,NU1,NRM,NUM,JS)	DDIAK131
GO TO 100	DDIAK132
280 CONTINUE	DDIAK133
CALL WTP(C3,NR3,NX,NRM,NXM,JS)	DDIAK134
GO TO 100	DDIAK135
285 CONTINUE	DDIAK136
C	DDIAK137
C READ GAIN MATRIX FROM DDATA FILE	DDIAK138
C	DDIAK139
READ(IR,20)HEAD	DDIAK140
290 CONTINUE	DDIAK141
READ(JD,20)CARD	DDIAK142
DO 295 J=1,20	DDIAK143
IF (CARD(J).NE.HEAD(I)) GO TO 290	DDIAK144
295 CONTINUE	DDIAK145
CALL INPTM(HK,NUM,NRM,JD)	DDIAK146
REWIND JD	DDIAK147
CALL WTP(BK,NU1,NR3,NUM,NRM,JS)	DDIAK148
GO TO 100	DDIAK149
300 CONTINUE	DDIAK150
END FILE JS	DDIAK151
REWIND JS	DDIAK152
IF ((IPRINT.LT.5).AND.(IPRINT.NE.3)) GO TO 400	DDIAK153
C	DDIAK154
C READ AND PRINT OUT TAPE	DDIAK155
C	DDIAK156
WRITE(IW,310)	DDIAK157
310 FORMAT(1H1,1X,23H*** DIAK INPUT DATA ***,//)	DDIAK158
320 CONTINUE	DDIAK159
READ(JS,120)CARD	DDIAK160
IF (EOF(JS)) 360,340	DDIAK161
340 WRITE(IW,350) CARD	DDIAK162
350 FORMAT(1X,20A4)	DDIAK163
GO TO 320	DDIAK164
360 CONTINUE	DDIAK165
REWIND JS	DDIAK166
WRITE(IW,380)	DDIAK167
380 FORMAT(//,1X,30H*** END OF DIAK INPUT DATA ***)	DDIAK168
400 CONTINUE	DDIAK169
RETURN	DDIAK170
END	DDIAK171

Figure 103. Subroutine DDIAK Program Listing (Concluded)

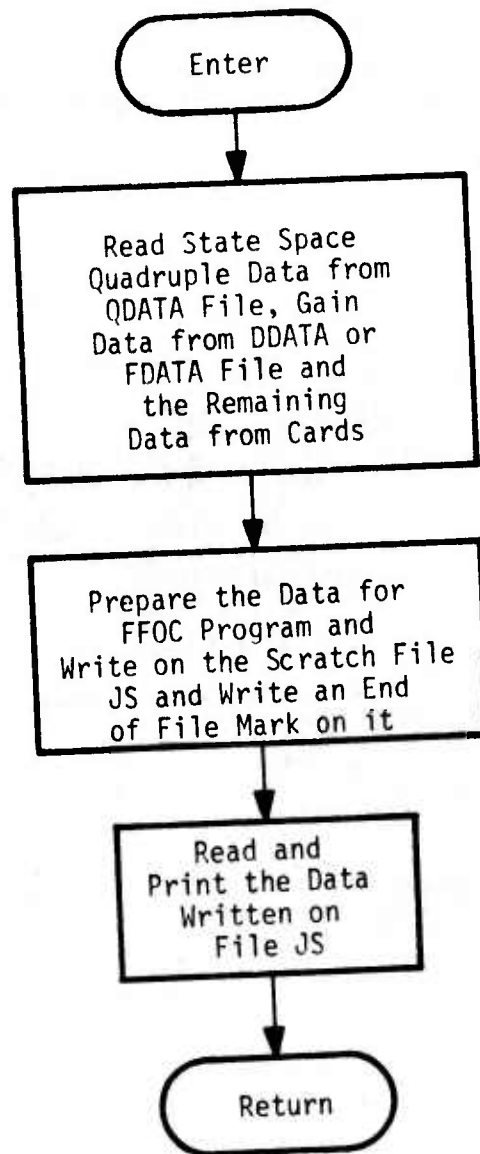


Figure 104. Subroutine DFFOC Flow Chart

```

C      SUBROUTINE DFFOC(A,B,C,D,B1,B2,C1,C3,D11,RK,NXM,NRM,NUM)      DFFOC  2
C      PURPOSE - TO PREPARE DATA FOR FFOC PROGRAM                  DFFOC  3
C      ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC        DFFOC  4
C      DATE WRITTEN - 1975                                          DFFOC  5
C                                                                    DFFOC  6
C                                                                    DFFOC  7
C      SUBPROGRAMS CALLED                                           DFFOC  8
C      ZERO                                                         DFFOC  9
C      FILE                                                         DFFOC 10
C      MPDS                                                         DFFOC 11
C      WTP                                                         DFFOC 12
C      INPTH                                                       DFFOC 13
C                                                                    DFFOC 14
C      ARGUMENTS LIST                                              DFFOC 15
C      A      STATE TRANSITION MATRIX                             DFFOC 16
C      B      CONTROL INPUT MATRIX                               DFFOC 17
C      C      STATE OUTPUT MATRIX                                DFFOC 18
C      D      CONTROL OUTPUT MATRIX                             DFFOC 19
C      B1     INPUT MATRIX FOR CONTROL INPUTS - G1              DFFOC 20
C      B2     INPUT MATRIX FOR GUST INPUTS - G2                 DFFOC 21
C      C1     STATE OUTPUT MATRIX FOR DESIGN OUTPUTS - H       DFFOC 22
C      C3     STATE OUTPUT MATRIX FOR MEASUREMENTS - H         DFFOC 23
C      D11    OUTPUT MATRIX FOR DESIGN OUTPUTS - D             DFFOC 24
C      RK     FEEDBACK GAIN MATRIX                              DFFOC 25
C      NXM    INPUT      MAXIMUM NO OF STATES                  DFFOC 26
C      NRM    INPUT      MAXIMUM NO OF OUTPUTS                 DFFOC 27
C      NUM    INPUT      MAXIMUM NO OF INPUTS                  DFFOC 28
C                                                                    DFFOC 29
C      DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM)    DFFOC 30
C      DIMENSION B1(NXM,NUM),B2(NXM,NUM)                        DFFOC 31
C      DIMENSION C1(NRM,NXM),C3(NRM,NXM)                       DFFOC 32
C      DIMENSION D11(NRM,NUM),RK(NUM,NRM)                      DFFOC 33
C      DIMENSION HEAD(20),CARD(20)                             DFFOC 34
C      COMMON/INOUT/IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20)   DFFOC 35
C      I,JQ,JS,JSD,JF,JD                                        DFFOC 36
C      DATA HRFBH,HRGIR,HRG2R,HRMBR/4H F ,4H G1 ,4H G2 ,4H H /  DFFOC 37
C      DATA HRDBB,HRAMB,HC,HRCAR,HRHAD/4H D ,4H AM ,JHC,4H CAR,4HREAD/ DFFOC 38
C      DATA HRTAP,HEND/4H TAP,4HEND /                          DFFOC 39
C      DATA HRBBB,HPERR/4H ,4HPP /                             DFFOC 40
C      DATA HRAKG,HRAKP,HRADEL/4H AKG,4H AK(,4H DEL/          DFFOC 41
C                                                                    DFFOC 42
C      READ IF DATA IS ON CARDS ONLY                            DFFOC 43
C                                                                    DFFOC 44
C      READ(IR,20)CARD                                           DFFOC 45
C      IF(CARD(6).EQ.HRRRH) GO TO 40                             DFFOC 46
C      IF(CARD(6).NE.HPERR) GO TO 162                            DFFOC 47
C      CALL ZERO(A,NXM,NXM)                                       DFFOC 48
C      CALL ZERO(B,NXM,NUM)                                        DFFOC 49
C      CALL ZERO(C,NRM,NXM)                                       DFFOC 50
C      CALL ZERO(D,NRM,NUM)                                       DFFOC 51
C      CALL ZERO(RK,NUM,NRM)                                       DFFOC 52
C      READ(IR,20)HEAD                                           DFFOC 53
C      20 FORMAT(20A4)                                           DFFOC 54
C      CALL FILE(JQ,LOCATE,HEAD)                                   DFFOC 55
C      READ(JQ)T,NX,NR,NU,((A(I,J),I=1,NX),J=1,NX),            DFFOC 56
C      1((B(I,J),I=1,NX),J=1,NU),((C(I,J),I=1,NR),J=1,NX),     DFFOC 57
C      2((D(I,J),I=1,NR),J=1,NU),NXA,NRA,NUA,NR1,NR2,NR3,NU1,NU2,NU3 DFFOC 58
C                                                                    DFFOC 59
C      PARTITION MATRICES B,C,D                                  DFFOC 60
C                                                                    DFFOC 61
C      IF(NU1.LE.0)STOP 111                                       DFFOC 62
C      IF(NU2.LE.0)STOP 111                                       DFFOC 63
C      IF(NR1.LE.0)STOP 111                                       DFFOC 64

```

Figure 105. Subroutine DFFOC Program Listing

	IF (NR1,LE,6) STOP 111	DDFOC 65
	DO 28 I=1,NX	DDFOC 66
	DO 24 J=1,NU1	DDFOC 67
24	B1(I,J)=B(I,J)	DDFOC 68
	DO 28 J=1,NU2	DDFOC 69
	JJ=NU1+J	DDFOC 70
28	B2(I,J)=B(I,JJ)	DDFOC 71
	DO 40 J=1,NX	DDFOC 72
	DO 34 I=1,NR1	DDFOC 73
34	C1(I,J)=C(I,J)	DDFOC 74
	DO 40 I=1,NR3	DDFOC 75
	II=NR1+NR2+I	DDFOC 76
40	C3(I,J)=C(II,J)	DDFOC 77
	DO 44 I=1,NR1	DDFOC 78
	DO 44 J=1,NU1	DDFOC 79
44	D11(I,J)=D(I,J)	DDFOC 80
	IF (IPOINT,LT,6) GO TO 80	DDFOC 81
	CALL *PRS(A,NX,NX,NX,NX,T,4HA)	DDFOC 82
	CALL *PRS(B,NX,NUM,NX,NU,T,4HB)	DDFOC 83
	CALL *PRS(C,NRM,NX,NR,NX,T,4HC)	DDFOC 84
	CALL *PRS(D,NRM,NUM,NR,NU,T,4HD)	DDFOC 85
	CALL *PRS(B1,NX,NUM,NX,NU1,T,4HB1)	DDFOC 86
	CALL *PRS(B2,NX,NUM,NX,NU2,T,4HB2)	DDFOC 87
	CALL *PRS(C1,NRM,NX,NR1,NX,T,4HC1)	DDFOC 88
	CALL *PRS(C3,NRM,NX,NR3,NX,T,4HC3)	DDFOC 89
	CALL *PRS(D11,NRM,NUM,NR1,NU1,T,4HD11)	DDFOC 90
80	CONTINUE	DDFOC 91
C		DDFOC 92
C	ORGANIZE CARD AND TAPE DATA ON TAPE	DDFOC 93
C		DDFOC 94
100	READ(IR,120) CARD	DDFOC 95
120	FORMAT(28A4)	DDFOC 96
	IF ((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HRTAP)) GO TO 160	DDFOC 97
	IF ((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HBCAR)) GO TO 100	DDFOC 98
	IF (CARD(1).EQ.HEND) GO TO 300	DDFOC 99
	WRITE(JS,120) CARD	DDFOC100
	GO TO 100	DDFOC101
160	CONTINUE	DDFOC102
	IF (CARD(6).EQ.HRFRR) GO TO 180	DDFOC103
	IF (CARD(6).EQ.HRG1R) GO TO 200	DDFOC104
	IF (CARD(6).EQ.HRG2R) GO TO 220	DDFOC105
	IF (CARD(6).EQ.HRHRR) GO TO 240	DDFOC106
	IF (CARD(6).EQ.HRDRR) GO TO 260	DDFOC107
	IF (CARD(6).EQ.HHAMR) GO TO 280	DDFOC108
	IF (CARD(6).EQ.HRAKR) GO TO 295	DDFOC109
	IF (CARD(6).EQ.HRAKP) GO TO 295	DDFOC110
	IF (CARD(6).EQ.HRDEL) GO TO 295	DDFOC111
162	CONTINUE	DDFOC112
	WRITE(IW,165)	DDFOC113
165	FORMAT(//IX,24MINPUT CONTROL CARD ERROR)	DDFOC114
	STOP 111	DDFOC115
C		DDFOC116
C	WRITE MATRIX DATA ON SCRATCH FILE FOR FFOC PROGRAM	DDFOC117
C		DDFOC118
180	CONTINUE	DDFOC119
	CALL *WTP(A,NX,NX,NX,NX,JS)	DDFOC120
	GO TO 100	DDFOC121
200	CONTINUE	DDFOC122
	CALL *WTP(B1,NX,NU1,NX,NUM,JS)	DDFOC123
	GO TO 100	DDFOC124
220	CONTINUE	DDFOC125
	CALL *WTP(B2,NX,NU2,NX,NUM,JS)	DDFOC126
	GO TO 100	DDFOC127
240	CONTINUE	DDFOC128
	CALL *WTP(C1,NR1,NX,NRM,NX,JS)	DDFOC129
	GO TO 100	DDFOC130

Figure 105. Subroutine DFFOC Program Listing (Continued)

260	CONTINUE	DFFOC131
	CALL WTP(D11,NR1,NI1,NRM,NUM,JS)	DFFOC132
	GO TO 100	DFFOC133
280	CONTINUE	DFFOC134
	CALL WTP(C3,NR3,NX,NRM,NUM,JS)	DFFOC135
	GO TO 100	DFFOC136
C		DFFOC137
C	READ GAINS FROM DDATA OR FDATA FILE	DFFOC138
C		DFFOC139
285	CONTINUE	DFFOC140
	JDF=JF	DFFOC141
	IF(CARD(6).EQ.HRAGG)JDF=JD	DFFOC142
	READ(1R,120)HEAD	DFFOC143
290	CONTINUE	DFFOC144
	READ(JDF,120)CARD	DFFOC145
	DO 295 I=1,20	DFFOC146
	IF(CARD(I).NE.HEAD(I))GO TO 290	DFFOC147
295	CONTINUE	DFFOC148
	CALL ZERO(BK,NUM,NPM)	DFFOC149
	CALL INPTM(BK,NUM,NPM,JDF)	DFFOC150
	REWIND JDF	DFFOC151
	CALL WTP(BK,NU1,NR1,NUM,NPM,JS)	DFFOC152
	GO TO 100	DFFOC153
300	CONTINUE	DFFOC154
	END FILE JS	DFFOC155
	REWIND JS	DFFOC156
	IF((IPRINT.LT.5).AND.(IPRINT.NE.3)) GO TO 400	DFFOC157
C		DFFOC158
C	READ AND PRINT OUT TAPE	DFFOC159
C		DFFOC160
	WRITE(IW,310)	DFFOC161
310	FORMAT(1H1,1X,23H*** FFOC INPUT DATA ***//)	DFFOC162
320	CONTINUE	DFFOC163
	READ(JS,120)CARD	DFFOC164
	IF (END(JS)) 360,340	DFFOC165
340	WRITE(IW,350) CARD	DFFOC166
350	FORMAT(//1X,20A4)	DFFOC167
	GO TO 320	DFFOC168
360	CONTINUE	DFFOC169
	REWIND JS	DFFOC170
400	CONTINUE	DFFOC171
	RETURN	DFFOC172
	END	DFFOC173

Figure 105. Subroutine DFFOC Program Listing (Concluded)

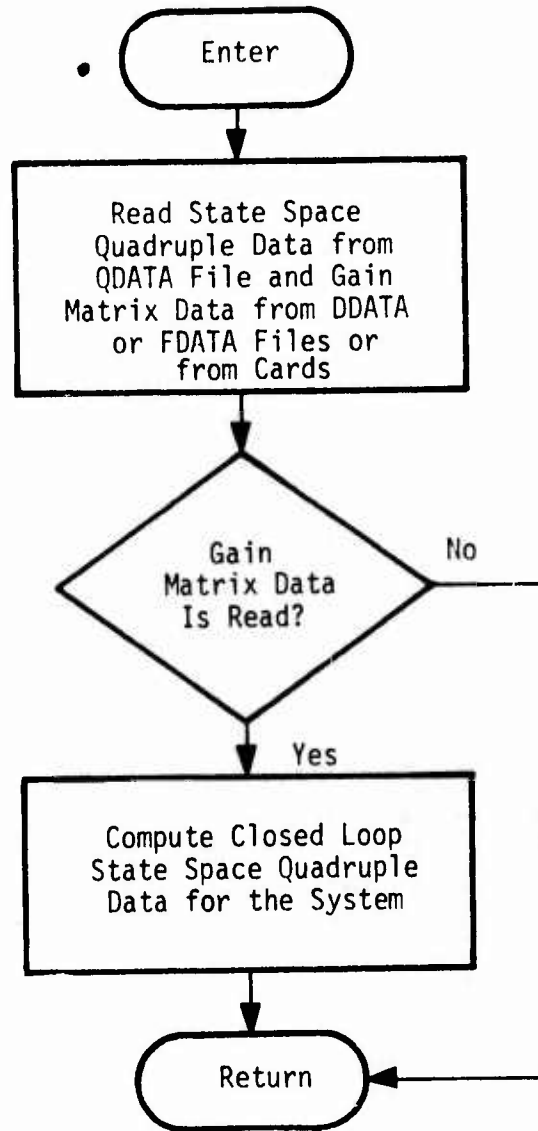


Figure 106. Subroutine DLSA Flow Chart

```

SUBROUTINE DL5A(A,R,C,D,B1,R2,C1,C3,D1),BK,BK03,NX,NR,NU, DLSA 2
1NXM,NNM,NUM) DLSA 3
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
PURPOSE - TO PREPARE DATA FOR LSA PROGRAM DLSA 4
ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC DLSA 5
DATE WRITTEN - 1975 DLSA 6
SUBPROGRAMS CALLED DLSA 7
ZERO DLSA 8
FILE DLSA 9
MPS DLSA 10
INPTM DLSA 11
ARGUMENTS LIST DLSA 12
A STATE TRANSITION MATRIX DLSA 13
H CONTROL INPUT MATRIX DLSA 14
C STATE OUTPUT MATRIX DLSA 15
D CONTROL OUTPUT MATRIX DLSA 16
B1 INPUT MATRIX FOR CONTROL INPUTS - G1 DLSA 17
B2 INPUT MATRIX FOR GUST INPUTS - G2 DLSA 18
C1 STATE OUTPUT MATRIX FOR DESIGN OUTPUTS - M DLSA 19
C3 STATE OUTPUT MATRIX FOR MEASUREMENTS - M DLSA 20
D1 OUTPUT MATRIX FOR DESIGN OUTPUTS - D DLSA 21
BK FEEDBACK GAIN MATRIX DLSA 22
BK03 BK03 DLSA 23
NXM INPUT MAXIMUM NO OF STATES DLSA 24
NRM INPUT MAXIMUM NO OF OUTPUTS DLSA 25
NUM INPUT MAXIMUM NO OF INPUTS DLSA 26
DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM) DLSA 27
DIMENSION B1(NXM,NUM),H2(NXM,NUM) DLSA 28
DIMENSION C1(NRM,NXM),C3(NRM,NXM) DLSA 29
DIMENSION D1(NRM,NUM),BK(NUM,NRM) DLSA 30
DIMENSION BK03(NUM,NXM) DLSA 31
DIMENSION HEAD(20),CARD(20) DLSA 32
COMMON/INOUT/IR,IN,IPRINT,INSERT,LOCATE,NULL,MARK(20) DLSA 33
1,JQ,JS,JSD,JF,JD DLSA 34
DATA HBFBH,HBG1R,HRG2R,HHHR/4H F ,4H G1 ,4H G2 ,4H H / DLSA 35
DATA HBDBB,HBKKG,HC,HPCAR,HREAD/4H D ,4H AKG,1HC,4H CAR,4HREAD/ DLSA 36
DATA HTAP,HEND/4H TAP,4HEND / DLSA 37
DATA HBBBR,HPEER/4H ,4HPE / DLSA 38
DATA HRAKP/4H AKP/ DLSA 39
IGAIN=0 DLSA 40
C C C READ QUADRUPLE DATA FROM QDATA FILE DLSA 41
READ(IR,20) CARD DLSA 42
IF(CARD(6).NE.HPEER) GO TO 420 DLSA 43
CALL ZERO(A,NXM,NXM) DLSA 44
CALL ZERO(B,NXM,NUM) DLSA 45
CALL ZERO(C,NRM,NXM) DLSA 46
CALL ZERO(D,NRM,NUM) DLSA 47
CALL ZERO(BK,NUM,NRM) DLSA 48
READ(IR,20) HEAD DLSA 49
20 FORMAT(20A4) DLSA 50
CALL FILE(JQ,LOCATE,HEAD) DLSA 51
READ(10,T,NX,NR,NU,((A(I,J),I=1,NX),J=1,NX), DLSA 52
1((R(I,J),I=1,NX),J=1,NU),((C(I,J),I=1,NR),J=1,NX), DLSA 53
2((D(I,J),I=1,NU),J=1,NU),NXA,NRA,NUA,NR1,NR2,NR3,NU1,NU2,NU3 DLSA 54
C C C PARTITION MATRICES B,C,D DLSA 55
DO 28 I=1,NX DLSA 56
DLSA 57
DLSA 58
DLSA 59
DLSA 60
DLSA 61
DLSA 62
DLSA 63
DLSA 64

```

Figure 107. Subroutine DL5A Program Listing

	DO 24 J=1,NU1	DLSA 65
24	R1(I,J)=B(I,J)	DLSA 66
	DO 28 J=1,NU2	DLSA 67
	JJ=NU1+J	DLSA 68
28	R2(I,J)=B(I,JJ)	DLSA 69
	DO 40 J=1,NX	DLSA 70
	DO 34 I=1,NR1	DLSA 71
34	C1(I,J)=C(I,J)	DLSA 72
	DO 40 I=1,NR3	DLSA 73
	II=NR1+NR2+I	DLSA 74
40	C3(I,J)=C(II,J)	DLSA 75
	DO 44 I=1,NR1	DLSA 76
	DO 44 J=1,NU1	DLSA 77
44	D11(I,J)=D(I,J)	DLSA 78
	IF(IPRINT.LT.6) GO TO 400	DLSA 79
	CALL MPRS(A,NXM,NXM,NX,NX,T,4HA)	DLSA 80
	CALL MPRS(B,NXM,NUM,NX,NU,T,4HB)	DLSA 81
	CALL MPRS(C,NRM,NXM,NR,NX,T,4HC)	DLSA 82
	CALL MPRS(D,NRM,NUM,NR,NU,T,4HD)	DLSA 83
	CALL MPRS(R1,NXM,NUM,NX,NU1,T,4HH1)	DLSA 84
	CALL MPRS(R2,NXM,NUM,NX,NU2,T,4HH2)	DLSA 85
	CALL MPRS(C1,NRM,NXM,NR1,NX,T,4HC1)	DLSA 86
	CALL MPRS(C3,NRM,NXM,NR3,NX,T,4HC3)	DLSA 87
	CALL MPRS(D11,NRM,NUM,NR1,NU1,T,4HD11)	DLSA 88
		DLSA 89
C	READ GAIN MATRIX DATA FROM DDATA OR FDATA FILE OR FROM INPUT DATA	DLSA 90
C		DLSA 91
400	CONTINUE	DLSA 92
	READ(IR,20) CARD	DLSA 93
	IF(CARD(1).EQ.HFND) GO TO 600	DLSA 94
	IGAIN=1	DLSA 95
	IF((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HRTAP)) GO TO 460	DLSA 96
	IF((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HRCAR)) GO TO 560	DLSA 97
420	CONTINUE	DLSA 98
	WRITE(IW,440)	DLSA 99
440	FORMAT(//,IX,24HINPUT CONTROL CARD ERROR)	DLSA 100
	STOP 111	DLSA 101
460	CONTINUE	DLSA 102
	IGAIN=1	DLSA 103
	IF(CARD(6).EQ.HRAKG) GO TO 480	DLSA 104
	IF(CARD(6).EQ.HRAKP) GO TO 480	DLSA 105
	GO TO 420	DLSA 106
480	CONTINUE	DLSA 107
	JDF=JF	DLSA 108
	IF(CARD(6).EQ.HRAKG) JDF=JD	DLSA 109
	READ(IR,20) HEAD	DLSA 110
500	CONTINUE	DLSA 111
	READ(JDF,20) CARD	DLSA 112
	DO 520 I=1,20	DLSA 113
	IF(CARD(I).NE.HEAD(I)) GO TO 500	DLSA 114
520	CONTINUE	DLSA 115
	CALL ZERO(RK,NUM,NRM)	DLSA 116
	CALL INPTM(RK,NUM,NRM,JDF)	DLSA 117
	REWIND JDF	DLSA 118
	GO TO 490	DLSA 119
560	CONTINUE	DLSA 120
	CALL ZERO(RK,NUM,NRM)	DLSA 121
	CALL INPTM(RK,NUM,NRM,IR)	DLSA 122
	GO TO 490	DLSA 123
600	CONTINUE	DLSA 124
	IF(IGAIN.EQ.0) RETURN	DLSA 125
C		DLSA 126
C	COMPUTE CLOSED LOOP QUADRUPLE DATA	DLSA 127
C		DLSA 128
	DO 70 I=1,NU1	DLSA 129
	DO 70 J=1,NX	DLSA 130

Figure 107. Subroutine DLSA Program Listing (Continued)

AKC3(I,J)=0.0	DLSA 131
DO 70 K=1,NR3	DLSA 132
70 AKC3(I,J)=AKC3(I,J)+RK(I,K)*C3(K,J)	DLSA 133
DO 80 I=1,NX	DLSA 134
DO 80 J=1,NX	DLSA 135
DO 80 K=1,NU1	DLSA 136
80 A(I,J)=A(I,J)+R1(I,K)*AKC3(K,J)	DLSA 137
DO 90 I=1,NR1	DLSA 138
DO 90 J=1,NX	DLSA 139
DO 90 K=1,NU1	DLSA 140
90 C(I,J)=C(I,J)+D11(I,K)*AKC3(K,J)	DLSA 141
DO 100 I=1,NX	DLSA 142
DO 100 J=1,NU2	DLSA 143
100 R(I,J)=H2(I,J)	DLSA 144
NU=NU2	DLSA 145
IF(IP4INT.LT.6) RETURN	DLSA 146
CALL MPRS(A,NXM,NXM,NX,NX,T,4HA)	DLSA 147
CALL MPRS(R,NXM,NUM,NX,NU,T,4HR)	DLSA 148
CALL MPRS(C,NRM,NXM,NR,NX,T,4HC)	DLSA 149
CALL MPRS(D,NRM,NUM,NR,NU,T,4HD)	DLSA 150
CALL MPRS(HK,NUM,NXM,NU1,NX,T,4HK)	DLSA 151
RETURN	DLSA 152
END	DLSA 153

Figure 107. Subroutine DLSA Program Listing (Concluded)

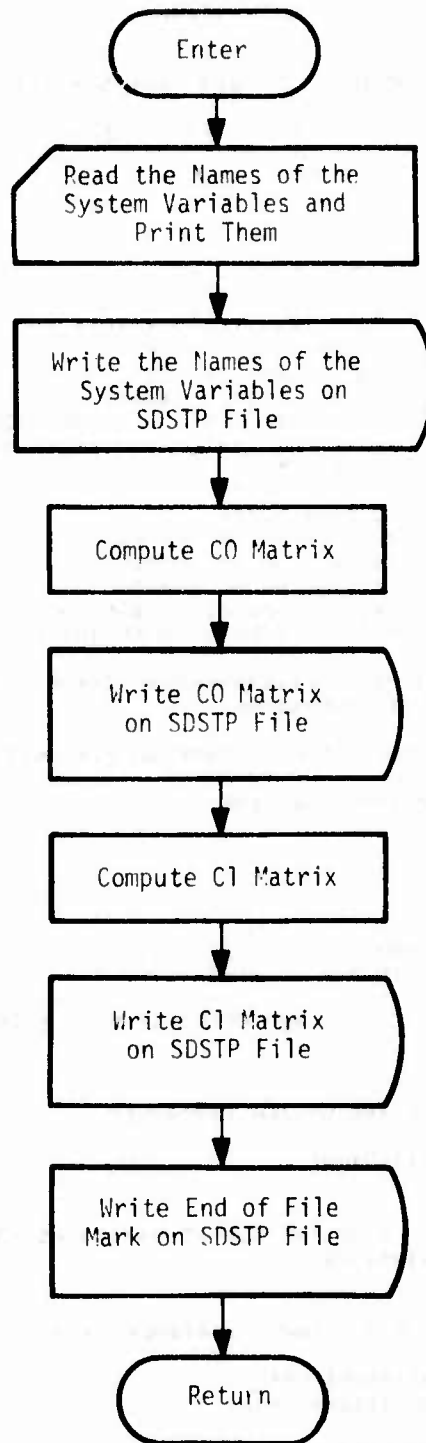


Figure 108. Subroutine FINK Flow Chart

	SUBROUTINE FINK(A,R,C,D,CC,NAME,NX,NR,NU, NXRM,NXRUM,NXRM,NXRUM)	FINK 2
C		FINK 3
C	PURPOSE - TO COMPUTE FREQUENCY DOMAIN REPRESENTATION	FINK 4
C	OF STATE SPACE QUADRUPLE DATA	FINK 5
C	ANALYSIS - A F KONGAR / J K MAHESH - THE HONEYWELL INC	FINK 6
C	DATE WRITTEN - 1975	FINK 7
C		FINK 8
C	SUBPROGRAMS CALLED	FINK 9
C	ZERO	FINK 10
C	MP-S	FINK 11
C		FINK 12
C	ARGUMENTS LIST	FINK 13
C	A INPUT STATE TRANSITION MATRIX	FINK 14
C	R INPUT CONTROL INPUT MATRIX	FINK 15
C	C INPUT STATE OUTPUT MATRIX	FINK 16
C	D INPUT CONTROL OUTPUT MATRIX	FINK 17
C	CC FOR STORING C0 AND C1 MATRICES	FINK 18
C	NAME ARRAY FOR SYSTEM VARIABLES NAMES	FINK 19
C	NX INPUT NO OF STATES	FINK 20
C	NR INPUT NO OF OUTPUTS	FINK 21
C	NU INPUT NO OF INPUTS	FINK 22
C	NXR INPUT MAXIMUM NO OF STATES	FINK 23
C	NXRUM INPUT MAXIMUM NO OF OUTPUTS	FINK 24
C	NUX INPUT MAXIMUM NO OF INPUTS	FINK 25
C	NXRUM INPUT MAXIMUM ROW DIMENSION FOR C0 AND C1	FINK 26
C	NXRUM INPUT MAXIMUM COLUMN DIMENSION FOR C0 AND C1	FINK 27
C		FINK 28
C		FINK 29
C	DIMENSION A(NX,NX),R(NX,NR),C(NR,NX),D(NR,NU)	FINK 30
C	DIMENSION CC(NXR,NXRUM),NAME(NXRUM)	FINK 31
C	DIMENSION CARD(20)	FINK 32
C	COMMON/INOUT/IR,IW,IP,INT,INSERT,LOCATE, NULL, MARK(20),	FINK 33
C	LJS,JS,JSO,JF,JD	FINK 34
C	DATA HC,HEND,HNAME/HC,4HEND,4HNAME/	FINK 35
C	NXR=NX+NR	FINK 36
C	NXRUM=NXR+NU	FINK 37
C	120 CONTINUE	FINK 38
C	READ(IR,140)CARD	FINK 39
C	140 FORMAT(20A4)	FINK 40
C	IF(CARD(1).EQ.HEND)RETURN	FINK 41
C	IF(CARD(1).EQ.HNAME)GO TO 200	FINK 42
C	WRITE(IW,180)	FINK 43
C	180 FORMAT(//,1X,37HDATA CONTROL CARD SPECIFICATION ERROR)	FINK 44
C	STOP 111	FINK 45
C	200 CONTINUE	FINK 46
C		FINK 47
C	READ AND WRITE NAMES OF THE SYSTEM VARIABLES	FINK 48
C		FINK 49
C	READ(IR,370)(NAME(I),I=1,NXRUM)	FINK 50
C	370 FORMAT(A10)	FINK 51
C	WRITE(IW,375)	FINK 52
C	375 FORMAT(1H),//,1X,20HNAMES OF THE OUTPUT VARIABLES,//)	FINK 53
C	WRITE(IW,380)(NAME(I),I=1,NXR)	FINK 54
C	380 FORMAT(1X,A10)	FINK 55
C	WRITE(IW,385)	FINK 56
C	385 FORMAT(//,1X,20HNAMES OF THE INPUT VARIABLES,//)	FINK 57
C	NXRPI=NXR+1	FINK 58
C	WRITE(IW,380)(NAME(I),I=NXRPI,NXRUM)	FINK 59
C	WRITE(JSO)NXRUM,NU,(NAME(I),I=1,NXRUM)	FINK 60
C		FINK 61
C	COMPUTE C0 AND WRITE ON SDSF FILE	FINK 62
C		FINK 63
C	CALL ZERO(CC,NXRUM,NXRUM)	FINK 64

Figure 109. Subroutine FINK Program Listing

	DO 30 I=1,NX	FINK 65
	DO 28 J=1,NX	FINK 66
280	CC(I,I)=-A(I,J)	FINK 67
	DO 30 J=1,NX	FINK 68
	JJ=NX+J	FINK 69
300	CC(I,J)=H(I,J)	FINK 70
	DO 34 I=1,NR	FINK 71
	II=NX+I	FINK 72
	DO 32 J=1,NX	FINK 73
320	CC(II,J)=-C(I,J)	FINK 74
	DO 34 J=1,NX	FINK 75
	JJ=NX+J	FINK 76
340	CC(II,JJ)=D(I,J)	FINK 77
	DO 36 I=1,NR	FINK 78
	II=NX+I	FINK 79
360	CC(II,II)=1.0	FINK 80
	CALL XPHS(CC,NXRM,NXRUM,NXR,NXRU,T,4HC0)	FINK 81
	WRITE(JSD)((CC(I,J),J=1,NXR),I=1,NXR)	FINK 82
C		FINK 83
C	COMPUTE C1 AND WRITE ON SDSTP FILE	FINK 84
C		FINK 85
	CALL XPHS(CC,NXRM,NXRUM)	FINK 86
	DO 26 I=1,NX	FINK 87
260	CC(I,I)=1.0	FINK 88
	CALL XPHS(CC,NXRM,NXRUM,NXR,NXRU,T,4HC1)	FINK 89
	WRITE(JSD)((CC(I,J),J=1,NXR),I=1,NXR)	FINK 90
C		FINK 91
C	WRITE AN END OF FILE MARK ON SDSTP	FINK 92
C		FINK 93
	ENDFILE JSD	FINK 94
	GO TO 120	FINK 95
	END	FINK 96

Figure 109. Subroutine FINK Program Listing (Concluded)

	SUBROUTINE MP(K,L,I,J;A)		
	DIMENSION A(K,L)		MP 2
	DO 1 II=1,I		MP 3
	WRITE(9,5)II		MP 4
	5 FORMAT(5H ROW 13)		MP 5
	1 WRITE(9,2)(A(II,JJ),JJ=1,J)		MP 6
2	FORMAT(2X,10E12.4)		MP 7
	RETURN		MP 8
	END		MP 9
			10

Figure 110. Subroutine MP Program Listing

SUBROUTINE OUTP(I, J, JJ, Y, I=)	OUTP	2
DIMENSION Y(I, J), YD(5), ID(5), JD(5)	OUTP	3
50 FORMAT(5(2I2,F12.5))	OUTP	4
III=0	OUTP	5
DO 10 K=1, JJ	OUTP	6
DO 10 M=1, JJ	OUTP	7
IF (Y(K, M).EQ.0.) GOTO 100	OUTP	8
III=III+1	OUTP	9
YD(III)=Y(K, M)	OUTP	10
ID(III)=K	OUTP	11
JD(III)=M	OUTP	12
IF (III.LT.5) GOTO 100	OUTP	13
WRITE (IR, 50) (ID(L), JD(L), YD(L), L=1, III)	OUTP	14
III=0	OUTP	15
100 CONTINUE	OUTP	16
IF (III.EQ.4) RETURN	OUTP	17
WRITE (IR, 50) (ID(L), JD(L), YD(L), L=1, III)	OUTP	18
RETURN	OUTP	19
END	OUTP	20

Figure 111. Subroutine OUTP Program Listing

SUBROUTINE POLES(NX,A,MX,RP,M)	POLES 2
DIMENSION A(MX,1),RR(1)	POLES 3
CALL HESSEN(NX,A,MX)	POLES 4
CALL ORCALL(MX,A,RP,M,NX)	POLES 5
WRITE(9,6087)	POLES 6
6087 FORMAT(1H1/7X,1)HEIGENVALUES/12X,4HREAL,9X,9HIMAGINARY,8X,13HDAMP/	POLES 7
1NG RAT10.5X,9HFREQUENCY//)	POLES 8
MM=M/2	POLES 9
DO 6083 K=1,MM	POLES 10
I=2*K-1	POLES 11
OMEGA=SQRT(RR(I)*RR(I)+RR(I+1)*RR(I+1))	POLES 12
IF(ABS(RR(I+1)).GT..0000001) GO TO 1	POLES 13
WRITE(9,6084) RR(I)	POLES 14
GO TO 6083	POLES 15
1 DELTA=RR(I)/OMEGA	POLES 16
WRITE(9,6084) RR(I),RR(I+1),DELTA,OMEGA	POLES 17
6083 CONTINUE	POLES 18
6084 FORMAT(8X,4F15,8)	POLES 19
RETURN	POLES 20
END	POLES 21

Figure 112. Subroutine POLES Program Listing


```

SUBROUTINE HESSEN(N,A,D)
DIMENSION A(1)
INTEGER P,PM,PX,D
ID=0+1
NN=(N-1)*ID+1
KX=NN-ID-ID+1
PX=1
PX=N
DO 70 K=2,KX,10
NK=PX
PM=PM*D
PX=PX*D
JP=PM
T=0.
R=0.
J=K
JC=JP
JK=J
30 T=ABS(A(J))
IF(T.LE.H) GO TO 35
JC=JP
JK=J
R=T
35 IF(J.GE.NK) GO TO 37
J=J+1
JP=JP*D
GO TO 30
37 IF(JK.EQ.K) GO TO 44
J=JC
DO 38 P=PM,PX
T=A(P)
A(P)=A(J)
A(J)=T
38 J=J+1
P=JK
DO 39 J=K,NN,D
T=A(J)
A(J)=A(P)
A(P)=T
39 P=P*D
44 IF(A(K).EQ.0.) GO TO 70
JC=PM*D
JK=K+1
T=1./A(K)
45 B=A(J+1)
IF(B.EQ.0.) GO TO 65
R=B*T
KM=K+1
JM=JK*D
AJM=A(JM)-R*A(KM)
50 IF(ABS(AJM).LE.(.1E-9*ABS(A(JM)))) AJM=0.
A(JM)=AJM
KM=KM*D
JM=JM*D
IF(JM.LE.NN) GO TO 50
J=JC
DO 60 P=PM,PX
AP=A(P)+B*A(J)
IF(ABS(AP).LE.(.1E-9*ABS(A(P)))) AP=0.
60 A(P)=AP
J=J+1
65 JK=JK+1
JC=JC*D

```

```

HESSEN 2
HESSEN 3
HESSEN 4
HESSEN 5
HESSEN 6
HESSEN 7
HESSEN 8
HESSEN 9
HESSEN10
HESSEN11
HESSEN12
HESSEN13
HESSEN14
HESSEN15
HESSEN16
HESSEN17
HESSEN18
HESSEN19
HESSEN20
HESSEN21
HESSEN22
HESSEN23
HESSEN24
HESSEN25
HESSEN26
HESSEN27
HESSEN28
HESSEN29
HESSEN30
HESSEN31
HESSEN32
HESSEN33
HESSEN34
HESSEN35
HESSEN36
HESSEN37
HESSEN38
HESSEN39
HESSEN40
HESSEN41
HESSEN42
HESSEN43
HESSEN44
HESSEN45
HESSEN46
HESSEN47
HESSEN48
HESSEN49
HESSEN50
HESSEN51
HESSEN52
HESSEN53
HESSEN54
HESSEN55
HESSEN56
HESSEN57
HESSEN58
HESSEN59
HESSEN60
HESSEN61
HESSEN62
HESSEN63
HESSEN64

```

Figure 113. Subroutine HESSEN Program Listing

```
70 IF(IJK,LE,NKI) GO TO 45  
CONTINUE  
RETURN  
END
```

```
HESSEN65  
HESSEN66  
HESSEN67  
HESSEN68
```

Figure 113. Subroutine HESSEN Program Listing (Concluded)

	SUBROUTINE QRCALL (D,A,P,M,N)	QRCALL 2
	INTEGER D	QRCALL 3
	DIMENSION A(D,1),R(1)	QRCALL 4
	N = NIN	QRCALL 5
	ANN = 1.	QRCALL 6
	ACT = .1E-7	QRCALL 7
	ITER = 0	QRCALL 8
	M = 0	QRCALL 9
	IF (N.LE.1) RETURN	QRCALL 10
	IF (N.FO.2) GO TO 25	QRCALL 11
15	DELTA=ACT*ABS(A(N,N))	QRCALL 12
	ACC = ABS(A(N,N-1))	QRCALL 13
	IF (ACC.EQ.0.) GO TO 16	QRCALL 14
	IF (ACC.GT.DELTA) GO TO 25	QRCALL 15
	IF (ITER.GT.25) GO TO 16	QRCALL 16
	IF (ANN.GT.ACT) GO TO 25	QRCALL 17
16	M = M+2	QRCALL 18
	R(M-1) = A(N,N)	QRCALL 19
	R(M) = 0.	QRCALL 20
17	K = NIN-N+1	QRCALL 21
	ITER = 0	QRCALL 22
	N = N-1	QRCALL 23
20	IF (N.GT.2) GO TO 15	QRCALL 24
	IF (N.FO.2) GO TO 25	QRCALL 25
	IF (N.FO.1) GO TO 16	QRCALL 26
	R(M+1)=ACT	QRCALL 27
	RETURN	QRCALL 28
25	B = .5*(A(N-1,N-1)+A(N,N))	QRCALL 29
	DAN=ABS(A(N,N)-A(N-1,N-1))	QRCALL 30
	SAN=ABS(A(N,N))+ABS(A(N-1,N-1))	QRCALL 31
	IF (DAN.LE.ACT*SAN) DAN=0.	QRCALL 32
	DAN=DAN*DAN*.25	QRCALL 33
	C=A(N,N-1)*A(N-1,N)	QRCALL 34
	T=DAN+C	QRCALL 35
	IF ((C.LT.0.) .AND. (ABS(T).LE.ACT*DAN)) T=0.	QRCALL 36
	IF (ABS(T).LE.ACT) T=0.	QRCALL 37
	C = SQRT(ABS(T))	QRCALL 38
	IF (N.NE.2) GO TO 50	QRCALL 39
26	IF (T.GE.0.) GO TO 30	QRCALL 40
	M = M+2	QRCALL 41
	R(M-1) = B	QRCALL 42
	R(M) = C	QRCALL 43
27	N = N-1	QRCALL 44
	GO TO 17	QRCALL 45
30	M = M+2	QRCALL 46
	R(M-1) = B+C	QRCALL 47
	R(M) = 0.	QRCALL 48
	K = NIN-N+1	QRCALL 49
	M = M+2	QRCALL 50
	R(M-1) = B-C	QRCALL 51
	R(M) = 0.	QRCALL 52
	GO TO 27	QRCALL 53
50	IF (T.GE.0.) GO TO 60	QRCALL 54
	R(M+5) = B	QRCALL 55
	R(M+6) = C	QRCALL 56
	R(M+7) = B	QRCALL 57
	R(M+8) = -C	QRCALL 58
	GO TO 70	QRCALL 59
60	X = B+C	QRCALL 60
	Y = B-C	QRCALL 61
	R(M+6) = 0.	QRCALL 62
	R(M+8) = 0.	QRCALL 63
	R(M+5) = X	QRCALL 64

Figure 114. Subroutine QRCALL Program Listing

	R(M*7) = Y	QRCALL65
	IF (ABS(X).GT.ARS(Y)) GO TO 70	QRCALL66
	R(M*5) = Y	QRCALL67
	R(M*7) = X	QRCALL68
70	IF (ITP.LE.0) GO TO 130	QRCALL69
	X = ARS(R(M*5)-R(M*1))*ABS(R(M*6)-R(M*2))	QRCALL70
	ACC = ABS(R(M*5))*ABS(R(M*1))*ABS(R(M*6))*ABS(R(M*2))	QRCALL71
	IF (ACC.GT.1.) X=X/ACC	QRCALL72
	Y = ARS(R(M*7)-R(M*3))*ABS(R(M*8)-R(M*4))	QRCALL73
	ACC = ABS(R(M*7))*ABS(R(M*3))*ABS(R(M*8))*ABS(R(M*4))	QRCALL74
	IF (ACC.GT.1.) Y=Y/ACC	QRCALL75
	ACC = ABS(A(N-1,N-2))	QRCALL76
	DELTA=AMAX1(DELTA,(ACT*ARS(A(N-1,N-1))))	QRCALL77
	IF (ACC.GT.DELTA) GO TO 80	QRCALL78
	IF (ITER.GT.25) GO TO 26	QRCALL79
	IF ((X.LE.ACT).AND.(Y.LE.ACT)) GO TO 26	QRCALL80
80	IF (ITER.GT.200) GO TO 200	QRCALL81
	IF ((X.GT..5).AND.(Y.GT..5)) GO TO 130	QRCALL82
	K = M*5	QRCALL83
	IF (Y.GT..5) GO TO 120	QRCALL84
	IF (X.GT..5) GO TO 110	QRCALL85
	RHO = R(M*5)*R(M*7)-R(M*6)*R(M*8)	QRCALL86
	SIGMA = R(M*5)*R(M*7)	QRCALL87
100	CONTINUE	QRCALL88
	ANN = A(N,N)	QRCALL89
	CALL QRIN(A,RHO,SIGMA,D,DELTA)	QRCALL90
	B = ARS(A(N,N))	QRCALL91
	ANN = ABS(ANN-A(N,N))	QRCALL92
	IF (B.GT.ACT) ANN = ANN/B	QRCALL93
	ITER = ITER+1	QRCALL94
	DO 105 I=1,4	QRCALL95
	K = M*I	QRCALL96
105	R(K) = R(K*4)	QRCALL97
	GO TO 15	QRCALL98
110	K = M*7	QRCALL99
120	RHO = R(K)*R(K)	QRCALL100
	SIGMA = R(K)*R(K)	QRCALL101
	GO TO 100	QRCALL102
130	RHO = 0.	QRCALL103
	SIGMA=0.	QRCALL104
	GO TO 100	QRCALL105
200	CONTINUE	QRCALL106
	WRITE(9,700)	QRCALL107
700	FORMAT(1H1,25HALL FIGENVALUES NOT FOUND)	QRCALL108
	RETURN	QRCALL109
	END	QRCALL110

Figure 114. Subroutine QRCALL Program Listing (Concluded)

	SUBROUTINE QR(N,A,RHO,SIGMA,D,DELTA)	QR	2
	DIMENSION A(1)	QR	3
	REAL KAPPA	QR	4
	INTEGER P,Q,D	QR	5
	EQUIVALENCE (P,Q)	QR	6
	ID = 0+1	QR	7
	N0 = ID*(N-1)+1	QR	8
	N1 = N0-ID	QR	9
	N2 = N1-ID	QR	10
	N3 = N2-ID	QR	11
	IF(N.GT.3) GO TO 5	QR	12
	IF(N.LE.2) RETURN	QR	13
2	Q = 1	QR	14
	GO TO 35	QR	15
5	I = N3+1	QR	16
7	IF(ABS(A(I)).LT.DELTA) GO TO 10	QR	17
	IF(I.LE.2) GO TO 2	QR	18
	I = I-ID	QR	19
	GO TO 7	QR	20
10	Q = I+D	QR	21
	A(I) = 0.	QR	22
35	I = P	QR	23
	I0 = 0	QR	24
	I0 = I-D	QR	25
	I1 = I+D	QR	26
	I2 = I1+D	QR	27
	G1 = A(I)*(A(I)-SIGMA)+A(I1)*A(I+1)+RHO	QR	28
	G2 = A(I+1)*(A(I)+A(I+1)-SIGMA)	QR	29
	G3 = A(I+1)*A(I+2)	QR	30
	A(I+2) = 0.	QR	31
	GO TO 45	QR	32
40	G1 = A(I0)	QR	33
	G2 = A(I0+1)	QR	34
	G3 = 0.	QR	35
	I0 = I0+D	QR	36
	IF(I.LE.N2) G3 = A(I0+2)	QR	37
45	KAPPA = SQRT(G1*G1+G2*G2+G3*G3)	QR	38
	IF(G1.LT.0.) KAPPA = -KAPPA	QR	39
	IF(KAPPA.NE.0.) GO TO 47	QR	40
	ALPHA = 2.	QR	41
	P1 = 0.	QR	42
	P2 = 0.	QR	43
	GO TO 48	QR	44
47	ALPHA = 1.+G1/KAPPA	QR	45
	P1 = 1./(G1+KAPPA)	QR	46
	P2 = P1*G3	QR	47
	P1 = P1*G2	QR	48
48	IF(I.EQ.Q) GO TO 49	QR	49
	A(I0) = -A(I0)	QR	50
	IF(I.NE.P) A(I0) = -KAPPA	QR	51
49	J = I-D	QR	52
50	J = J+D	QR	53
	IF(J.GE.N3) GO TO 51	QR	54
	ETA = A(J)+P1*A(J+1)	QR	55
	IF(I.LE.N2) ETA = ETA+P2*A(J+2)	QR	56
	ETA = ALPHA*ETA	QR	57
	A(J) = A(J)-ETA	QR	58
	A(J+1) = A(J+1)-P1*ETA	QR	59
	IF(I.LE.N2) A(J+2) = A(J+2)-P2*ETA	QR	60
	GO TO 50	QR	61
51	J = I-1	QR	62
	JINX = MIN0(I+2,N1+1)	QR	63
60	J = J+1	QR	64

Figure 115. Subroutine QR Program Listing

K = J+D	QR	65
ETA = A(J)+P1*A(K)	QR	66
L = K+D	QR	67
IF(I.LE.N2) ETA = CTA+P2*A(L)	QR	68
ETA = ETA*ALPHA	QR	69
A(J) = A(J)-ETA	QR	70
A(K) = A(K)-P1*ETA	QR	71
IF(I.LE.N2) A(L) = A(L)-P2*ETA	QR	72
IF(J.LT.JINX) GO TO 67	QR	73
IF(I.GT.N3) GO TO 65	QR	74
ETA = ALPHA*P2*A(I2+3)	QR	75
A(I+3) = -ETA	QR	76
A(I1+3) = -P1*ETA	QR	77
A(I2+3) = A(I2+3)-D2*ETA	QR	78
65 IF(I.GE.N1) RETURN	QR	79
I0 = I + 1	QR	80
I = I1+1	QR	81
I1 = I2+1	QR	82
I2 = I2+I0	QR	83
GO TO 4.	QR	84
END	QR	85

Figure 115. Subroutine QR Program Listing (Concluded)

	SUBROUTINE INPTM(A,II,JJ,IR)	INPTM 2
C		INPTM 3
C	PURPOSE - TO READ NONZERO ELEMENTS OF A MATRIX FROM FILE IN	INPTM 4
C	ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC	INPTM 5
C	DATE WRITTEN - 1975	INPTM 6
C		INPTM 7
C	ARGUMENTS LIST	INPTM 8
C	A INPUT MATRIX DATA	INPTM 9
C	II INPUT MAXIMUM NO OF ROWS	INPTM 10
C	JJ INPUT MAXIMUM NO OF COLUMNS	INPTM 11
C	IR INPUT FILE NO FOR READING MATRIX DATA	INPTM 12
C		INPTM 13
	DIMENSION A(II,JJ),ID(5),JD(5),YD(5)	INPTM 14
	2 FORMAT (5(2I2,F12.5))	INPTM 15
	1 READ(IR,2)(ID(L),JD(L),YD(L),L=1,5)	INPTM 16
	IF (EOF(IR))10,6	INPTM 17
6	CONTINUE	INPTM 18
	IF (ID(1))3,10,3	INPTM 19
3	DO 5 I=1,5	INPTM 20
	IF (ID(L))4,1,4	INPTM 21
4	I=ID(L)	INPTM 22
	J=JD(L)	INPTM 23
5	A(I,J)=YD(L)	INPTM 24
	GO TO 1	INPTM 25
10	CONTINUE	INPTM 26
	RETURN	INPTM 27
	END	INPTM 28

Figure 116. Subroutine INPTM Program Listing

C	SUBROUTINE WTP(A,NR,NC,NR4,NC4,JW)	WTP	2
C		WTP	3
C	PURPOSE - TO WRITE NONZERO ELEMENTS OF A MATRIX ON A FILE	WTP	4
C	ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC	WTP	5
C	DATE WRITTEN - 1975	WTP	6
C		WTP	7
C	ARGUMENTS LIST	WTP	8
C	A INPUT MATRIX DATA	WTP	9
C	NR INPUT NO OF ROWS	WTP	10
C	NC INPUT NO OF COLUMNS	WTP	11
C	NR4 INPUT MAXIMUM NO OF ROWS	WTP	12
C	NC4 INPUT MAXIMUM NO OF COLUMNS	WTP	13
C	JW INPUT FILE NO FOR WRITING DATA	WTP	14
C		WTP	15
C	DIMENSION A(NR4,NC4),RCARD(20)	WTP	16
C	DIMENSION AD(5),ID(5),JD(5)	WTP	17
C	INTEGER RCARD	WTP	18
C	IF(NR.EQ.0)GO TO 100	WTP	19
C	IF(NC.EQ.0)GO TO 100	WTP	20
C	III=0	WTP	21
C	DO 80 K=1,NR	WTP	22
C	DO 80 M=1,NC	WTP	23
C	IF(A(K,M).EQ.0.)GO TO 80	WTP	24
C	III=III+1	WTP	25
C	AD(III)=A(K,M)	WTP	26
C	ID(III)=K	WTP	27
C	JD(III)=M	WTP	28
C	IF(III.LT.5)GO TO 80	WTP	29
C	WRITE(JW,60)(ID(L),JD(L),AD(L),L=1,III)	WTP	30
C	60 FORMAT(5(2I2,F12.5))	WTP	31
C	III=0	WTP	32
C	80 CONTINUE	WTP	33
C	IF(III.EQ.0)GO TO 100	WTP	34
C	WRITE(JW,60)(ID(L),JD(L),AD(L),L=1,III)	WTP	35
C	100 CONTINUE	WTP	36
C	IRLANK=4H	WTP	37
C	DO 110 I=1,2	WTP	38
C	110 RCARD(I)=IRLANK	WTP	39
C	WRITE(JW,120)(RCARD(I),I=1,2)	WTP	40
C	120 FORMAT(2,A4)	WTP	41
C	RETURN	WTP	42
C	END	WTP	43

Figure 117. Subroutine WTP Program Listing

FUNCTION GRAN(N)	GRAN	2
X=N	GRAN	3
IF (N .EQ. 0) GO TO 1	GRAN	4
ISEED = 31973679892	GRAN	5
X=ISEED	GRAN	6
TEM=RANF(X)	GRAN	7
X=0.	GRAN	8
1 TEM = 0.0	GRAN	9
DO 2 I = 1,12	GRAN	10
2 TEM=TEM+RANF(X)	GRAN	11
TEM = TEM - 6.0	GRAN	12
GRAN = TEM	GRAN	13
RETURN	GRAN	14
END	GRAN	15

Figure 118. Subroutine GRAN Program Listing

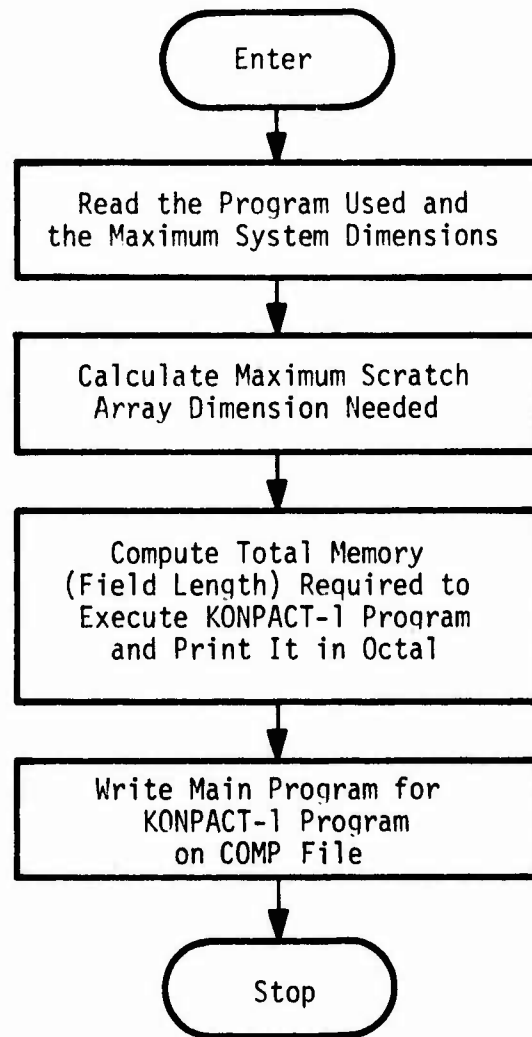


Figure 119. Program PRECOM Flow Chart

APPENDIX

PRECOMPILER PROGRAM FOR KONPACT-1

The precompiler program performs the task of writing the MAIN program for KONPACT-1. A brief description of the precompiler program is presented in this section.

The precompiler program reads the system dimensions and the KONPACT-1 program names and computes the maximum sizes of the scratch arrays. It writes the MAIN program for KONPACT-1 on file COMPIL. The flow chart is given in Figure 119 and the program listing is given in Figure 120.

	PROGRAM PRECOM(INPUT,OUTPUT,COMP,TAPE5=INPUT,TAPE9=OUTPUT	PRECOM 2
	1,TAPE6=COMP)	PRECOM 3
C		PRECOM 4
C	ANALYSIS - A F KONAR / J R MAHESH - THE HONEYWELL INC	PRECOM 5
C	PURPOSE - TO READ THE PROGRAMS USED AND THE MAXIMUM SYSTEM	PRECOM 6
C	DIMENSIONS AND SET UP THE MAIN PROGRAM FOR KONPACT=1 PROGRAMS	PRECOM 7
C	DATE WRITTEN - DECEMBER 1975	PRECOM 8
C		PRECOM 9
	DIMENSION CARD(20)	PRECOM10
	DATA MNAME,MNME,MNUME,MNYME/4MNXM,4MNRM,4MNU,4MNYM/	PRECOM11
	DATA MMSHE,MMTRE,MCH,MKABB/4MMSH,4MTH,2MC,4MK /	PRECOM12
	DATA MK1H,MK2RB,MK3RB,MK4RB/4MK1,4MK2,4MK3,4MK4 /	PRECOM13
	MS1F=0 \$ MS2F=0 \$ MS3F=0 \$ MS4F=0 \$ MS5F=0	PRECOM14
C		PRECOM15
C	INITIALIZE MAXIMUM SYSTEM DIMENSIONS	PRECOM16
C		PRECOM17
	NXM=0 \$ NHM=0 \$ NUM=0 \$ NYM=0 \$ MSB=0 \$ MTR=0	PRECOM18
C		PRECOM19
C	READ THE PROGRAMS USED AND THE MAXIMUM SYSTEM DIMENSIONS	PRECOM20
C		PRECOM21
	100 CONTINUE	PRECOM22
	READ(5,120)CARD	PRECOM23
	120 FORMAT(20A4)	PRECOM24
	IF(EOF(5))220,140	PRECOM25
	140 CONTINUE	PRECOM26
	DECODE(4,160,CARD(1))CC,DUMMY	PRECOM27
	160 FORMAT(A2,A2)	PRECOM28
	IF(CC.EQ.MCH)GO TO 100	PRECOM29
C		PRECOM30
C	SET THE PROGRAM FLAGS	PRECOM31
C		PRECOM32
	CODE=CARD(2)	PRECOM33
	IF(CODE.EQ.MK1HR)MS1F=1	PRECOM34
	IF(CODE.EQ.MK1HR)GO TO 100	PRECOM35
	IF(CODE.EQ.MK2RB)MS2F=1	PRECOM36
	IF(CODE.EQ.MK2RB)GO TO 100	PRECOM37
	IF(CODE.EQ.MK3RB)MS3F=1	PRECOM38
	IF(CODE.EQ.MK3RB)GO TO 100	PRECOM39
	IF(CODE.EQ.MK4RB)MS4F=1	PRECOM40
	IF(CODE.EQ.MK4RB)GO TO 100	PRECOM41
	IF(CODE.EQ.MK4RB)MS5F=1	PRECOM42
	IF(CODE.EQ.MK4RB)GO TO 100	PRECOM43
C		PRECOM44
C	SET THE MAXIMUM SYSTEM DIMENSIONS	PRECOM45
C		PRECOM46
	CODE=CARD(1)	PRECOM47
	DECODE(4,180,CARD(2))MAX,DUMMY	PRECOM48
	180 FORMAT(I3,A)	PRECOM49
	IF(CODE.EQ.MNAME)NXM=MAX	PRECOM50
	IF(CODE.EQ.MNAME)GO TO 100	PRECOM51
	IF(CODE.EQ.MNME)NHM=MAX	PRECOM52
	IF(CODE.EQ.MNME)GO TO 100	PRECOM53
	IF(CODE.EQ.MNUME)NUM=MAX	PRECOM54
	IF(CODE.EQ.MNUME)GO TO 100	PRECOM55
	IF(CODE.EQ.MNYME)NYM=MAX	PRECOM56
	IF(CODE.EQ.MNYME)GO TO 100	PRECOM57
	IF(CODE.EQ.MMSHE)MSH=MAX	PRECOM58
	IF(CODE.EQ.MMSHE)GO TO 100	PRECOM59
	IF(CODE.EQ.MMTRE)MTR=MAX	PRECOM60
	IF(CODE.EQ.MMTRE)GO TO 100	PRECOM61
C		PRECOM62
C	IF DATA CARD IS IN ERROR PRINT ERROR MESSAGE	PRECOM63
C		PRECOM64

Figure 120. Program PRECOM Program Listing

```

WRITE(9,200)CARD
200 FORMAT(1H1,/,1X,25HEPROG IN PRECOMPILER DATA,/,1X
1,19HLAST CARD READ *AS,/,1X,20A4)
STOP 111
C
C CALCULATE DIMENSIONS WHICH ARE USEFUL TO COMPUTE
C MAXIMUM SCRATCH ARRAY DIMENSIONS REQUIRED
C
220 CONTINUE
NXRM=NXM*NRM
NXUM=NXM*NUM
NYUM=NYM*NUM
NRUM=NRM*NUM
NXRUM=NXM*NRUM
NXRYM=NXM*NYM
M=ORD=17 & NRSM=1
NDM1=MAX0(M=ORD,NXM,NRM,NRSM)
NUM12=MAX0(NXUM,NRM)
NDM21=MAX0(NRM,NXM,NRSM)
NDM22=MAX0(NAM,NUM)
MM=MAX0(NUM,NRM)
C
C CALCULATE MAXIMUM DIMENSIONS FOR SCRATCH ARRAY S1
C TO USE THE VARIOUS KONPACT=1 PROGRAMS
C
MS111=1+3*NXM+2*(NYUM)+NXRYM+(2*NXM*NYUM)+NRM
MS112=1+MSH*(14*NXRUM+3)
MS11=MAX0(MS111,MS112)
MS121=MS111+MTB*15
MS122=MS112
MS12=MAX0(MS121,MS122)
MS131=MS111+MSH*(3*NARUM+NXM)+NRM*(MSB+1)
MS132=MS112
MS13=MAX0(MS131,MS132)
MS14=MS11
MS15=1+NXM*(NRUM+NRM)+NRM*(2*NRM+3*NUM)+3*NARUM
1+NDM11+NDM12+NDM21+NDM22+NUM
C
C CALCULATE MAXIMUM DIMENSIONS FOR SCRATCH ARRAY S2
C TO USE THE VARIOUS KONPACT=1 PROGRAMS
C
MS211=1+NXRM+NXUM
MS212=10000
MS21=MAX0(MS211,MS212)
MS221=MS211
MS222=13*NMM+NUM+MTB*(48+MTB+NRUM)
MS22=MAX0(MS221,MS222)
MS231=MS211
MS232=1+NRM+NUM+MSB*MM*(MSB+MM+NRUM)+MSB*NXRM+NXUM
MS23=MAX0(MS231,MS232)
MS24=MS211
MS25=1+(NXRM+NRM)*NXUM
C
C CALCULATE MAXIMUM DIMENSIONS FOR SCRATCH ARRAY S3
C TO USE THE VARIOUS KONPACT=1 PROGRAMS
C
MS31=1+17*NXUM
MS32=MS31
MS331=MS31
MS332=1+4*MSB+MM*(2*MM+NRUM)+MSB
MS33=MAX0(MS331,MS332)
MS34=MS31
MS35=2*MS31
C
C IF NO SPECIFIC PROGRAMS ARE READ SET ALL PROGRAM FLAGS TO 1
C
C

```

```

PRECOM65
PRECOM66
PRECOM67
PRECOM68
PRECOM69
PRECOM70
PRECOM71
PRECOM72
PRECOM73
PRECOM74
PRECOM75
PRECOM76
PRECOM77
PRECOM78
PRECOM79
PRECOM80
PRECOM81
PRECOM82
PRECOM83
PRECOM84
PRECOM85
PRECOM86
PRECOM87
PRECOM88
PRECOM89
PRECOM90
PRECOM91
PRECOM92
PRECOM93
PRECOM94
PRECOM95
PRECOM96
PRECOM97
PRECOM98
PRECOM99
PRECOM100
PRECOM101
PRECOM102
PRECOM103
PRECOM104
PRECOM105
PRECOM106
PRECOM107
PRECOM108
PRECOM109
PRECOM110
PRECOM111
PRECOM112
PRECOM113
PRECOM114
PRECOM115
PRECOM116
PRECOM117
PRECOM118
PRECOM119
PRECOM120
PRECOM121
PRECOM122
PRECOM123
PRECOM124
PRECOM125
PRECOM126
PRECOM127
PRECOM128
PRECOM129
PRECOM130

```

Figure 120. Program PRECOM Program Listing (Continued)

```

IF((MS1F.NE.0).OR.(MS2F.NE.0).OR.(MS3F.NE.0).OR.(MS4F.NE.0).OR.
1(MS5F.NE.0))GO TO 221
MS1F=1 $ MS2F=1 $ MS3F=1 $ MS4F=1 $ MS5F=1
C
C CALCULATE MAXIMUM SCRATCH ARRAY DIMENSIONS NEEDED
C
221 CONTINUE
IF(MS1F.EQ.1)GO TO 222
MS11=1 $ MS21=1 $ MS31=1
222 CONTINUE
IF(MS2F.EQ.1)GO TO 224
MS12=1 $ MS22=1 $ MS32=1
224 CONTINUE
IF(MS3F.EQ.1)GO TO 226
MS13=1 $ MS23=1 $ MS33=1
226 CONTINUE
IF(MS4F.EQ.1)GO TO 228
MS14=1 $ MS24=1 $ MS34=1
228 CONTINUE
IF(MS5F.EQ.1)GO TO 230
MS15=1 $ MS25=1 $ MS35=1
230 CONTINUE
MS1=MAX0(MS11,MS12,MS13,MS14,MS15)
MS2=MAX0(MS21,MS22,MS23,MS24,MS25)
MS3=MAX0(MS31,MS32,MS33,MS34,MS35)
MS4=1
C
C COMPUTE MEMORY REQUIRED FOR SCRATCH ARRAYS
C
MST=MS1+MS2+MS3+MS4
C
C SET THE MEMORY REQUIRED FOR THE PROGRAM CODE
C
MPT=30000
C
C COMPUTE TOTAL MEMORY REQUIRED TO EXECUTE KONPACT=1 PROGRAM AND
C PRINT THE FIELD LENGTH REQUIRED IN OCTAL BASE
C
MT=MST+MPT
WRITE(9,240)
240 FORMAT(////)
WRITE(9,240)
260 FORMAT(/,10X,56(1H*))
WRITE(9,280)MT
280 FORMAT(/,10X,50HFIELD LENGTH REQUIRED FOR EXECUTING KONPACT=1 =
1,0A)
WRITE(9,240)
C
C WRITE MAIN PROGRAM FOR KONPACT=1 PROGRAM ON COMP FILE
C
WRITE(6,300)
300 FORMAT(38H PROGRAM MAIN(BINPUT,INPUT,NDATA,
1,40HNDATA,OUTPUT,TAPES=8INPUT,
2,/,36H 1TAPE6=INPUT,TAPE7=NDATA,TAPE8=
3,40HNDATA,TAPE9=OUTPUT,VDATA,
4,/,38H 2TAPE4=VDATA,SCRATCH,TAPE3=SCRATCH))
WRITE(6,320)
320 FORMAT(39HC ANALYSIS - A F KONAR / J K MAHESH
1,40H - THE HONEYWELL INC
2,/,44HC PURPOSE - TO SET UP MAXIMUM DIMENSIONS)
WRITE(6,330)
330 FORMAT(36H COMMON /DIM/ MS1,MS2,MS3,MS4
1,26H,MAXM,MAXN,NXM,NRM,NUM,NVM,/,
2,42H 1,MM,MP,MQ,MH,MSB,NR,MS,MN,MTR,MST,MT)
WRITE(6,340)MS1,MS2,MS3,MS4
340 FORMAT(22M COMMON /SC1/ S1(,15,1H),/,

```

```

PREC0131
PREC0132
PREC0133
PREC0134
PREC0135
PREC0136
PREC0137
PREC0138
PREC0139
PREC0140
PREC0141
PREC0142
PREC0143
PREC0144
PREC0145
PREC0146
PREC0147
PREC0148
PREC0149
PREC0150
PREC0151
PREC0152
PREC0153
PREC0154
PREC0155
PREC0156
PREC0157
PREC0158
PREC0159
PREC0160
PREC0161
PREC0162
PREC0163
PREC0164
PREC0165
PREC0166
PREC0167
PREC0168
PREC0169
PREC0170
PREC0171
PREC0172
PREC0173
PREC0174
PREC0175
PREC0176
PREC0177
PREC0178
PREC0179
PREC0180
PREC0181
PREC0182
PREC0183
PREC0184
PREC0185
PREC0186
PREC0187
PREC0188
PREC0189
PREC0190
PREC0191
PREC0192
PREC0193
PREC0194
PREC0195
PREC0196

```

Figure 120. Program PRECOM Program Listing (Continued)

1	22H	COMMON /SC2/ S2(,15,1H),/,	PREC0197
2	22H	COMMON /SC3/ S3(,15,1H),/,	PREC0198
3	22H	COMMON /SC4/ S4(,15,1H)	PREC0199
		WRITE(6,360)	PREC0200
360	FORMAT(3)HC	MAXIMUM SCRATCH ARRAY DIMENSIONS)	PREC0201
		WRITE(6,380)MS1,MS2,MS3,MS4	PREC0202
380	FORMAT(6X,4)MS1= ,15,7H & MS2= ,15,7H & MS3= ,15,7H & MS4= ,15)		PREC0203
		WRITE(6,400)	PREC0204
400	FORMAT(3)HC	MAXIMUM SYSTEM DIMENSIONS)	PREC0205
		WRITE(6,420)NXM,NPM,NUM,NYM,MSH,MTB	PREC0206
420	FORMAT(6X,4)NXM= ,13,7H & NPM= ,13,7H & NUM= ,13,7H & NYM=		PREC0207
		1,13,7H & MSH= ,13,7H & MTB= ,13)	PREC0208
		WRITE(6,440)	PREC0209
440	FORMAT(4)HC	CALL KOMPACT ORGANIZING SUBROUTINE)	PREC0210
		WRITE(6,460)	PREC0211
460	FORMAT(16H	CALL KORG),/,25HC STOP EXECUTION	PREC0212
	1,/,10H	STOP,/,9H END)	PREC0213
		STOP	PREC0214
		END	PREC0215

Figure 120. Program PRECOM Program Listing (Concluded)

REFERENCES

1. Hink, G. R., et. al., "A Method for Predicting the Stability Characteristics of Control Configured Vehicles," Volume II - FLEXSTAB 2.01.00 User's Manual, AFFDL-TR-74-91, November 1974.
2. VanDierendonck, A. J., Hartmann, G. L., "Quadratic Methodology," Volume II - Documentation for Computer Program, Honeywell Report No. F0161-FR, October 1973.
3. "Scope Reference Manual," for the Control Data CYBER 70/Models 72, 73, and 74 and 6000 series computers, publication number 60307200, 1973.
4. Phillips, A. C., Rosen, R. W., "A Note on Dynamic Data Storage in FORTRAN IV," Computer Journals, Vol. 18, No. 4, November 1975.
5. Konar, A. F., "Improved Iterative Algorithm for Solving the Lyapunov Equation," Research Memo, MR-10702, Honeywell Systems and Research Center, Minneapolis, Minnesota, June 1968.
6. Stein, A., "Practicalization Algorithms," Research Memo, MR-10861, Honeywell Systems and Research Center, Minneapolis, Minnesota, January 1970.