

[REDACTED]

[REDACTED]

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFFDL-TR-75-146, Vol. II	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 76SRC20
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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

quadratic optimal or suboptimal control systems.

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

It 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
- Volume II. KONPACT Program Listing
- Volume III. KONPACT Users Manual

# *Contrails*

## 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
	38
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 KOR2	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 KORG2 Flow Chart	247 - 248
88	Subroutine KORG2 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

# *Contrails*

## 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)
	CONDK	Modifies the state space model by scaling, shuffling, truncating and residualizing the system variables
KONPACT-2		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

# Contrails

- 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	KORG1	
KORG1	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 RESPK
IDRO	Reorganizes the input data.	59	60	0,0		KORG1
FILE	Locates and inserts system labels on disc files and writes end of data mark on the disc files.	65	66	0,0		KORG1 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 SDRD SHIFT
ERRM	Prints error message.		77	0,0		RESPK MNAME RSDRD SDRD

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 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 REDUCE
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	SIMK1 DERRM TDINVR MPRS NAMEL DEBUG QDIO HPR	MAIN1
SIMK1	Reads simulator deck data and load equation data and implements them into simulation equations.	17	18	1, 0	DEBUG DERRM INPTJ MPRS1	STAMK1
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 DEBUG STAMK2	
STAMK2	Obtains state space model from Transfer function data and connection data (implemented in SIMKT subroutine).	19	20	2, 0	SIMKT DERRM TDINVR DEBUG MPRS HPR QDIO DERRMS NAMEL	MAIN2
SIMKT	Reads transfer function data and Connection data and implements them into simulation equations.	21	22	2, 0	DEBUG 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	DEBUG	SIMKT
DFN	Selects the specified pade approximation to transport (time) delay from a table of pade approximations.	25	26	2, 0	DEBUG	SIMKT
PHERR	Computes the phase error of pade approximation to transport (time) delay.	27	28	2, 0	DEBUG	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 inter-connection 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	DEBUG DERRM CONDK	

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	MAIN5
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	CONDK
SHIFT	Shifts the contents of old name list arrays into new name list arrays.		80	5, 0	DEBUG	MNAME

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 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	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)	NRM	Defined in MAIN
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	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)		

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	XDOT(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	RI(MTFB)		
S1(N7)	N7 = N6 + MTFB	UI(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	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 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 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	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(NRM*MB)		
S1(N10)	N10 = N9 + NRM*MB	NNX(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	UNFTI(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 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	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	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 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(I7, 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	NSHUF0(NRM)		
S1(L8)	L8 = L7 + NRM	NSHUF1(NUM)		
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	R(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	DESI(NUM, 10)		
S3(N12)	N12 = N11 + NUM*10	UNITI(NUM, 4)		
S3(N13)	N13 = N12 + NUM*4	NNNS(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	VNNO(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	DESNi(NUM, 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	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	DESH(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	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)		
S2(M1)	M1 = 1	AT(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*MN	Q(MN, NUM)		
S2(M7)	M7 = M6 + MN*NUM	R(NRM, MN)		
S2(M8)	M8 = M7 + NRM*MN	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)	NU	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	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)		

## 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		DIAK CALJ FFOC GCAI CAL
MP	Prints matrix data.		110	0,0		DIAK STRIC RESP FFOC
OUTP	Writes nonzero elements of a matrix on a data file.		111	0,0		DIAK FFOC
INPT	Reads nonzero elements of a matrix.			0,0		DIAK 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 eigenvalues of a matrix.		112	0,0	HESSEN QRCALL	DIAK EFOC
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 CALL OUTP TIMER POLES	
TIMER	Computes time response.		89	1,0	SGUST	DIAK
SGUST	Computes step gust input.		90	1,0		TIMER
CALL	Solves square Lyapunov equation.		91	1,0	TDINVR	COVAR COSTAT DIAK
STRIC	Computes stable set of starting gains for DIAK.		92	1,0	VIP	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
FFOC	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 VIP ZERO INPT SHUF TDINVR	
SHUF	Reorders rows and columns of matrices.		94	2,0		FFOC
RESP	Computes covariances for disturbance inputs.		95	2,0		FFOC
COVAR	Computes covariance matrix.		96	2,0	CAL GCAL	RESP
COSTAT	Computes costate matrix.		97	2,0	CAL GCAL	FFOC
TRANS	Computes gradient transformation matrix.		98	2,0		FFOC
UNSCR	Transforms the gradient transformation matrix.		99	2,0		FFOC
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	DDIAR DFFOC DELSA LTKK DERRAI ERRAI	

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 DIAK 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 KOR2

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.

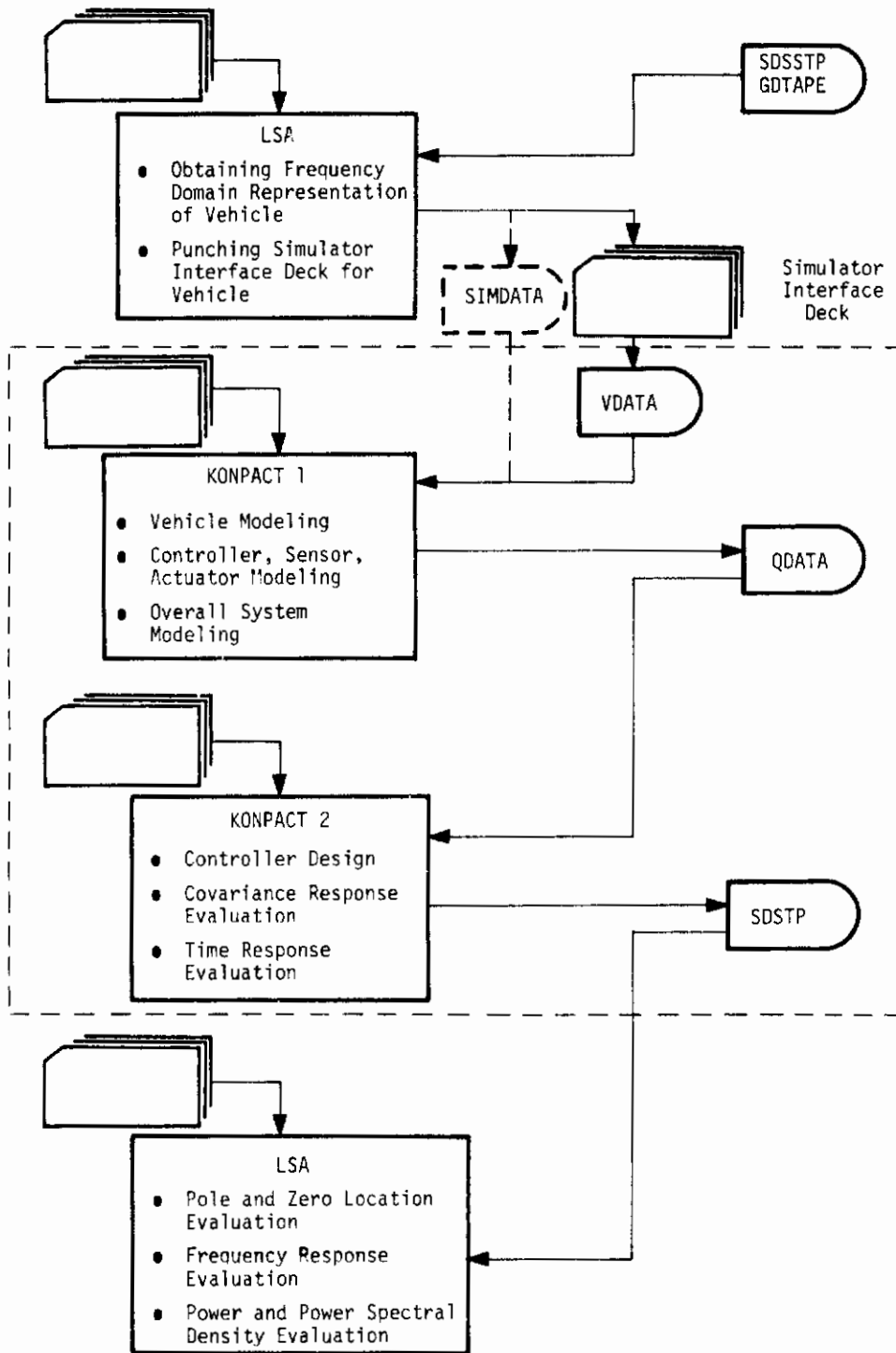


Figure 1. Interface Between LSA and KONPACT Programs



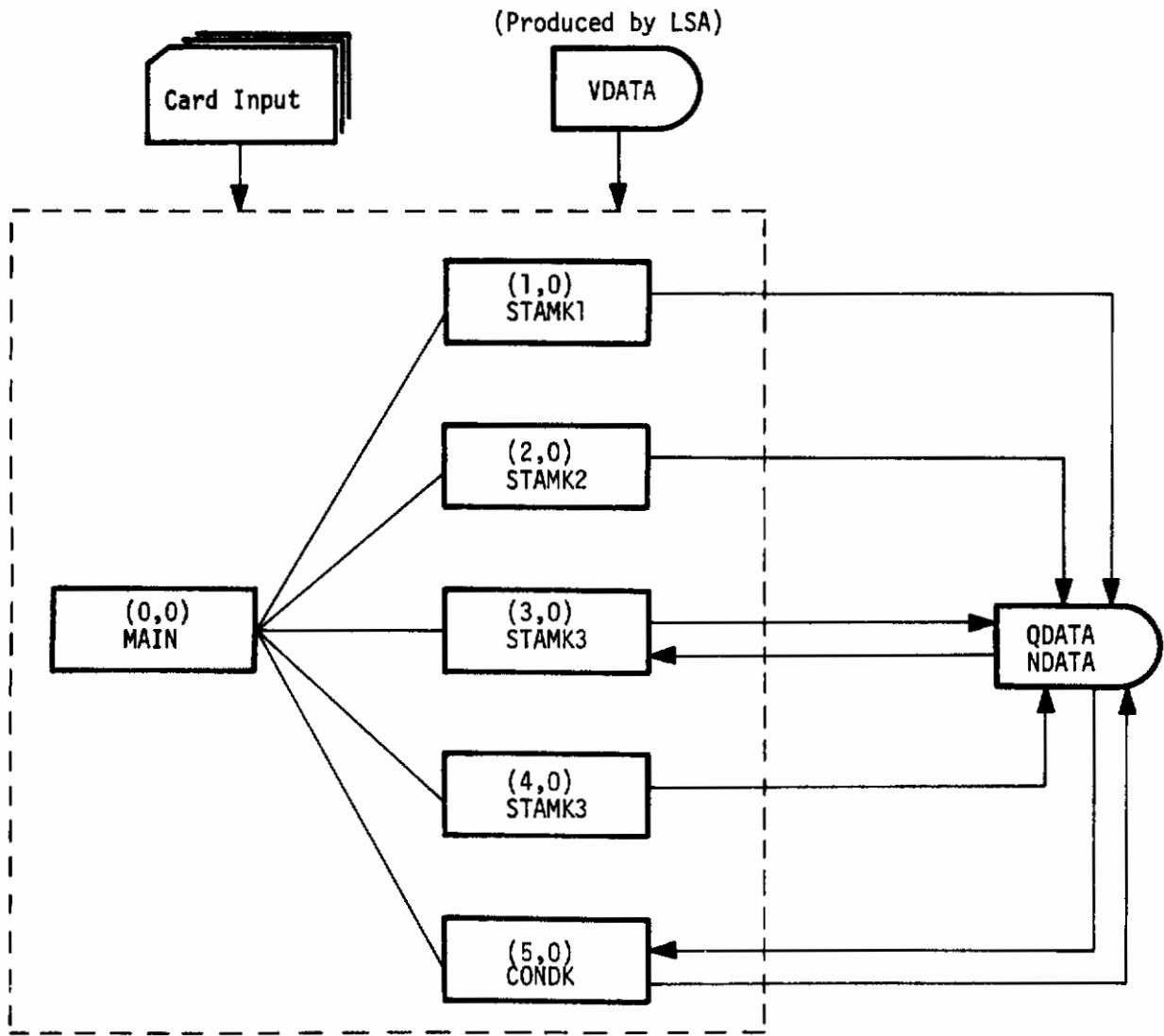


Figure 2. Overlay Structure of KONPACT-1

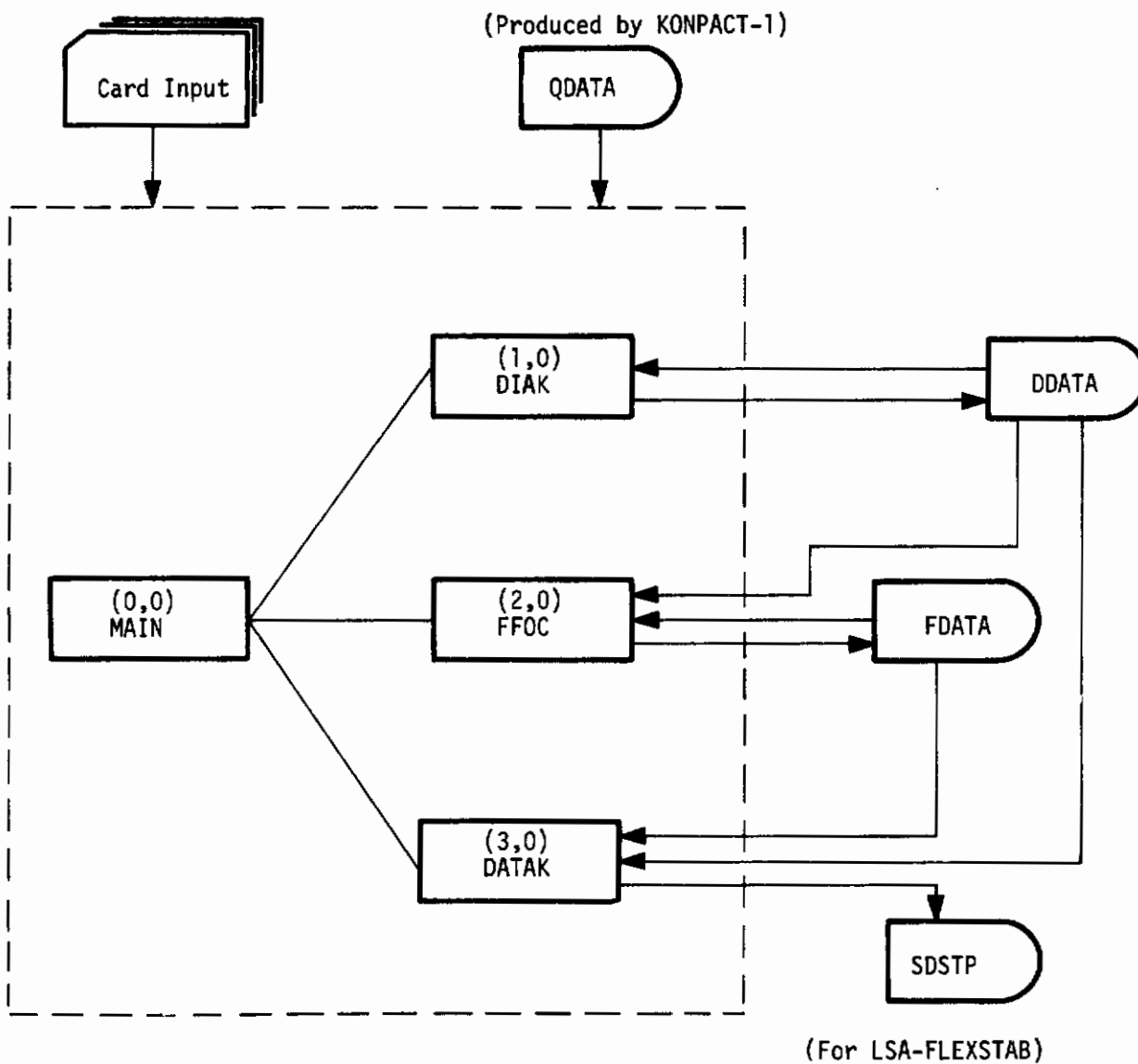


Figure 3. Overlay Structure of KONPACT-2

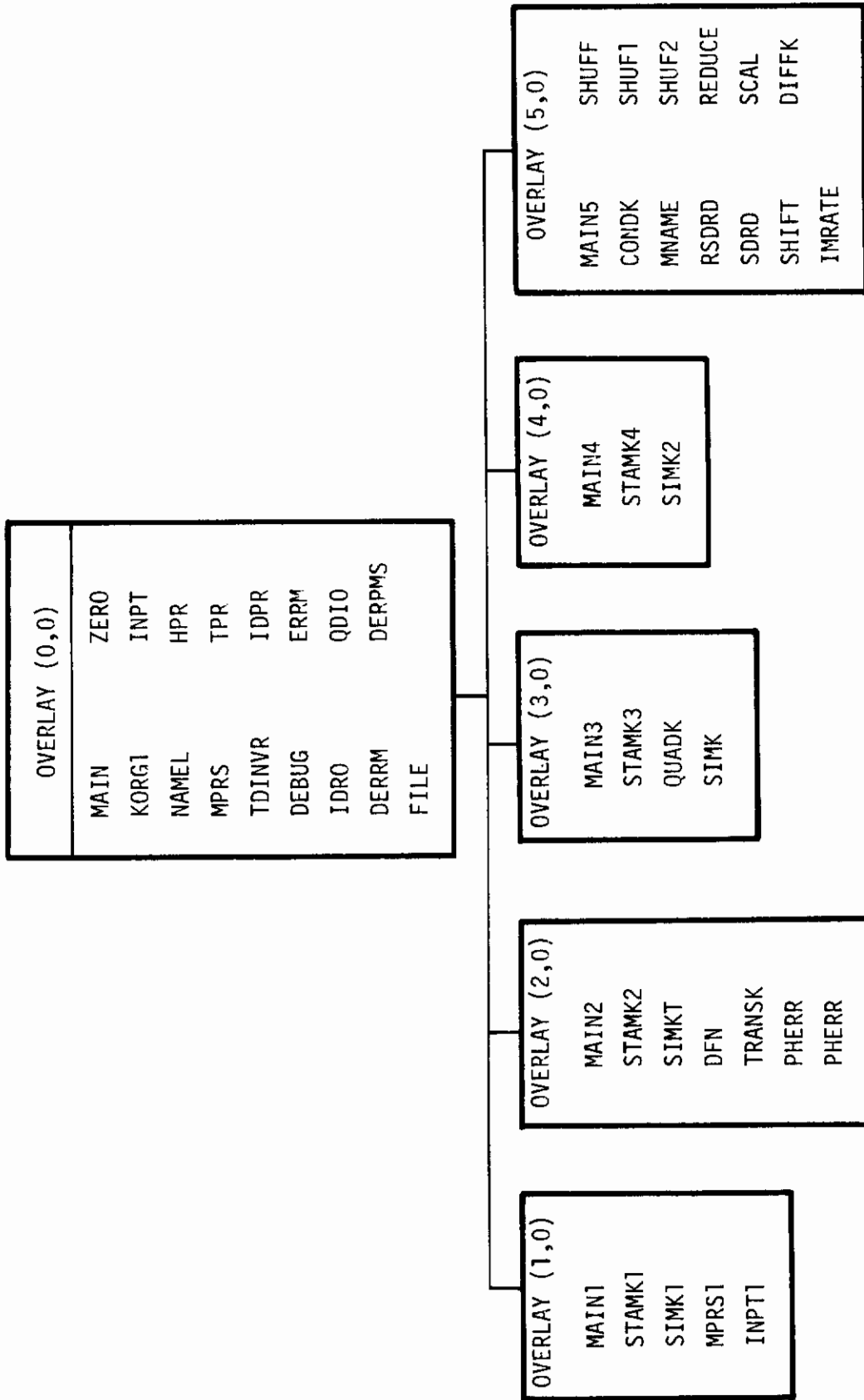


Figure 4. Overlay Structure and Subroutines in KONPACT-1

# Contrails

```

OVERLAY(KOVL,0,1)                                MAIN  2
PROGRAM MAIN(CHINPUT,INPUT,NRDATA,QDATA,OUTPUT,TAPES=RINPUT, MAIN  3
1TAPE6=INPUT,TAPF7=NRDATA,TAPF8=QDATA,TAPE9=OUTPUT,VDATA,  MAIN  4
2TAPE4=VDATA,SCRATCH,TAPE3=SCRATCH)             MAIN  5
C                                                MAIN  6
C PURPOSE - TO SET UP MAXIMUM DIMENSIONS          MAIN  7
C ANALISTS - 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   KORGI                                         MAIN 12
C                                                MAIN 13
C LABELLED COMMON LIST                          MAIN 14
C   MS1      MAXIMUM DIMENSION FOR SCRATCH ARRAY S1  MAIN 15
C   MS2      MAXIMUM DIMENSION FOR SCRATCH ARRAY S2  MAIN 16
C   MS3      MAXIMUM DIMENSION FOR SCRATCH ARRAY S3  MAIN 17
C   MS4      MAXIMUM DIMENSION FOR SCRATCH ARRAY S4  MAIN 18
C   MAXN     MAXIMUM ROW DIMENSION FOR SIMULA MATRIX F MAIN 19
C   MAXM     MAXIMUM COLUMN DIMENSION FOR SIMU MATRIX F MAIN 20
C   NX       MAXIMUM NUMBER OF STATES                MAIN 21
C   NR       MAXIMUM NUMBER OF OUTPUTS               MAIN 22
C   NU       MAXIMUM NUMBER OF INPUTS                MAIN 23
C   NY       MAXIMUM DIMENSION FOR INTERCONN EQUATIONS MAIN 24
C   NM       MAXIMUM OF (NRM,NUM)                   MAIN 25
C   MP       MAXIMUM DIMENSION FOR P ARRAY           MAIN 26
C   MQ       MAXIMUM DIMENSION FOR Q ARRAY           MAIN 27
C   MR       MAXIMUM DIMENSION FOR R ARRAY           MAIN 28
C   MS.(MR)  MAXIMUM NUMBER OF SYSTEMS FOR COMBINING MAIN 29
C   NF       MAXIMUM SYSTEM NUMBER - IMPLICIT MODEL MAIN 30
C   MS       SAME AS NXM                             MAIN 31
C   MN       MN=MS                                    MAIN 32
C   MTR.(MTR) MAXIMUM NO OF TRANSFER FUNCTION BLOCKS MAIN 33
C   MST      MAXIMUM POWER OF S IN THE TRANSFER FUNCTIONMAIN 34
C   MT       MAXIMUM NO OF TERMS IN THE TRANSFER FN  MAIN 35
C   S1       SCRATCH ARRAY FOR DYNAMIC STORAGE      MAIN 36
C   S2       SCRATCH ARRAY FOR DYNAMIC STORAGE      MAIN 37
C   S3       SCRATCH ARRAY FOR DYNAMIC STORAGE      MAIN 38
C   S4       SCRATCH ARRAY FOR DYNAMIC STORAGE      MAIN 39
C                                                MAIN 40
C   COMMON: /DIM/ MS1,MS2,MS3,MS4,MAXN,MAXM,NXM,NRM,NUM,NYM MAIN 41
C   1,MM,MP,MQ,MR,MSR,MR,MS,MN,MTR,MST,MT          MAIN 42
C   COMMON: /SC1/ S1(03591)                         MAIN 43
C   COMMON: /SC2/ S2(10000)                         MAIN 44
C   COMMON: /SC3/ S3(01213)                         MAIN 45
C   COMMON: /SC4/ S4(00001)                         MAIN 46
C                                                MAIN 47
C   MAXIMUM SCRATCH ARRAY DIMENSIONS                MAIN 48
C                                                MAIN 49
C   MS1=03591 & MS2=10000 & MS3=01213 & MS4=00001 MAIN 50
C                                                MAIN 51
C   MAXIMUM SYSTEM DIMENSIONS                      MAIN 52
C                                                MAIN 53
C   NXM=1 & NRM=1 & NUM=10 & NYM=30 & MSR=3 & MTR=10 MAIN 54
C                                                MAIN 55
C   CALL -ONPACT ORGANIZING SUBROUTINE             MAIN 56
C                                                MAIN 57
C   CALL KORGI                                       MAIN 58
C   STOP                                             MAIN 59
C   END                                              MAIN 60

```

Figure 5. Program MAIN Program Listing

# Contracts

```

OVERLAY(KONI,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 KONAR / J K MAHESH - THE HONEYWELL INC MAIN1 6
C DATE WRITTEN - 1975 MAIN1 7
C MAIN1 8
C SUBPROGRAMS CALLED MAIN1 9
C   DEBUG MAIN1 10
C   STAMK1 MAIN1 11
C   DEPRM MAIN1 12
C MAIN1 13
COMMON /DIM/ MS1,MS2,MS3,MS4,MAXN,MAXM,NXM,NRM,NUM,NYM
),MM,MP,MQ,MR,MR,MP,NR,MS,MN,MTFB,MTS,MT MAIN1 14
COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,J MAIN1 15
COMMON /SC1/ S1(1) MAIN1 16
C DIMENSION V(MAXN),X(MAXM),F(MAXN*MAXM),U(NUM) MAIN1 17
COMMON /SC2/ S2(1) MAIN1 18
C DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM) MAIN1 19
COMMON /SC3/ S3(1) MAIN1 20
C DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4) MAIN1 21
C DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4) MAIN1 22
C DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4) MAIN1 23
C IF(IPRINT.EQ.6)CALL DEBUG(1.4+MAIN,4H1 .1,0,IW) MAIN1 24
C MAIN1 25
C COMPUTE ARRAY START INDEXES MAIN1 26
C MAIN1 27
C FOR V,W,F,IJ MAIN1 28
C MAIN1 29
C N1=1 * N2=N1+MAXN * N3=N2+MAXM * N4=N3+MAXN*MAXM MAIN1 30
C N5=N4+NUM MAIN1 31
C MAIN1 32
C FOR A,R,C,D MAIN1 33
C MAIN1 34
C M1=1 * M2=M1+NXM*NXM * M3=M2+NXM*NUM * M4=M3+NRM*NXM MAIN1 35
C M5=M4+NRM*NUM MAIN1 36
C MAIN1 37
C FOR NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI MAIN1 38
C MAIN1 39
C L1=1 * L2=L1+NXM * L3=L2+NXM**2 * L4=L3+NXM*10 * L5=L4+NXM**4 MAIN1 40
C L6=L5+NRM * L7=L6+NRM**2 * L8=L7+NRM*10 * L9=L8+NRM**4 MAIN1 41
C L10=L9+NUM * L11=L10+NUM**2 * L12=L11+NUM*10 * L13=L12+NUM**4 MAIN1 42
C MAIN1 43
C CHECK IF SCRATCH ARRAY SIZES ARE SUFFICIENT MAIN1 44
C MAIN1 45
C IF((N5.GT.MS1).OR.(M5.GT.MS2).OR.(L13.GT.MS3)) MAIN1 46
C 1CALL DEPRM(N5,M5,L13,MS4,MS1,MS2,MS3,MS4,1,0,4+MAIN,4H1 .1,W) MAIN1 47
C IF(IPRINT.EQ.6)CALL DEBUG(2.4+MAIN,4H1 .1,0,IW) MAIN1 48
C MAIN1 49
C CALL SUBROUTINE STAMK1 MAIN1 50
C MAIN1 51
C CALL STAMK1(S1(N1),S1(N2),S1(L3),S1(N4),S2(M1),S2(M2),S2(M3), MAIN1 52
C 1S2(M4),S3(L1),S3(L2),S3(L3),S3(L4),S3(L5),S3(L6),S3(L7), MAIN1 53
C 2S3(L8),S3(L9),S3(L10),S3(L11),S3(L12),MAXN,MAXM,NXM,NRM,NUM, MAIN1 54
C 3NYM,M1,MS1,MS2,MS3,MS4,N5) MAIN1 55
C IF(IPRINT.EQ.6)CALL DEBUG(3.4+MAIN,4H1 .1,0,IW) MAIN1 56
C MAIN1 57
C MAIN1 58
C RETURN TO MAIN OVERLAY MAIN1 59
C MAIN1 60
C END MAIN1 61

```

Figure 6. Program MAIN1 Program Listing

# Contrails

```

OVERLAY(KON1,2,3)                                MAIN2  2
PROGRAM MAIN2                                     MAIN2  3
C                                                  MAIN2  4
C PURPOSE - TO SET UP DIMENSIONS AND CALL STAMK2  MAIN2  5
C ANALYSTS - A F KONAR / J K VAHESH - THE HONEYWELL INC MAIN2  6
C DATE WRITTEN - 1975                             MAIN2  7
C                                                  MAIN2  8
C SUBPROGRAMS CALLED                               MAIN2  9
C   DEHUG                                         MAIN2 10
C   STAMK2                                        MAIN2 11
C   DEPRM                                        MAIN2 12
C                                                  MAIN2 13
COMMON /INOUT/ IP,IM,IPRINT,INSERT,LOCATE, NULL, MARK(20),JN,JU,JS MAIN2 14
COMMON /DIM/ MS1,MS2,MS3,MS4,MAXN,MAXM,MAXV,NR1,NUM,NYM          MAIN2 15
1,MM,MP,MQ,MR,MR,NR,MS,MN,MTR,MST,MT                          MAIN2 16
COMMON /SC1/ S1(1)                                             MAIN2 17
C DIMENSION V(MAXN),W(MAXM),F(MAXM,MAXM)                   MAIN2 18
C DIMENSION XDOT(MST,MTR),X(MST,MTR),R1(1,MTR),UI(1,MTR)     MAIN2 19
C DIMENSION U(NUM),NX(MTR),NR(MTR),NU(MTR)                 MAIN2 20
COMMON /SC2/ S2(1)                                             MAIN2 21
C DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM)     MAIN2 22
C DIMENSION AT(MST,MST,MTR),BT(MST,1,MTR)                  MAIN2 23
C DIMENSION CT(1,MST,MTR),DT(1,1,MTR)                      MAIN2 24
C DIMENSION P(MTR,MTR),Q(MTR,NUM),R(NRM,MTR),S(NRM,NUM)    MAIN2 25
C DIMENSION PRINT(2,MT),HS(2,MT,MTR)                       MAIN2 26
COMMON /SC3/ S3(1)                                             MAIN2 27
C DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)   MAIN2 28
C DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)   MAIN2 29
C DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)   MAIN2 30
IF(IPRINT.EQ.6)CALL DEHUG(1.4HMAIN,4M2  2,0,14)              MAIN2 31
C                                                  MAIN2 32
C COMPUTE ARRAY START INDEXES                               MAIN2 33
C                                                  MAIN2 34
C FOR V,W,F,XDOT,X,R1,UI,U,NX,NR,NU                       MAIN2 35
C                                                  MAIN2 36
N1=1  $ N2=N1+MAXM  $ N3=N2+MAXM  $ N4=N3+MAXM*MAXM          MAIN2 37
N5=N4+MST*MTR  $ N6=N5+MST*MTR  $ N7=N6+MTR  $ N8=N7+MTR    MAIN2 38
N9=NR+NUM  $ N10=N9+MTR  $ N11=N10+MTR  $ N12=N11+MTR      MAIN2 39
C                                                  MAIN2 40
C FOR A,B,C,D                                               MAIN2 41
C                                                  MAIN2 42
N1=1  $ M2=M1+NXM*NXM  $ M3=M2+NXM*NUM  $ M4=M3+NRM*NXM    MAIN2 43
M5=M4+NRM*NUM                                               MAIN2 44
C                                                  MAIN2 45
C FOR AT,BT,CT,DT,P,Q,R,S,PRINT,HS                        MAIN2 46
C                                                  MAIN2 47
K1=1  $ K2=K1+MST*MST*MTR  $ K3=K2+MST*MTR  $ K4=K3+MST*MTR MAIN2 48
K5=K4+MTR  $ K6=K5+MTR*MTR  $ K7=K6+MTR*NUM  $ K8=K7+NRM*MTR MAIN2 49
K9=K8+NRM*NUM  $ K10=K9+2*MTR  $ K11=K10+2*MTR*MTR        MAIN2 50
C                                                  MAIN2 51
C FOR NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI MAIN2 52
C                                                  MAIN2 53
L1=1  $ L2=L1+NXM  $ L3=L2+NXM*2  $ L4=L3+NXM*10  $ L5=L4+NXM*4 MAIN2 54
L6=L5+NRM  $ L7=L6+NRM*2  $ L8=L7+NRM*10  $ L9=L8+NRM*4    MAIN2 55
L10=L9+NUM  $ L11=L10+NUM*2  $ L12=L11+NUM*10  $ L13=L12+NUM*4 MAIN2 56
C                                                  MAIN2 57
C CHECK IF SCRATCH ARRAY SIZES ARE SUFFICIENT             MAIN2 58
C                                                  MAIN2 59
MKS11=M5                                                     MAIN2 60
IF(K11.GT.M5)MKS11=M11                                       MAIN2 61
IF((N12.GT.MS1).OR.(M5.GT.MS2).OR.(K11.GT.MS2).OR.(L13.GT.MS3)) MAIN2 62
CALL DEPRM(N12,MKS11,L13,MS4,MS1,MS2,MS3,MS4,2,0,4HMAIN,4M2  14)MAIN2 63
IF(IPRINT.EQ.6)CALL DEHUG(2.4HMAIN,4M2  2,0,14)              MAIN2 64

```

**Figure 7. Program MAIN2 Program Listing**

```
C          MAIN2 65
C          CALL SUBROUTINE STAMK2          MAIN2 66
C          CALL STAMK2(S1(N1),S1(N2),S1(N3),S1(N4),S1(N5),S1(N6),S1(N7),
1S1(N8),S1(N9),S1(N10),S1(N11),S2(M1),S2(M2),S2(M3),S2(M4),S2(K1),
2S2(K2),S2(K3),S2(K4),S2(K5),S2(K6),S2(K7),S2(K8),S2(K9),S2(K10),
3S3(L1),S3(L2),S3(L3),S3(L4),S3(L5),S3(L6),S3(L7),S3(L8),
4S3(L9),S3(L10),S3(L11),S3(L12),MAXN,MAXM,NXM,NRM,NUM,VYM,MR,MTFB,
5MST,MT,MS1,MS2,MS3,MS4,NR)          MAIN2 72
          IF(IPPRINT.EQ.6)CALL DEHUG(3.4HMAIN.4H2  *2.0*TW)          MAIN2 74
C          RETURN TO MAIN OVERLAY          MAIN2 75
C          MAIN2 76
C          MAIN2 77
          END          MAIN2 78
```

Figure 7. Program MAIN2 Program Listing (Concluded)

# Contracts

```

OVERLAY(KOR1,3, ) MAIN3 2
PROGRAM MAIN3 MAIN3 3
C MAIN3 4
C PURPOSE - TO SET UP DIMENSIONS AND CALL STACKS MAIN3 5
C ANALYSIS - A F KONAR / J K MARSH - THE HONEYWELL INC MAIN3 6
C DATE - PITIEN - 1975 MAIN3 7
C MAIN3 8
C SUBPROGRAMS CALLED MAIN3 9
C DEFORM MAIN3 10
C STACKS MAIN3 11
C MAIN3 12
COMMON /I/O/ IZ, IR, IW, IPRINT, I, ISET, LOCATE, NULL, MARK(20), JN, JU, JS MAIN3 13
COMMON /DIM/ MS1, MS2, MS3, MS4, MAXN, MAXM, NXM, NRM, NUM, NYM MAIN3 14
COMMON /M/ M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12, M13, M14, M15, M16, M17, M18, M19, M20, M21, M22, M23, M24, M25, M26, M27, M28, M29, M30, M31, M32, M33, M34, M35, M36, M37, M38, M39, M40, M41, M42, M43, M44, M45, M46, M47, M48, M49, M50, M51, M52, M53, M54, M55, M56, M57, M58, M59, M60, M61, M62, M63, M64
COMMON /S/ S1(1) MAIN3 16
C DIMENSION V(MAXN), W(MAXM), F(MEAN, MAX4) MAIN3 17
C DIMENSION XDOT(NXM, MR), X(NX, MR), RI(NXM, MR), UI(NUM, MB) MAIN3 18
C DIMENSION RIN(NRM, MR), R(NUM, N), X(MR), NNR(MR), NNI(MR) MAIN3 19
COMMON /SC2/ S2(1) MAIN3 20
C DIMENSION A(NXM, NX), B(NXM, NR), C(NRM, NX), D(NRM, NR) MAIN3 21
COMMON /SC3/ S3(1) MAIN3 22
C DIMENSION VNS(NXM), VNS(NXM, 2), DESS(NXM, 10), UNITS(NXM, 4) MAIN3 23
C DIMENSION VNO(NRM), VNO(NRM, 2), DESO(NRM, 10), UNITS(NRM, 4) MAIN3 24
C DIMENSION VNI(NUM), 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 (IPRINT.EQ.5) WRITE(IW,165)MS1,MS2,MS3,MS4,MAXN,MAXM MAIN3 29
COMMON /M/ M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11, M12, M13, M14, M15, M16, M17, M18, M19, M20, M21, M22, M23, M24, M25, M26, M27, M28, M29, M30, M31, M32, M33, M34, M35, M36, M37, M38, M39, M40, M41, M42, M43, M44, M45, M46, M47, M48, M49, M50, M51, M52, M53, M54, M55, M56, M57, M58, M59, M60, M61, M62, M63, M64
165 FORMAT(1X,15(15,1X)) MAIN3 31
C MAIN3 32
C COMPUTE MAXIMUM SIZE FOR RI MAIN3 33
C MAIN3 34
NRMM=NRM*MR MAIN3 35
C MAIN3 36
C COMPUTE ARRAY START INDEXES MAIN3 37
C MAIN3 38
FOR V,W,F,XDOT,X,RI,UI,RIN,I,NNX,NNR,VNU MAIN3 39
C MAIN3 40
N1=1 N2=N1+MAXN N3=N2+MAXM N4=N3+MAXN*MAXM MAIN3 41
N5=N4+NX*MR N6=N5+NX*MR N7=N6+NR*MR N8=N7+NUM*MB MAIN3 42
N9=N8+NUM N10=N9+NR*MR N11=N10+MR N12=N11+MR N13=N12+MB MAIN3 43
C MAIN3 44
C FOR A,B,C,D MAIN3 45
C MAIN3 46
M1=1 M2=M1+NX*NX M3=M2+NX*NUM M4=M3+NR*NX MAIN3 47
M5=M4+NR*NUM MAIN3 48
C MAIN3 49
C FOR VNS,VNS,DESS,UNITS,VNO,VNO,DESO,UNITS,VNI,VNI,DESI,UNITS MAIN3 50
C MAIN3 51
L1=1 L2=L1+NX M L3=L2+NX*MR M L4=L3+NX*10 M L5=L4+NX*4 MAIN3 52
L6=L5+NR M L7=L6+NR*MR M L8=L7+NR*10 M L9=L8+NR*4 MAIN3 53
L10=L9+NUM M L11=L10+NUM*MR M L12=L11+NUM*10 M L13=L12+NUM*4 MAIN3 54
C MAIN3 55
C PRINT ARRAY OVERLAPPING NUMBERS IF NEEDED MAIN3 56
C MAIN3 57
C IF (IPRINT.EQ.6) WRITE(IW,165)N1,N2,N3,N4,N5,N6 MAIN3 58
C IF (IPRINT.EQ.6) WRITE(IW,165)N7,N8,N9,N10,N11,N12,N13 MAIN3 59
C IF (IPRINT.EQ.6) WRITE(IW,165)M1,M2,M3,M4,M5 MAIN3 60
C IF (IPRINT.EQ.6) WRITE(IW,165)L1,L2,L3,L4,L5,L6 MAIN3 61
C 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



# Contracts

```
C                                     MAIN3 65
      IF ((N13.GT.MS1).OR.(M5.GT.MS2).OR.(L13.GT.MS3))
      1CALL DCRRM(N13,MK59,L13,MS4,MS1,MS2,MS3,MS4,3,1,4HMAIN,4H3   .IW) MAIN3 66
C                                     MAIN3 67
C      CALL SUBROUTINE STAMK3                                     MAIN3 68
C                                     MAIN3 69
C      CALL STAMK3(S1(N1),S1(N2),S1(N3),S1(N4),S1(N5),S1(N6),S1(N7),
      1S1(N8),S1(N9),S1(N10),S1(N11),S1(N12),S2(M1),S2(M2),S2(M3),S2(M4),MAIN3 70
      2S3(L1),S3(L2),S3(L3),S3(L4),S3(L5),S3(L6),S3(L7),S3(L8),
      3S3(L9),S3(L10),S3(L11),S3(L12),MAXN,MAXM,NXM,NPM,NUM,NYM,MH,
      4MN,MM,MP,MQ,MR,MS1,MS2,MS3,4S1,NB,NRMMH) MAIN3 71
C                                     MAIN3 72
C                                     MAIN3 73
C      RETURN TO MAIN OVERLAY                                     MAIN3 74
C                                     MAIN3 75
C                                     MAIN3 76
C                                     MAIN3 77
C      END                                                       MAIN3 78
C                                     MAIN3 79
```

Figure 8. Program MAIN3 Program Listing (Concluded)

# Contrails

```

OVERLAY(KON),4.0)
PROGRAM MAIN4
C
C PURPOSE - TO SET UP DIMENSIONS AND CALL STAMK4
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C DATE WRITTEN - 1975
C
C SUBPROGRAMS CALLED
C DERRM
C STAMK4
C
COMMON /DIM/ MS1,MS2,MS3,MS4,MAXN,MAXM,NXM,NRM,NUM,NYM
1,MM,MP,MO,MR,NR,MS,MN,MTFB,MST,MT
COMMON /INPUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JO,JS
COMMON /SC1/ S1(1)
C DIMENSION V(MAXN),W(MAXM),F(MAXN*MAXM),U(NUM)
COMMON /SC2/ S2(1)
C DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
COMMON /SC3/ S3(1)
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
C COMPUTE ARRAY START INDEXES
C
C FOR V,W,F,U
C
N1=1 $ N2=N1+MAXN $ N3=N2+MAXM $ N4=N3+MAXN*MAXM
N5=N4+NUM
C
C FOR A,R,C,D
C
M1=1 $ M2=M1+NXM*NXM $ M3=M2+NXM*NUM $ M4=M3+NRM*NXM
M5=M4+NRM*NUM
C
C FOR NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI
C
L1=1 $ L2=L1+NXM $ L3=L2+NXM*2 $ L4=L3+NXM*10 $ L5=L4+NXM*4
L6=L5+NRM $ L7=L6+NRM*2 $ L8=L7+NRM*10 $ L9=L8+NRM*4
L10=L9+NUM $ L11=L10+NUM*2 $ L12=L11+NUM*10 $ L13=L12+NUM*4
C
C CHECK IF SCRATCH APRAY SIZES ARE SUFFICIENT
C
IF((N5.GT,MS1).OR.(M5.GT,MS2).OR.(L13.GT,MS3))
1CALL DERRM(N5,M5,L13,MS4,MS1,MS2,MS3,MS4,4,0,4+MAIN,4M4 .IW)
C
CALL SUBROUTINE STAMK4
C
CALL STAMK4(S1(N1),S1(N2),S1(N3),S1(N4),S2(M1),S2(M2),S2(M3),
1S2(M4),S3(L1),S3(L2),S3(L3),S3(L4),S3(L5),S3(L6),S3(L7),
2S3(L8),S3(L9),S3(L10),S3(L11),S3(L12),MAXM,MAXM,NXM,NRM,NUM,
3NYM,MM,MS1,MS2,MS3,MS4,NB)
C
C RETURN TO MAIN OVERLAY
C
END
MAIN4 2
MAIN4 3
MAIN4 4
MAIN4 5
MAIN4 6
MAIN4 7
MAIN4 8
MAIN4 9
MAIN4 10
MAIN4 11
MAIN4 12
MAIN4 13
MAIN4 14
MAIN4 15
MAIN4 16
MAIN4 17
MAIN4 18
MAIN4 19
MAIN4 20
MAIN4 21
MAIN4 22
MAIN4 23
MAIN4 24
MAIN4 25
MAIN4 26
MAIN4 27
MAIN4 28
MAIN4 29
MAIN4 30
MAIN4 31
MAIN4 32
MAIN4 33
MAIN4 34
MAIN4 35
MAIN4 36
MAIN4 37
MAIN4 38
MAIN4 39
MAIN4 40
MAIN4 41
MAIN4 42
MAIN4 43
MAIN4 44
MAIN4 45
MAIN4 46
MAIN4 47
MAIN4 48
MAIN4 49
MAIN4 50
MAIN4 51
MAIN4 52
MAIN4 53
MAIN4 54
MAIN4 55
MAIN4 56
MAIN4 57

```

Figure 9. Program MAIN4 Program Listing

# Contrails

```

OVERLAY(KONI,S,P)
PROGRAM MAIN5
MAINS 2
MAINS 3
MAINS 4
PURPOSE - TO SET UP DIMENSIONS AND CALL CONDK
MAINS 5
ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
MAINS 6
DATE WRITTEN - 1975
MAINS 7
SUBPROGRAMS CALLED
MAINS 8
  DERRM
MAINS 9
  RESPK
MAINS 10
  DEBUG
MAINS 11
MAINS 12
MAINS 13
COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
MAINS 14
COMMON /DIM/ MS1,MS2,MS3,MS4,MAXN,MAXM,NRM,NUM,NYM
MAINS 15
1,MM,MP,MQ,MR,MR,NH,MS,MN,MTFB,MST,MT
MAINS 16
COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)
MAINS 17
1,PHEAD(20)
MAINS 18
COMMON /SC2/ S2(1)
MAINS 19
: DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
MAINS 20
: DIMENSION CM(NRM,NXM),DM(NRM,NUM)
MAINS 21
COMMON /SC3/ S3(1)
MAINS 22
: DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
MAINS 23
: DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
MAINS 24
: DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
MAINS 25
: DIMENSION NNNS(NXM),VNNNS(NXM,2),DESNS(NXM,10),UNITNS(NXM,4)
MAINS 26
: DIMENSION NNNO(NRM),VNNNO(NRM,2),DESN0(NRM,10),UNITNO(NRM,4)
MAINS 27
: DIMENSION NNNI(NUM),VNNNI(NUM,2),DESVI(NUM,10),UNITNI(NUM,4)
MAINS 28
COMMON /SC1/ S1(1)
MAINS 29
: DIMENSION DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22),DUMMY3(NUM)
MAINS 30
: DIMENSION ES(NXM,NUM),ER(NRM,NUM)
MAINS 31
: DIMENSION NSHUF5(NXM),NSHUF0(NRM),NSHJFI(NUM)
MAINS 32
: DIMENSION CS(NRM,NXM),DS(NRM,NUM),CW(NRM,NXM),DW(NRM,NUM)
MAINS 33
: DIMENSION IPS(NRM),Q(NRM,NRM)
MAINS 34
IF(IPRINT,EQ,6)CALL DEBUG(1,4HMAIN,4H5 .5,0,1W)
MAINS 35
NXUM=NXM*NUM $ MWORD=17 $ NRSM=1
MAINS 36
NDM11=MAX0(MWORD,NXM,NRM,NRSM)
MAINS 37
NDM12=MAX0(NXUM,NRM)
MAINS 38
NDM21=MAX0(NRM,NXM,NRSM)
MAINS 39
NDM22=MAX0(NXM,NUM,NRM)
MAINS 40
C
MAINS 41
C
MAINS 42
PRINT ERROR MESSEGE IF DIMENSION OF SCRATCH ARRAYS ARE INSUFFICIENT
MAINS 43
C
M1=M1 $ M2=M1+NXM*NXM $ M3=M2+NXM*NUM $ M4=M3+NRM*NXM
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*4
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+NUM*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+NUM*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
C
MAINS 60
CALL SUBROUTINE CONDK
MAINS 61
C
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

# Contrails

```
2S3(N7),S3(N8),S3(N9),S3(N10),S3(N11),S3(N12),      MAIN5 65
3S3(N13),S3(N14),S3(N15),S3(N16),S3(N17),S3(N18),    MAIN5 66
4S3(N19),S3(N20),S3(N21),S3(N22),S3(N23),S3(N24),    MAIN5 67
5S1(L1),S1(L2),S1(L3),S1(L4),S1(L5),S1(L6),        MAIN5 68
6S1(L7),S1(L8),S1(L9),S1(L10),S1(L11),S1(L12),     MAIN5 69
7S1(L13),S1(L14),NXM,NRM,NUM,NDM11,NDM12,NDM21,NDM22) MAIN5 70
  IF(IPRINT.EQ.6)CALL DFRUG(3,4HMAIN,4H5  .5,0,IW)    MAIN5 71
C                                                       MAIN5 72
C RETURN TO MAIN OVEPLAY                               MAIN5 73
C                                                       MAIN5 74
END                                                     MAIN5 75
```

Figure 10. Program MAIN5 Program Listing (Concluded)

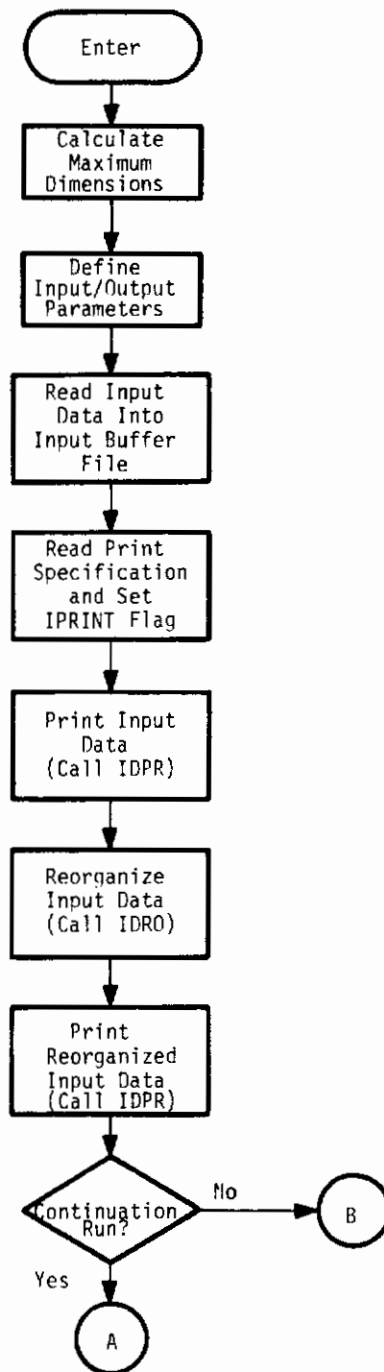


Figure 11. Subroutine KORGI Flow Chart

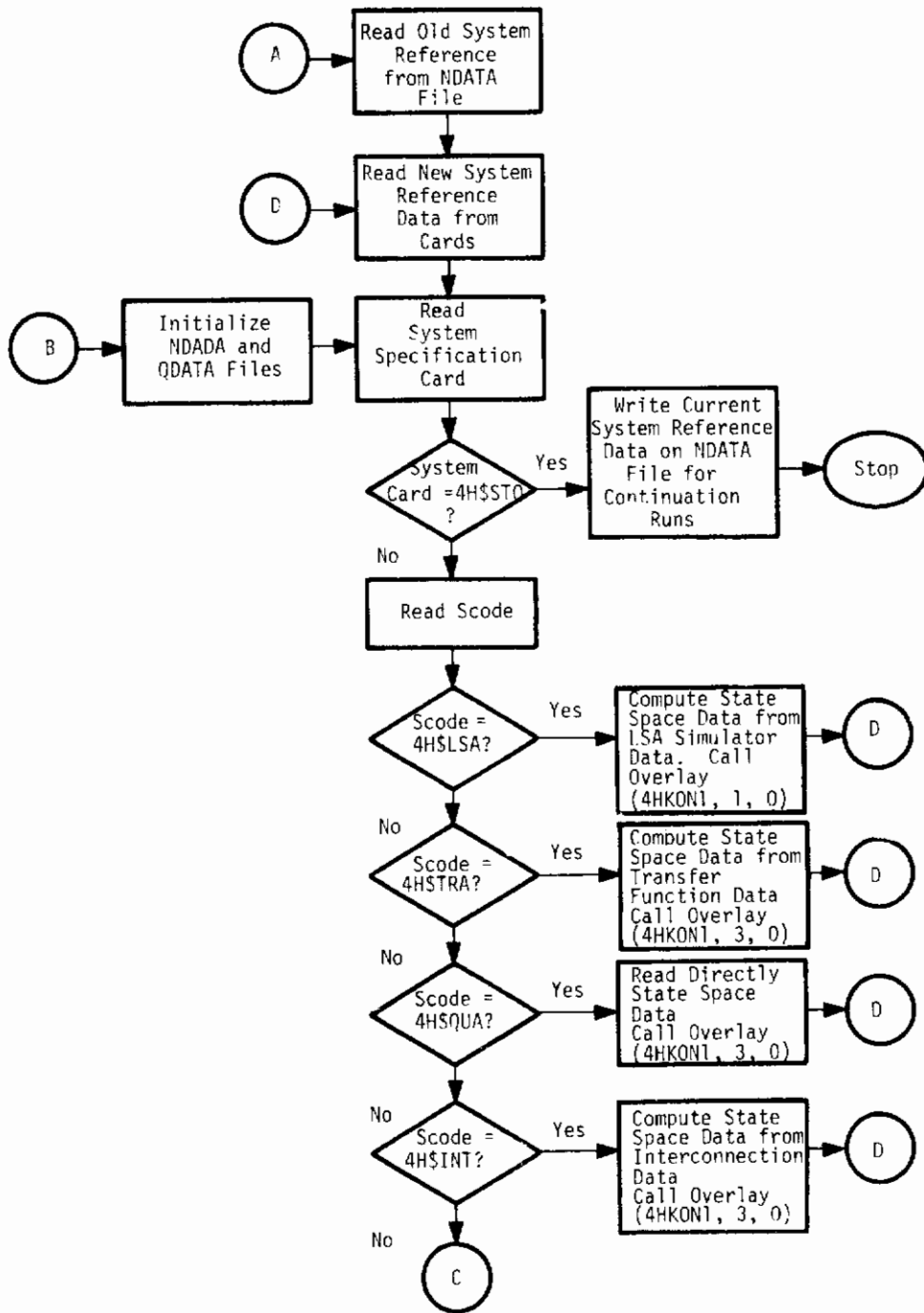


Figure 11. Subroutine KORGI Flow Chart (Continued)

# Contrails

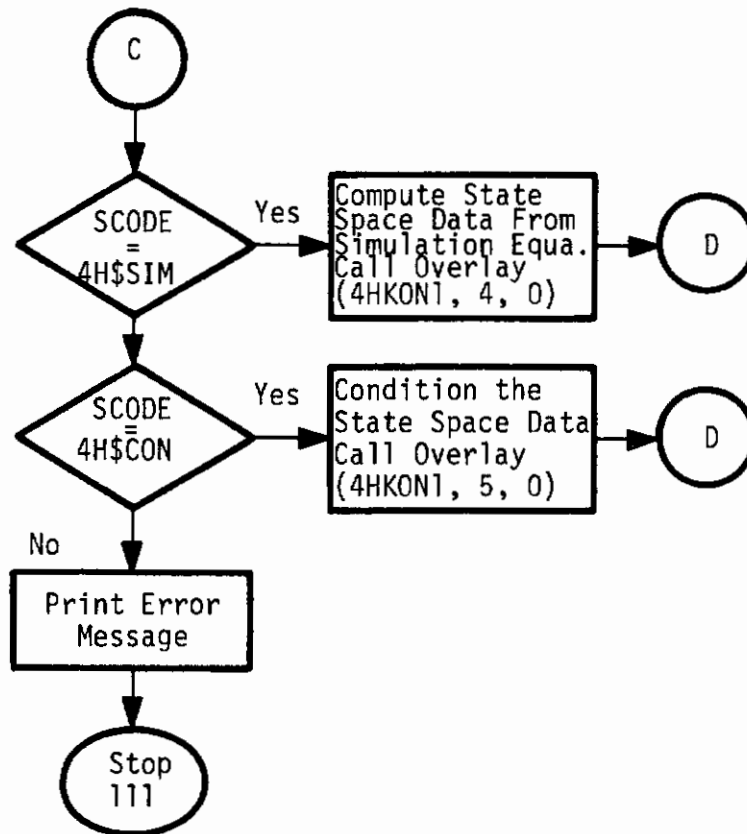


Figure 11. Subroutine KORGI Flow Chart (Concluded)

# Contrails

```

SUBROUTINE KORGI
C
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C PURPOSE - TO ORGANIZE EXECUTION OF KONPACT-1 PROGRAMS
C DATE WRITTEN - JULY 1975
C
C SUBPROGRAMS CALLED
C HPR
C IDPO
C IDPR
C FILE
C
C LABELLED COMMON LIST
C IR FILE NUMBER FOR CARD READER
C IW FILE NUMBER FOR LINE PRINTER
C IPRINT PRINT CONTROL FLAG
C INSERT HOLLFRITH INSE
C LOCATE HOLLFRITH LOCA
C NULL HOLLFRITH NULL
C MARK HOLLFRITH $$$
C JN FILE NO FOR NAME LIST DATA FILE
C JO FILE NO FOR QUADRUPLE DATA FILE
C JS FILE NO FOR SCRATCH FILE
C IHEAD LABEL NAME
C SCODE SYSTEM CODE WORD
C SDES SYSTEM DESCRIPTION
C MSYS PRESENT SYSTEM NUMBER
C HEAD PRESENT SYSTEM HEADING
C NSYS SYSTEM NUMBERS
C SHEAD SYSTEM HEADINGS
C PHEAD PREVIOUS SYSTEM HEADING
C
COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)
1,PHEAD(20)
COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JO,JS
COMMON /DIM/ MS1,MS2,MS3,MS4,MAXM,MAXN,NXM,NRM,NUM,NYM
1,MM,MO,MQ,MR,MR,NB,MS,MN,MTF8,MST,MT
DIMENSION CARD(20),LABEL(20),AHEAD(20,20)
INTEGER HINSE,HLOCA,HNULL,HNOLR
DATA HINSE,HLOCA,HNULL,HNOLR/4HINSE,4HLOCA,4HNULL,4H$$$ /
DATA HSTOP,HSYSTEM,HFM,HC,HCONT/4HSTOP,4HSYSTEM,4HFM,2HC,4HCONT/
DATA HPRIN,HTHIN,HFRYT/4HPRIN,4HTHIN,4HFRYT/
DATA HPUT,HNAL,HPUT/4HPUT,4HNAL,4HPUT /
DATA HDLSA,HDCON,HQUA/4HLSA,4HCON,4HSUA/
DATA HDTRA,HDINT,HDSIM/4HSTRA,4H$INT,4HSSIM/
DATA HREFE,HRENC,HFBR,HRBR/4HREFE,4HPENC,4HE,4H /
DATA HEND/4HEND /
C
C MAXIMUM DIMENSIONS FOR V,W AND F
C
C MAXM=NXM+NYM+NRM $ MAXN=NXM+NYM+NXM+NUM
C
C MAXIMUM DIMENSIONS FOR COMBINING TRANSFER FUNCTION BLOCKS
C
C MST=5 $ MT=6
C
C MAXIMUM DIMENSIONS FOR COMBINING SUBSYSTEMS
C
C MM=NRM
C IF(MM,LT,NUM)MM=NUM
C MS=NXM $ MQ=MR $ MR=MR $ MP=MR*2 $ MN=MM*MR
C
C MAXIMUM SYSTEM NUMBER

```

Figure 12. Subroutine KORGI Program Listing



# Contrails

```
C                                     KORGI 65
      NR=9                             KORGI 66
C                                     KORGI 67
C      DEFINE INPUT/OUTPUT PARAMETERS  KORGI 68
C                                     KORGI 69
      IR=5 * IW=9 $ IPRINT=4 $ JN=7 $ JQ=8 $ JS=3  KORGI 70
      INSERT=HINSE $ LOCATE=HLOCA $ NULL=HNULL    KORGI 71
      DO 10 I=1,20                             KORGI 72
100  MARK(I)=HDOLR                             KORGI 73
      LABEL(1)=HREFE                            KORGI 74
      LABEL(2)=HRENC                             KORGI 75
      LABEL(3)=HERRR                             KORGI 76
      DO 105 I=4,20                              KORGI 77
105  LABEL(I)=HRRRR                             KORGI 78
C                                     KORGI 79
C      READ INPUT DATA INTO INPUT DATA BUFFER FILE  KORGI 80
C                                     KORGI 81
      REWIND IR                                  KORGI 82
110  CONTINUE                                   KORGI 83
      READ(A,120)CARD                            KORGI 84
      IF(EOP(6))140,115                          KORGI 85
115  CONTINUE                                   KORGI 86
      WRITE(IR,120)CARD                          KORGI 87
120  FORMAT(20A4)                                KORGI 88
      GO TO 110                                   KORGI 89
140  CONTINUE                                   KORGI 90
      ENDFILE IR                                 KORGI 91
      REWIND IR                                  KORGI 92
C                                     KORGI 93
C      READ PRINT SPECIFICATION AND SET IPRINT  KORGI 94
C                                     KORGI 95
142  CONTINUE                                   KORGI 96
      READ(IR,170)CARD                            KORGI 97
      DECODE(4,143,CARD(1))CC,DUMMY              KORGI 98
143  FORMAT(A2,A2)                               KORGI 99
      IF(CC.EQ.HC)GO TO 142                       KORGI100
      IF(CARD(1).NE.HPRIN)GO TO 152              KORGI101
      IF(CARD(3).EQ.HTHIN)IPRINT=0              KORGI102
      IF(CARD(3).EQ.HTHIN)GO TO 142             KORGI103
      IF(CARD(3).EQ.HERYT)IPRINT=6              KORGI104
      IF(CARD(3).EQ.HERYT)GO TO 142            KORGI105
      IF(CARD(3).NE.HTPUT)GO TO 144            KORGI106
      IF(IPRINT.EQ.1)IPRINT=5                   KORGI107
      IF(IPRINT.EQ.5)GO TO 142                  KORGI108
      IPRINT=3                                   KORGI109
      GO TO 142                                  KORGI110
144  CONTINUE                                   KORGI111
      IF(CARD(3).NE.HNAL)GO TO 144              KORGI112
      IF(IPRINT.EQ.1)IPRINT=4                   KORGI113
      IF(IPRINT.EQ.4)GO TO 142                  KORGI114
      IPRINT=2                                   KORGI115
      GO TO 142                                  KORGI116
146  CONTINUE                                   KORGI117
      IF(CARD(3).NE.HPUT)GO TO 148              KORGI118
      IF(IPRINT.EQ.4)IPRINT=1                   KORGI119
      IF(IPRINT.EQ.1)GO TO 142                  KORGI120
      IF(IPRINT.EQ.2)IPRINT=4                   KORGI121
      IF(IPRINT.EQ.3)IPRINT=5                   KORGI122
      IF(IPRINT.EQ.4)GO TO 142                  KORGI123
      IF(IPRINT.EQ.5)GO TO 142                  KORGI124
      IPRINT=1                                   KORGI125
      GO TO 142                                  KORGI126
C                                     KORGI127
C      PRINT ERROR MESSAGE                     KORGI128
C                                     KORGI129
148  CONTINUE                                   KORGI130
```

Figure 12. Subroutine KORGI Program Listing (Continued)

# Contrails

```
      WRITE(IW,150)
150 FORMAT(1H1,/,1X,30HPRINT CARD SPECIFICATION ERROR,/,1X,
143HINPUT AND FINAL OUTPUT DATA WILL BE PRINTED)
      IPRINT=4
C
C      PRINT INPUT DATA
C
152 CONTINUE
      REWIND IR
      IF((IPRINT.NE.1).AND.(IPRINT.LT.4))GO TO 158
      WRITE(IW,154)
154 FORMAT(1H1,/,1X,24H*** INPUT DATA CARDS **/,/)
      CALL IDPR(IR,IW)
      REWIND IR
158 CONTINUE
C
C      REORGANIZE INPUT DATA
C
C      CALL IDRO(IR,IW,JS)
C
C      PRINT REORGANIZED INPUT DATA
C
      IF(IPRINT.LT.6)GO TO 164
      WRITE(IW,160)
160 FORMAT(1H1,/,1X,30H*** REORGANIZED INPUT DATA **/,/)
      CALL IDPR(IR,IW)
C
C      READ INITIALIZING INSTRUCTIONS
C
164 CONTINUE
      ISYS=0
      DO 166 I=1,9
      DO 166 J=1,20
166 SHEAD(I,J)=HRRRR
168 CONTINUE
      READ(IR,170)CARD
170 FORMAT(20A4)
      IF(CARD(1).EQ.HPRIN)GO TO 168
      IF(CARD(1).NE.HCONT)GO TO 175
      CALL FILE(JN,LOCATE,LABEL)
      READ(IN)((SHEAD(I,J),J=1,20),I=1,9)
      CALL FILE(JN,NULL,LABEL)
      WRITE(IW,430)
      WRITE(IW,440)((SHEAD(I,J),J=1,20),I=1,9)
      GO TO 180
175 CONTINUE
      CALL FILE(JN,INSERT,MARK)
      CALL FILE(JQ,INSERT,MARK)
      GO TO 190
C
C      READ SYSTEM REFERENCE DATA
C
180 CONTINUE
      READ(IR,170)CARD
      IF(CARD(1).NE.HREFE)GO TO 190
183 CONTINUE
      READ(IR,170)CARD
      IF(CARD(1).EQ.HEND)GO TO 188
      DECODE(4,220,CARD(3))D1,NSYSNO,D2
      DO 185 I=1,20
185 SHEAD(NSYSNO,I)=CARD(I)
      GO TO 183
188 CONTINUE
      WRITE(IW,430)
      WRITE(IW,440)((SHEAD(I,J),J=1,20),I=1,9)
C
      KORG1131
      KORG1132
      KORG1133
      KORG1134
      KORG1135
      KORG1136
      KORG1137
      KORG1138
      KORG1139
      KORG1140
      KORG1141
      KORG1142
      KORG1143
      KORG1144
      KORG1145
      KORG1146
      KORG1147
      KORG1148
      KORG1149
      KORG1150
      KORG1151
      KORG1152
      KORG1153
      KORG1154
      KORG1155
      KORG1156
      KORG1157
      KORG1158
      KORG1159
      KORG1160
      KORG1161
      KORG1162
      KORG1163
      KORG1164
      KORG1165
      KORG1166
      KORG1167
      KORG1168
      KORG1169
      KORG1170
      KORG1171
      KORG1172
      KORG1173
      KORG1174
      KORG1175
      KORG1176
      KORG1177
      KORG1178
      KORG1179
      KORG1180
      KORG1181
      KORG1182
      KORG1183
      KORG1184
      KORG1185
      KORG1186
      KORG1187
      KORG1188
      KORG1189
      KORG1190
      KORG1191
      KORG1192
      KORG1193
      KORG1194
      KORG1195
      KORG1196
```

Figure 12. Subroutine KORG1 Program Listing (Continued)

# Contrails

```
C      READ SYSTEM SPECIFICATION CARD                                KORGI197
C      READ(1R,170)CARD                                           KORGI198
190  CONTINUE                                                     KORGI199
      IF(CARD(1).EQ.HSTOP)GO TO 400                                KORGI200
      IF(CARD(1).NE.HSYST)GO TO 260                              KORGI201
      IF(IPRINT.LT.6)GO TO 210                                   KORGI202
      CALL HPR(CARD,IW)                                          KORGI203
      WRITE(IW,200)MS1,MS2,MS3,MS4,MAXN,MAXM                    KORGI204
      1,NXM,NRM,NUM,NYM,MM,MP,MQ,MR,MB,NB,MS,MN,MTFB,MST,MT     KORGI205
200  FORMAT(1X,15(15,1X))                                       KORGI206
210  CONTINUE                                                     KORGI207
      DECODE(4,220,CARD(7:10),NSYSNO,D2)                        KORGI208
220  FORMAT(A2,11,A1)                                           KORGI209
      IF(NSYSNO.GT.NR)GO TO 260                                  KORGI210
      ISYS=1SYS+1                                               KORGI211
      IF(ISYS.GT.20)GO TO 260                                   KORGI212
      DO 24 I=1,5                                               KORGI213
      II=5+I                                                    KORGI214
240  SDSES(I)=CARD(II)                                          KORGI215
      DO 245 I=1,20                                             KORGI216
245  PHEAD(I)=SHEAD(NSYSNO,I)                                   KORGI217
      DO 25 I=1,20                                              KORGI218
      HEAD(I)=CARD(I)                                          KORGI219
      AHEAD(ISYS,I)=CARD(I)                                     KORGI220
250  SHEAD(NSYSNO,I)=CARD(I)                                   KORGI221
      NSYS(ISYS)=NSYSNO                                        KORGI222
      IF(IPRINT.LT.6)GO TO 256                                  KORGI223
      WRITE(IW,253)CARD                                         KORGI224
      WRITE(IW,253)HEAD                                         KORGI225
      WRITE(IW,253)PHEAD                                        KORGI226
      WRITE(IW,253)(SHEAD(NSYSNO,I),I=1,20)                   KORGI227
253  FORMAT(1X,20A4)                                           KORGI228
256  CONTINUE                                                  KORGI229
      READ(1R,170)SCODE                                         KORGI230
      IF(SCODE.EQ.HOLSA)GO TO 300                               KORGI231
      IF(SCODE.EQ.HDTRA)GO TO 320                               KORGI232
      IF(SCODE.EQ.HDOUA)GO TO 340                               KORGI233
      IF(SCODE.EQ.HDINT)GO TO 340                              KORGI234
      IF(SCODE.EQ.HDSIM)GO TO 360                              KORGI235
      IF(SCODE.EQ.HDCON)GO TO 380                              KORGI236
C      PRINT ERROR MESSEGE                                       KORGI237
C      PRINT ERROR MESSEGE                                       KORGI238
C      PRINT ERROR MESSEGE                                       KORGI239
260  CONTINUE                                                  KORGI240
      WRITE(IW,280)                                             KORGI241
280  FORMAT(1H1,/,1X,31HSYSTEM CARD SPECIFICATION ERROR)     KORGI242
      WRITE(IW,290)CARD                                         KORGI243
      WRITE(IW,290)SCODE                                        KORGI244
290  FORMAT(1X,20A4)                                           KORGI245
      WRITE(IW,295)NSYSNO,NR                                    KORGI246
295  FORMAT(1X,I2,1X,I2)                                       KORGI247
      STOP 111                                                 KORGI248
C      CALL OVERLAY LOADER TO LOAD REQUIRED PROGRAMS FOR EXECUTION KORGI249
C      CALL OVERLAY LOADER TO LOAD REQUIRED PROGRAMS FOR EXECUTION KORGI250
300  CONTINUE                                                  KORGI251
      CALL OVERLAY(4HKON1,1,0)                                  KORGI252
      GO TO 180                                                 KORGI253
320  CONTINUE                                                  KORGI254
      CALL OVERLAY(4HKON1,2,0)                                  KORGI255
      GO TO 180                                                 KORGI256
340  CONTINUE                                                  KORGI257
      CALL OVERLAY(4HKON1,3,0)                                  KORGI258
      GO TO 180                                                 KORGI259
360  CONTINUE                                                  KORGI260
      CALL OVERLAY(4HKON1,4,0)                                  KORGI261
      GO TO 180                                                 KORGI262
```

Figure 12. Subroutine KORGI Program Listing (Continued)

# Contrails

```
      CALL OVERLAY(4HKON),4.0)                                KORG1263
      GO TO 180                                              KORG1264
380  CONTINUE                                              KORG1265
      CALL OVERLAY(4HKON),5.0)                                KORG1266
      GO TO 180                                              KORG1267
C
C   WRITE SYSTEM LABELS ON NFILE FOR CONTINUATION RUNS     KORG1268
C
400  CONTINUE                                              KORG1269
      CALL FILE(JN,INSERT,LABEL)                             KORG1270
      WRITE(JN)((SHEAD(I,J),J=1,20),I=1,9)                  KORG1271
      CALL FILE(JN,INSERT,MARK)                              KORG1272
      WRITE(IW,430)                                          KORG1273
430  FORMAT(1H1,/,/,1X,34H*** REFERENCE OF SYSTEM LABELS **/,/) KORG1274
      WRITE(IW,440)((SHEAD(I,J),J=1,20),I=1,9)              KORG1275
440  FORMAT(/,1X,20A4,/)                                    KORG1276
      WRITE(IW,450)                                          KORG1277
450  FORMAT(1H1,/,/,1X,41H*** LIST OF SYSTEM LABELS CREATED IN THIS,
      18H RU! **/,/)                                        KORG1278
      WRITE(IW,440)((AHEAD(I,J),J=1,20),I=1,ISYS)          KORG1279
      STOP                                                  KORG1280
      END                                                  KORG1281
                                                    KORG1282
                                                    KORG1283
                                                    KORG1284
```

Figure 12. Subroutine KORG1 Program Listing (Concluded)

# Contrails

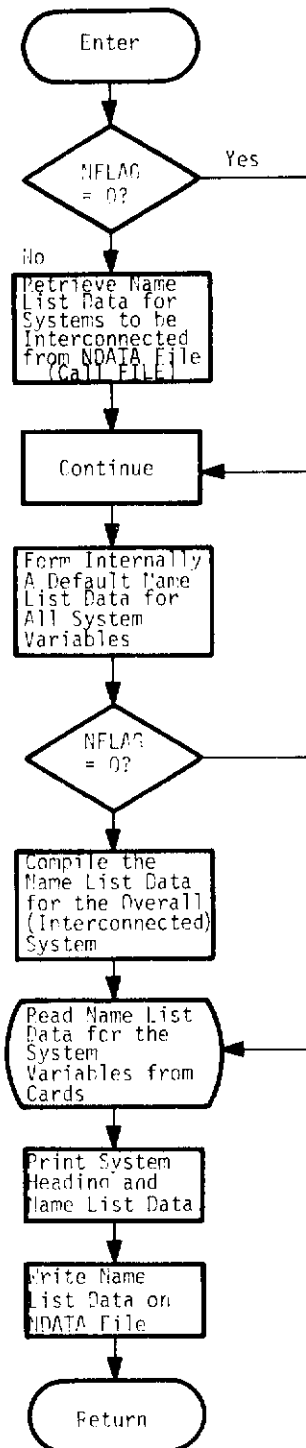


Figure 13. Subroutine NAMEL Flow Chart

# Contrails

```

SUBROUTINE NANEL(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,
10ESI,UNITI,DESSS,UNITSS,DES00,UNIT00,DESII,UNITII,NXX,NRR,NUU,
2NXM,NRM,NUM,NX,NR,NU,NFLAG,MB,KB,NR)      NANEL 2
C                                          NANEL 3
C                                          NANEL 4
C                                          NANEL 5
C PURPOSE - TO READ, PRINT AND UPDATE NANELIST DATA FOR SYSTEMS NANEL 6
C ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC NANEL 7
C DATE WRITTEN - 1975 NANEL 8
C                                          NANEL 9
C SUBPROGRAMS CALLED NANEL 10
C   DERUG NANEL 11
C   HPR NANEL 12
C   FILE NANEL 13
C                                          NANEL 14
C ARGUMENTS LIST NANEL 15
C   NNS      IN/OUT  NUMBER ARRAY FOR STATE NANEL 16
C   VNS      IN/OUT  VARIABLE NAME ARRAY FOR STATE NANEL 17
C   DESS     IN/OUT  DESCRIPTION ARRAY FOR STATE NANEL 18
C   UNITS    IN/OUT  UNIT ARRAY FOR STATE NANEL 19
C   NNO      IN/OUT  NUMBER ARRAY FOR OUTPUT NANEL 20
C   VNO      IN/OUT  VARIABLE NAME ARRAY FOR OUTPUT NANEL 21
C   DES0    IN/OUT  DESCRIPTION ARRAY FOR OUTPUT NANEL 22
C   UNIT0    IN/OUT  UNIT ARRAY FOR OUTPUT NANEL 23
C   NNI      IN/OUT  NUMBER ARRAY FOR INPUT NANEL 24
C   VNI      IN/OUT  VARIABLE NAME ARRAY FOR INPUT NANEL 25
C   DESI     IN/OUT  DESCRIPTION ARRAY FOR INPUT NANEL 26
C   UNITI    IN/OUT  UNIT ARRAY FOR INPUT NANEL 27
C   DESSS    STATE DESCRIPTION ARRAY FOR ALL SUBSYSTEMS NANEL 28
C   UNITSS   STATE UNIT ARRAY FOR ALL SUBSYSTEMS NANEL 29
C   DES00    OUTPUT DESCRIPTION ARRAY FOR ALL SUBSYSTEMS NANEL 30
C   UNIT00   OUTPUT UNIT ARRAY FOR ALL SUBSYSTEMS NANEL 31
C   DESII    INPUT DESCRIPTION ARRAY FOR ALL SUBSYSTEMS NANEL 32
C   UNITII   INPUT UNIT ARRAY FOR ALL SUBSYSTEMS NANEL 33
C   NXX      NO OF STATE ARRAY FOR ALL SUBSYSTEMS NANEL 34
C   NRR      NO OF OUTPUT ARRAY FOR ALL SUBSYSTEMS NANEL 35
C   NUU      NO OF INPUT ARRAY FOR ALL SUBSYSTEMS NANEL 36
C   NXM      INPUT   MAXIMUM NUMBER OF STATES NANEL 37
C   NRM      INPUT   MAXIMUM NUMBER OF OUTPUTS NANEL 38
C   NUM      INPUT   MAXIMUM NUMBER OF INPUTS NANEL 39
C   NX       INPUT   NUMBER OF STATES NANEL 40
C   NR       INPUT   NUMBER OF OUTPUTS NANEL 41
C   NU       INPUT   NUMBER OF INPUTS NANEL 42
C   NFLAG    INPUT   CONTROLS ENTRY POINT IN THE SUBROUTINE NANEL 43
C   MB       INPUT   MAXIMUM NO OF SYSTEMS FOR COMBINING NANEL 44
C   KB       OUTPUT  NO OF SYSTEMS FOR COMBINING NANEL 45
C   NB       INPUT   MAXIMUM SYSTEM NO - IMPLICIT MODEL NANEL 46
C                                          NANEL 47
C DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4) NANEL 48
C DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4) NANEL 49
C DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4) NANEL 50
C DIMENSION DESSS(NXM,10,MB),UNITSS(NXM,4,MB) NANEL 51
C DIMENSION DES00(NRM,10,MB),UNIT00(NRM,4,MB) NANEL 52
C DIMENSION DESII(NUM,10,MB),UNITII(NUM,4,MB) NANEL 53
C DIMENSION NXX(MB),NRR(MB),NUU(MB) NANEL 54
C DIMENSION CARD(20) NANEL 55
C DIMENSION VN(2),DES(10),UNIT(4) NANEL 56
C COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS NANEL 57
C COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20) NANEL 58
C I,PHEAD(20) NANEL 59
C DATA 4BBB,4MODE,4HLRFO,4LLBE/3H ,4HMODE,4HLFO,4HLL E/ NANEL 60
C DATA 4RROR/4HRROR/ NANEL 61
C DATA 4C,4HSTAT,4HOUTP,4HINPU,4HEND/1HC,4HSTAT,4HOUTP,4HINPU,4HEND / NANEL 62
C DATA 4XP,4RP,4UP,4P/2HX(,2HR(,2HU(,1H)/ NANEL 63
C DATA 4R,4E,4UT,4T,4BLANK/1H ,1HE,2HUT,1HT,4H / NANEL 64

```

Figure 14. Subroutine NANEL Program Listing

# Contrails

	REWIND JS	NAMEL 65
	NRI=0	NAMEL 66
	IF(NFLAG.EQ.0)GO TO 380	NAMEL 67
	IF(IPRINT.EQ.6)CALL DEBUG(1,4HNAME,4HL ,0,0,1W)	NAMEL 68
C		NAMEL 69
C	RETRIEVE NAME LIST DATA OF SUBSYSTEMS FOR COMBINING FROM FILE	NAMEL 70
C	NDATA	NAMEL 71
C		NAMEL 72
	DO 10 I=1,20	NAMEL 73
10	CARD(I)=HEAD(I)	NAMEL 74
	DO 120 N=1,KB	NAMEL 75
	NSY=NSYS(N)	NAMEL 76
	DO 40 I=1,20	NAMEL 77
40	HEAD(I)=SHEAD(NSY,I)	NAMEL 78
	CALL FILE(JN,LOCATF,HEAD)	NAMEL 79
	READ(IN)NXN,NRN,NUN,	NAMEL 80
	1 (NNS(I),(VNS(I,J),J=1,2),	NAMEL 81
	2 (DESS(I,J),J=1,10),(UNITS(I,J),J=1,4),I=1,NXN),	NAMEL 82
	3 (NNO(I),(VNO(I,J),J=1,2),	NAMEL 83
	4 (DESO(I,J),J=1,10),(UNITO(I,J),J=1,4),I=1,NRN),	NAMEL 84
	5 (NNI(I),(VNI(I,J),J=1,2),	NAMEL 85
	6 (DESI(I,J),J=1,10),(UNITI(I,J),J=1,4),I=1,NUN)	NAMEL 86
	IF(IPRINT.EQ.6)CALL DFRUG(2,4HNAME,4HL ,0,0,1W)	NAMEL 87
	NXX(N)=NXN	NAMEL 88
	NRR(N)=NRN	NAMEL 89
	NUU(N)=NUN	NAMEL 90
C		NAMEL 91
C	IF THE SUBSYSTEM IS AN IMPLICIT MODEL, THEN SET NRI=NRN	NAMEL 92
C		NAMEL 93
	IF(NSY.EQ.NR)NRI=NRN	NAMEL 94
	DO 60 I=1,NXN	NAMEL 95
	DO 50 J=1,10	NAMEL 96
50	DESS(I,J,N)=DESS(I,J)	NAMEL 97
	DO 60 J=1,4	NAMEL 98
60	UNITSS(I,J,N)=UNITS(I,J)	NAMEL 99
	DO 80 I=1,NRN	NAMEL 100
	DO 70 J=1,10	NAMEL 101
70	DESOO(I,J,N)=DESO(I,J)	NAMEL 102
	DO 80 J=1,4	NAMEL 103
80	UNITOO(I,J,N)=UNITO(I,J)	NAMEL 104
	DO 100 I=1,NUN	NAMEL 105
	DO 90 J=1,10	NAMEL 106
90	DESOI(I,J,N)=DESI(I,J)	NAMEL 107
	DO 100 J=1,4	NAMEL 108
100	UNITII(I,J,N)=UNITI(I,J)	NAMEL 109
120	CONTINUE	NAMEL 110
	DO 130 I=1,20	NAMEL 111
130	HEAD(I)=CARD(I)	NAMEL 112
380	CONTINUE	NAMEL 113
	IF(IPRINT.EQ.6)CALL DEBUG(3,4HNAME,4HL ,0,0,1W)	NAMEL 114
C		NAMEL 115
C	FORM A DEFAULT NAME LIST TABLE FOR THE SYSTEM	NAMEL 116
C		NAMEL 117
C	FORM NAME LIST FOR STATES	NAMEL 118
C		NAMEL 119
	DO 500 II=1,NX	NAMEL 120
	NNS(II)=II	NAMEL 121
	ENCODE(4,420,VNS(II,1))HXP,II	NAMEL 122
420	FORMAT(A2,I2)	NAMEL 123
	VNS(II,2)=HP	NAMEL 124
	UNITS(II,1)=HBLANK	NAMEL 125
	UNITS(II,2)=HBLANK	NAMEL 126
	UNITS(II,3)=HBLANK	NAMEL 127
	UNITS(II,4)=HBLANK	NAMEL 128
	J=0	NAMEL 129
	JFLAG=0	NAMEL 130

Figure 14. Subroutine NAMEL Program Listing (Continued)

# Contrails

<pre> 440 CONTINUE    IF (J.GE.5) GO TO 460    IF (JFLAG.EQ.1) GO TO 460    J=J+1    IF (SDFS(J).EQ.HBLANK) JFLAG=JFLAG+1    DESS(II,J)=SDES(J)    GO TO 440 460 CONTINUE    J=J+1    DESS(II,J)=HSTAT    J=J+1    ENCODE(4,476,DESS(II,J))HE,II,HB 470 FORMAT(A1,I2,A1) 480 CONTINUE    J=J+1    IF (J.GT.10) GO TO 500    DESS(II,J)=HBLANK    GO TO 480 500 CONTINUE    IF (IPRINT.EQ.6) CALL DFBUG(4,4HNAME,4HL  .0,0,1W) C C   FORM NAME LIST FOR OUTPUTS C    NRJ=NR-NRI    DO 700 II=1,NR    NNO(II)=II    ENCODE(4,420,VNO(II,1))HRP,II    VNO(II,2)=HP    UNITO(II,1)=HBLANK    UNITO(II,2)=HBLANK    UNITO(II,3)=HBLANK    UNITO(II,4)=HBLANK    J=0    JFLAG=0 C C   FORM NAME LIST FOR THE IMPLICIT MODEL ERROR RESPONSES C    IF (II.LE.NRJ) GO TO 640    NJ=II-NRJ    DECODE(4,610,DESO(NJ,1,KB))IT1,JT1 610 FORMAT(A3,A1)    ENCODE(4,615,DESO(II,1))HE,IT1 615 FORMAT(A1,A3)    ENCODE(4,615,DESO(II,2))JT1,HRRB    DESO(II,3)=HMODE    DESO(II,4)=HLRFO    DESO(II,5)=HLLRE    DESO(II,6)=HRROR    DESO(II,7)=HBLANK    DESO(II,8)=HBLANK    DESO(II,9)=HBLANK    DESO(II,10)=HBLANK    DO 62 J=1,4 620 UNITO(II,J)=UNITO(NJ,J,KR)    GO TO 700 640 CONTINUE    IF (J.GE.5) GO TO 660    IF (JFLAG.EQ.1) GO TO 660    J=J+1    IF (SDFS(J).EQ.HBLANK) JFLAG=JFLAG+1    DESO(II,J)=SDES(J)    GO TO 640 660 CONTINUE    J=J+1    DESO(II,J)=HOUTP    J=J+1 </pre>	<pre> NAMEL131 NAMEL132 NAMEL133 NAMEL134 NAMEL135 NAMEL136 NAMEL137 NAMEL138 NAMEL139 NAMEL140 NAMEL141 NAMEL142 NAMEL143 NAMEL144 NAMEL145 NAMEL146 NAMEL147 NAMEL148 NAMEL149 NAMEL150 NAMEL151 NAMEL152 NAMEL153 NAMEL154 NAMEL155 NAMEL156 NAMEL157 NAMEL158 NAMEL159 NAMEL160 NAMEL161 NAMEL162 NAMEL163 NAMEL164 NAMEL165 NAMEL166 NAMEL167 NAMEL168 NAMEL169 NAMEL170 NAMEL171 NAMEL172 NAMEL173 NAMEL174 NAMEL175 NAMEL176 NAMEL177 NAMEL178 NAMEL179 NAMEL180 NAMEL181 NAMEL182 NAMEL183 NAMEL184 NAMEL185 NAMEL186 NAMEL187 NAMEL188 NAMEL189 NAMEL190 NAMEL191 NAMEL192 NAMEL193 NAMEL194 NAMEL195 NAMEL196 </pre>
---	--

Figure 14. Subroutine NAMEL Program Listing (Continued)



# Contrails

	ENCODE(4,420,DESO((I,J))HUT,I)		NAMEL197
680	CONTINUE		NAMEL198
	J=J+1		NAMEL199
	IF(J.GT.10)GO TO 700		NAMEL200
	DESO(I,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))HUP,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 840		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))HT,II,HB		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,K9		NAMEL244
	NXXK=NXX(K)		NAMEL245
	DO 1040 I=1,NXXK		NAMEL246
	II=II+1		NAMEL247
	NNS(II)=II		NAMEL248
	ENCODE(4,420,VNS(II,1))HXP,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(JS,160)CARD		NAMEL260
	IF(CARD(1).NE.HOUTP)GO TO 1320		NAMEL261
1050	CONTINUE		NAMEL262

Figure 14. Subroutine NAMEL Program Listing (Continued)

# Contrails

	READ(JS,1060)II,K,I	NAMEL263
1060	FORMAT(3I2)	NAMEL264
	IF(II.EQ.-1)GO TO 1110	NAMEL265
	NNO(II)=II	NAMEL266
	ENCODE(4,420,VNO(II,1))HRP,II	NAMEL267
	VNO(II,2)=HP	NAMEL268
	DO 1040 J=1,10	NAMEL269
1080	DESO(II,J)=DESOO(I,J,K)	NAMEL270
	DO 1100 J=1,4	NAMEL271
1100	UNITO(II,J)=UNITOO(I,J,K)	NAMEL272
	GO TO 1050	NAMEL273
1110	CONTINUE	NAMEL274
	IF(IPRINT.EQ.6)CALL DEBUG(7,4HNAME,4HL .0,0,1W)	NAMEL275
C		NAMEL276
C	READ NAME LIST DATA FOR INPUTS OBTAINABLE FROM	NAMEL277
C	INTERCONNECTION EQUATIONS WRITTEN ON SCRATCH FILE JS	NAMEL278
C	BY SUBROUTINE SIMK	NAMEL279
C		NAMEL280
	READ(JS,160)CARD	NAMEL281
	IF(CARD(1).NE.HINPI)GO TO 1320	NAMEL282
1120	CONTINUE	NAMEL283
	READ(JS,1060)II,K,I	NAMEL284
	IF(II.EQ.-1)GO TO 1170	NAMEL285
	NNI(II)=II	NAMEL286
	ENCODE(4,420,VNI(II,1))HUP,II	NAMEL287
	VNI(II,2)=HP	NAMEL288
	DO 1140 J=1,10	NAMEL289
1140	DESI(II,J)=DESI(II,J,K)	NAMEL290
	DO 1160 J=1,4	NAMEL291
1160	UNITI(II,J)=UNITII(I,J,K)	NAMEL292
	GO TO 1120	NAMEL293
1170	CONTINUE	NAMEL294
	READ(JS,160)CARD	NAMEL295
	IF(CARD(1).NE.HEND)GO TO 1320	NAMEL296
C		NAMEL297
C	READ NAME LIST DATA FROM CARDS	NAMEL298
C		NAMEL299
1220	CONTINUE	NAMEL300
	IF(IPRINT.EQ.6)CALL DEBUG(8,4HNAME,4HL .0,0,1W)	NAMEL301
	READ(IP,160)CARD	NAMEL302
160	FORMAT(20A4)	NAMEL303
	IF(CARD(1).EQ.HEND)GO TO 1340	NAMEL304
	IF(CARD(1).EQ.HSTAT)GO TO 1240	NAMEL305
	IF(CARD(1).EQ.HOUTP)GO TO 1260	NAMEL306
	IF(CARD(1).EQ.HINPI)GO TO 1300	NAMEL307
	GO TO 200	NAMEL308
C		NAMEL309
C	READ NAME LIST DATA FOR STATES	NAMEL310
C		NAMEL311
1240	CONTINUE	NAMEL312
	READ(IP,280)NNNN,(VN(J),J=1,2),(DES(J),J=1,10),(UNIT(J),J=1,4)	NAMEL313
280	FORMAT(I2,6X,2A4,4X,10A4,4X,4A4)	NAMEL314
	IF(NNNN.EQ.-1)GO TO 1220	NAMEL315
	NNS(NNNN)=NNNN	NAMEL316
	DO 1245 J=1,2	NAMEL317
1245	VNS(NNNN,J)=VN(J)	NAMEL318
	DO 1250 J=1,10	NAMEL319
1250	DESS(NNNN,J)=DES(J)	NAMEL320
	DO 1255 J=1,4	NAMEL321
1255	UNITS(NNNN,J)=UNIT(J)	NAMEL322
	GO TO 1240	NAMEL323
C		NAMEL324
C	READ NAME LIST DATA FOR OUTPUTS	NAMEL325
C		NAMEL326
1260	CONTINUE	NAMEL327
	READ(IP,280)NNNN,(VN(J),J=1,2),(DES(J),J=1,10),(UNIT(J),J=1,4)	NAMEL328

Figure 14. Subroutine NAMEL Program Listing (Continued)

# Contrails

```
      IF(NNNN.EQ.-1)GO TO 1220
      NNO(N,NN)=NNNN
      DO 1265 J=1,2
1265  VNO(N,NN,J)=VN(J)
      DO 1270 J=1,10
1270  DESO(N,NN,J)=DES(J)
      DO 1275 J=1,4
1275  UNITO(N,NN,J)=UNIT(J)
      GO TO 1260
C
C   READ NAME LIST DATA FOR INPUTS
C
1300  CONTINUE
      READ(1R,280)NNMN,(VN(J),J=1,2),(DES(J),J=1,10),(UNIT(J),J=1,4)
      IF(NNNN.EQ.-1)GO TO 1220
      NNI(N,NN)=NNNN
      DO 1285 J=1,2
1285  VNI(N,NN,J)=VN(J)
      DO 1290 J=1,10
1290  DESI(N,NN,J)=DES(J)
      DO 1295 J=1,4
1295  UNITI(N,NN,J)=UNIT(J)
      GO TO 1300
1340  CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(9,4HNAME,4HL  .0,0,IW)
C
C   PRINT HEADING AND NAME LIST DATA
C
      IF(IPRINT.LT.2)GO TO 1540
      CALL HPR(HEAD,IW)
      WRITE(9,1360)NX,NR,NU
1360  FORMAT(//,1X,18HNUMBER OF STATES =,I2,//,1X,
118HNUMBER OF OUTPUTS=,I2,//,1X,18HNUMBER OF INPUTS =,I2,//)
      WRITE(IW,1380)
1380  FORMAT(//,20X,23H*** NAME LIST TABLE ***,/)
      WRITE(IW,1400)
1400  FORMAT(/,1X,8HVARIABLE,6H NAME ,6X,13H DESCRIPTION ,
131X,6H UNIT ,/)
      IF(IPRINT.EQ.6)CALL DEBUG(10,4HNAME,4HL  .0,0,IW)
C
C   PRINT NAME LIST DATA FOR STATES
C
      WRITE(IW,1460)
1460  FORMAT(/,1X,6HSTATE ,/)
      WRITE(IW,1480)(NNS(I),(VNS(I,J),J=1,2),(DESS(I,J),J=1,10),
1
      (UNITS(I,J),J=1,4),I=1,NX)
1480  FORMAT(1X,12,6X,2A4,4X,10A4,4X,4A6)
C
C   PRINT NAME LIST DATA FOR OUTPUTS
C
      WRITE(IW,1500)
1500  FORMAT(/,1X,6HOUTPUT,/)
      WRITE(IW,1480)(NNO(I),(VNO(I,J),J=1,2),(DESO(I,J),J=1,10),
1
      (UNITO(I,J),J=1,4),I=1,NR)
C
C   PRINT NAME LIST DATA FOR INPUTS
C
      WRITE(IW,1520)
1520  FORMAT(/,1X,6HINPUT ,/)
      WRITE(IW,1480)(NNI(I),(VNI(I,J),J=1,2),(DESI(I,J),J=1,10),
1
      (UNITI(I,J),J=1,4),I=1,NJ)
1540  CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(11,4HNAME,4HL  .0,0,IW)
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)

# Contrails

```
CALL FILE(JN,INSERT,HEAD)
WRITE(JN)NX,NR,NU,
1      (NNS(I),(VNS(I,J),J=1,2),
2      (DESS(I,J),J=1,10),(UNITS(I,J),J=1,4),I=1,NX),
3      (NNO(I),(VNO(I,J),J=1,2),
4      (DESO(I,J),J=1,10),(UNITO(I,J),J=1,4),I=1,NR),
5      (NNI(I),(VNI(I,J),J=1,2),
6      (DESI(I,J),J=1,10),(UNITI(I,J),J=1,4),I=1,NU)
CALL FILE(JN,INSERT,MARK)
IF(IPRINT.EQ.6)CALL DEBUG(17,4HNAME,4HL  ,0,0,IW)
RETURN
C
C PRINT ERROR MESSEGE
C
200 CONTINUE
WRITE(IW,220)
220 FORMAT(1H1,/,1X,37HDATA CONTROL CARD SPECIFICATION ERROR)
STOP 111
1320 CONTINUE
WRITE(IW,1330)
1330 FORMAT(1H1,/,1X,36HERROR IN DATA PROVIDED BY SIMK)
STOP 111
END
```

NAMEL395  
NAMEL396  
NAMEL397  
NAMEL398  
NAMEL399  
NAMEL400  
NAMEL401  
NAMEL402  
NAMEL403  
NAMEL404  
NAMEL405  
NAMEL406  
NAMEL407  
NAMEL408  
NAMEL409  
NAMEL410  
NAMEL411  
NAMEL412  
NAMEL413  
NAMEL414  
NAMEL415  
NAMEL416  
NAMEL417

Figure 14. Subroutine NAMEL Program Listing (Concluded)

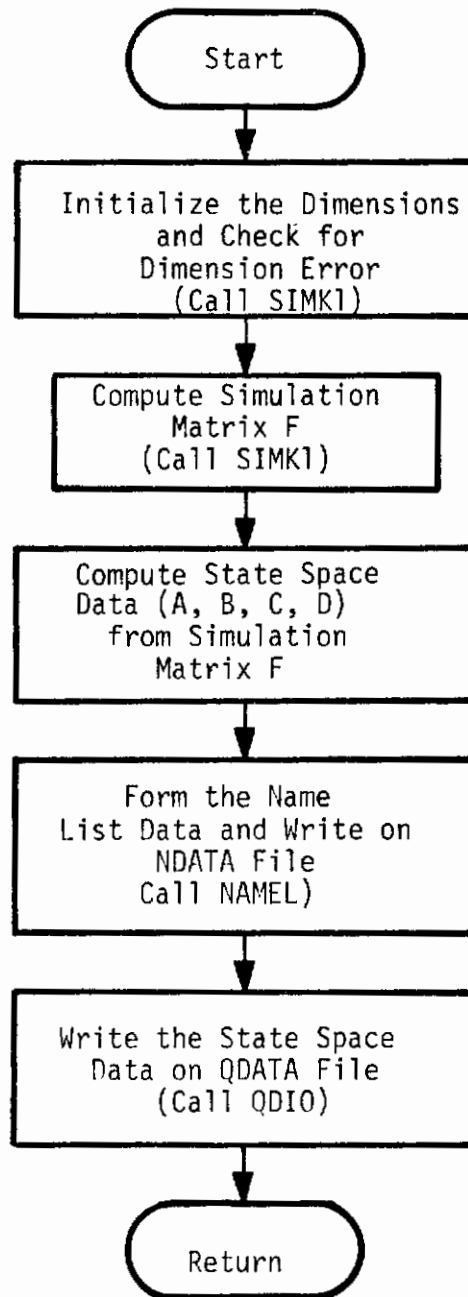


Figure 15. Subroutine STAMK1 Flow Chart

```
SUBROUTINE STAMK1(V,W,F,U,A,B,C,D,           STAMK1 2
INNS,V,S,DESS,UNITS,NNO,VNO,DESO,UNITO,NNT,VNT,DESI,UNITI,  STAMK1 3
2MAXN,MAXM,NXM,NRM,NUM,NYM,MR,MS1,MS2,MS3,MS4,NR)           STAMK1 4
C                                                                 STAMK1 5
C                                                                 STAMK1 6
C   PURPOSE - TO OBTAIN STATE SPACE MODEL OF THE VEHICLE   STAMK1 7
C   DESCRIBED BY SIMULATOR DECK DATA FROM LSA PROGRAM     STAMK1 8
C   ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC   STAMK1 9
C   DATE WRITTEN - 1975                                     STAMK110
C                                                                 STAMK111
C   SUBPROGRAMS CALLED                                       STAMK112
C     DEBUG                                                 STAMK113
C     DEPRMS                                               STAMK114
C     MPDS                                                 STAMK115
C     QDIO                                                 STAMK116
C     TDINVR                                              STAMK117
C     DEPRM                                               STAMK118
C     HPP                                                 STAMK119
C     NAMEL                                               STAMK120
C     SIMK1                                               STAMK121
C                                                                 STAMK122
C   ARGUMENTS LIST                                          STAMK123
C     V                                                    STAMK124
C     W                                                    STAMK125
C     F                                                    STAMK126
C     U                                                    STAMK127
C     A            IN/OUT  STATE TRANSITION MATRIX         STAMK128
C     B            IN/OUT  CONTROL INPUT MATRIX            STAMK129
C     C            IN/OUT  STATE OUTPUT MATRIX             STAMK130
C     D            IN/OUT  CONTROL OUTPUT MATRIX          STAMK131
C     NNS          IN/OUT  NUMBER ARRAY FOR STATE         STAMK132
C     VNS          IN/OUT  VARIABLE NAME ARRAY FOR STATE  STAMK133
C     DESS        IN/OUT  DESCRIPTION ARRAY FOR STATE     STAMK134
C     UNITS       IN/OUT  UNIT ARRAY FOR STATE            STAMK135
C     NNO         IN/OUT  NUMBER ARRAY FOR OUTPUT         STAMK136
C     VNO         IN/OUT  VARIABLE NAME ARRAY FOR OUTPUT  STAMK137
C     DESO        IN/OUT  DESCRIPTION ARRAY FOR OUTPUT    STAMK138
C     UNITO       IN/OUT  UNIT ARRAY FOR OUTPUT           STAMK139
C     NNT         IN/OUT  NUMBER ARRAY FOR INPUT          STAMK140
C     VNT         IN/OUT  VARIABLE NAME ARRAY FOR INPUT   STAMK141
C     DESI        IN/OUT  DESCRIPTION ARRAY FOR INPUT     STAMK142
C     UNITI       IN/OUT  UNIT ARRAY FOR INPUT            STAMK143
C     MAXN        INPUT   MAXIMUM ROW DIMENSION FOR SIMULA MATRIX F STAMK144
C     MAXM        INPUT   MAXIMUM COLUMN DIMENSION FOR SIMU MATRIX F STAMK145
C     NXM         INPUT   MAXIMUM NUMBER OF STATES        STAMK146
C     NRM         INPUT   MAXIMUM NUMBER OF OUTPUTS       STAMK147
C     NUM         INPUT   MAXIMUM NUMBER OF INPUTS        STAMK148
C     NYM         INPUT   MAXIMUM DIMENSION FOR INTERCONN EQUATIONS STAMK149
C     MB          INPUT   MAXIMUM NO OF SUBSYSTEMS FOR COMBINING STAMK150
C     MS1         INPUT   MAXIMUM DIMENSION FOR SCRATCH ARRAY S1 STAMK151
C     MS2         INPUT   MAXIMUM DIMENSION FOR SCRATCH ARRAY S2 STAMK152
C     MS3         INPUT   MAXIMUM DIMENSION FOR SCRATCH ARRAY S3 STAMK153
C     MS4         INPUT   MAXIMUM DIMENSION FOR SCRATCH ARRAY S4 STAMK154
C     NR          INPUT   MAXIMUM SYSTEM NO - IMPLICIT MODEL STAMK155
C                                                                 STAMK156
C   COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS STAMK157
C   COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20) STAMK158
C   I,PHED(20)                                             STAMK159
C   DIMENSION V(MAXN),W(MAXM),F(MAXN,MAXM)                STAMK160
C   DIMENSION U(NUM)                                       STAMK161
C   DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM) STAMK162
C   DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4) STAMK163
C   DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4) STAMK164
C   DIMENSION NNT(NUM),VNT(NUM,2),DESI(NUM,10),UNITI(NUM,4)
```

Figure 16. Subroutine STAMK1 Program Listing

# Contrails

```

COMMON /SCI/ SI(1)
C DIMENSION DESSS(NXM,10,MB),UNITSS(NXM,4,MR)
C DIMENSION DES00(NRM,10,MB),UNIT00(NRM,4,MR)
C DIMENSION DES11(NUM,10,MB),UNIT11(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*MB*10
L5=L4+NRM*MR*4 $ L6=L5+NUM*MB*10 $ L7=L6+NUM*MR*4
L8=L7+MB $ L9=L8+MR $ L10=L9+MR
IF(L1.GT.MS1)
CALL DERRM(L10,MS2,MS3,MS4,MS1,MS2,MS3,MS4,1,0,4HSTAM,4HK1 ,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 HPR(HEAD,1W)
C
C INITIALIZING CALL TO SUBROUTINE SIMK1
C
INIT=
NX=0 $ NY=0 $ NP=0 $ NU=0
N1=1 $ N2=N1+NX $ N3=N2+NY $ N4=N3+NX
CALL SIMK1(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+NY+NR
IF((NX.GT.NXM).OR.(NR.GT.NRM).OR.(NU.GT.NUM).OR.(NY.GT.NYM))
CALL 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
W(J)=0.
DO 501 J=1,M
W(J)=1.
CALL SIMK1(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)=0.6
DO 501 I=1,N
F(I,J)=V(I)
501
C
C COMPUTE THE SIMULATION MATRIX
C
NV=NX+NY
IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,N,M,T,4HSTMI)
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(IISOL,IDSOL,NV,-M,F,MAXN,KDIM,DET)
IR=NV+1
IE=NV+NR
JB=JB
JE=M
DO 53 I=IR,IE
DO 53 J=JB,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
530 CONTINUE
IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,N,M,T,4HSTMI)

```

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-J1+1	STAMK142
6001	A(I,J)=F(I,J)	STAMK143
	DO 6002 I=1,NX	STAMK144
	DO 6002 J=J3,J4	STAMK145
	JJ=J-J3+1	STAMK146
6002	B(I,J)=F(I,J)	STAMK147
	DO 6003 I=I1,I2	STAMK148
	I1=I-I1+1	STAMK149
	DO 6003 J=J1,J2	STAMK150
	JJ=J-J1+1	STAMK151
6003	C(I1,JJ)=F(I,J)	STAMK152
	DO 6004 I=I1,I2	STAMK153
	I1=I-I1+1	STAMK154
	DO 6004 J=J3,J4	STAMK155
	JJ=J-J3+1	STAMK156
6004	D(I1,JJ)=F(I,J)	STAMK157
	IF(IPRINT.EQ.6)CALL DEBUG(4,4HSTAM,4HX1 .1,0,IW)	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),NXM,NRM,NUM,NX,NR,NU,NFLAG,NB,KB,NB)	STAMK165
	IF(IPRINT.EQ.6)CALL DEBUG(5,4HSTAM,4HX1 .1,0,IW)	STAMK166
C		STAMK167
C	WRITE QUADRUPLE DATA ON FILE QDATA	STAMK168
C		STAMK169
	IQ=0	STAMK170
	MFLAG=2	STAMK171
	NXA=NX \$ NRA=NR \$ NUA=NU	STAMK172
	CALL QDIO(A,B,C,D,A,NX,NR,NI,NXM,NRM,NUM,NXA,NRA,NUA,	STAMK173
	INR1,NR2,NR3,NU),NU2,NU3,T,IQ,IPRINT,IW,JQ,HEAD,MARK,	STAMK174
	ZLOCATF=NULL,INSERT,MFLAG)	STAMK175
	IF(IPRINT.EQ.6)CALL DEBUG(6,4HSTAM,4HX1 .1,0,IW)	STAMK176
	RETURN	STAMK177
	END	STAMK178

Figure 16. Subroutine STAMK1 Program Listing (Concluded)



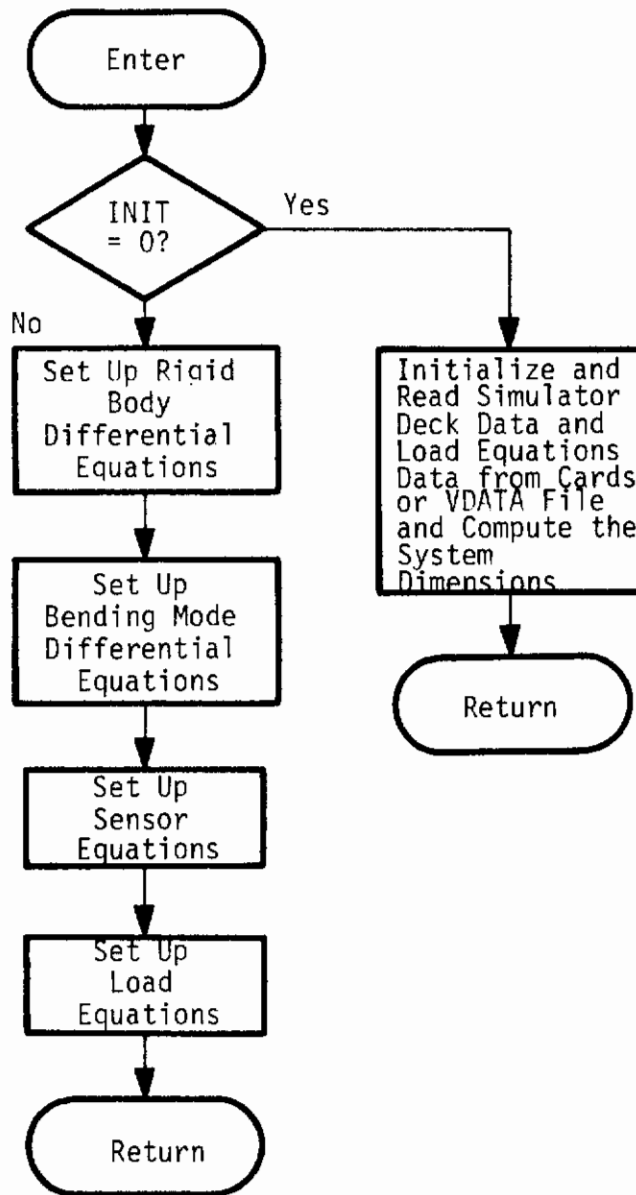


Figure 17. Subroutine SIMK1 Flow Chart

# Contrails

```

SUBROUTINE SIMK1(XDOT,Y,X,U,XDOTL,YL,RL,NX,NY,NR,NU,INIT,T,
1MS1,MS2,MS3,MS4)
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   MPHS1
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,YW,IPRINT,INSERT,LOCATE,NUll,MARK(20),JN,JQ,JS
REAL LVP0,LVP1,LR0,LR1,LUF0,LUE1,LUE2,LDLS0,LDLS1,LDLS2
REAL LVG0,LVG1,LVG0,LVG1,LWG0,LWG1
COMMON /SC2/ VPVP0(6,6),VPVP1(6,6),VPR0(6,3),VPR1(6,3)
1.VVPF0(6,30),VVPF1(6,30),VVPF2(6,30),VPDELS0(6,3)
2.VPDELS1(6,3),VPDELS2(6,3),VPUG0(6,3),VPUG1(6,3)
3.VPVG0(6,3),VPVG1(6,3),VPWG0(6,3),VPWG1(6,3)
4.RVP0(3,6),RVP1(3,6),PRO(3,3),PRI(3,3)
5.UFVPO(30,6),UFVPI(30,6),UERO(30,3)
6.UFER1(30,3),UFUE0(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.UEUG0(30,3),UEVG1(30,3),TVP0(9,6),TVP1(9,6),TRO(9,3)
A.TR1(9,3),TUF0(9,30),TUE1(9,30),TUF2(9,30)
B.TDELS0(9,3),TDELS1(9,3),TDFLS2(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.LR0(15,3),LR1(15,3),LUF0(15,30),LUE1(15,30),LUE2(15,30)
E.LDELS0(15,3),LDELS1(15,3),LDFLS2(15,3),LVG0(15,3),LVG1(15,3)
F.LVG0(15,3),LVG1(15,3),LWG0(15,3),LWG1(15,3)
G.BANDING(1,3),UNITY(30,30)
DIMENSION JHEAD(120),TDRM(120),IDCM(120)
DIMENSION ICARD(8),IHFA0(8)
DIMENSION SC(1)
EQUIVALENCE (SC(1),VPVP0(1,1))
IF (IPRINT.EQ.6) CALL DEBUG(1,4HSIMK,4H1 ,1,0,IW)
IF (INIT.NE.0) GO TO 150
C
C INITIALIZE AND SET MAX DIMENSIONS FOR SIMULATOR MATRIX DATA
C
C IEND=10HEND
C NXVP=0 % NXR=0 % NXUE=0

```

Figure 18. Subroutine SIMK1 Program Listing

# Contrails

```
NUC1=0 $ NUC2=0 $ NUC3=0 SIMK1 66
NUG0=0 $ NUG1=0 $ NUG50=0 $ NUGS1=0 SIMK1 67
NVG0=0 $ NVG1=0 $ NVG50=0 $ NVGS1=0 SIMK1 68
NWG0=0 $ NWG1=0 $ NWG50=0 $ NWGS1=0 SIMK1 69
NRT=0 $ NRL=0 $ NB=1 $ NL=3 SIMK1 70
NXVPM=6 $ NXRM=3 $ NXUEM=30 SIMK1 71
NUCM=3 $ NUGM=3 SIMK1 72
NRTM=9 $ NRLM=15 $ NBM=1 $ NL=3 SIMK1 73
C SIMK1 74
C DEFINE NAMES FOR SIMULATOR MATRIX DATA SIMK1 75
C SIMK1 76
C RIGID BODY VELOCITY COEFF MATRIX NAMES SIMK1 77
C SIMK1 78
JHEAD(1)=10HVP/VP0 $ JHEAD(2)=10HVP/VP1 SIMK1 79
JHEAD(3)=10HVP/R0 $ JHEAD(4)=10HVP/R1 SIMK1 80
JHEAD(5)=10HVP/UE0 $ JHEAD(6)=10HVP/UE1 SIMK1 81
JHEAD(7)=10HVP/UE2 $ JHEAD(8)=10HVP/DELS0 SIMK1 82
JHEAD(9)=10HVP/DELS1 $ JHEAD(10)=10HVP/DELS2 SIMK1 83
JHEAD(11)=10HVP/UG0 $ JHEAD(12)=10HVP/UG1 SIMK1 84
JHEAD(13)=10HVP/VG0 $ JHEAD(14)=10HVP/VG1 SIMK1 85
JHEAD(15)=10HVP/WG0 $ JHEAD(16)=10HVP/WG1 SIMK1 86
JHEAD(17)=10HVP/UG50 $ JHEAD(18)=10HVP/UGS1 SIMK1 87
JHEAD(19)=10HVP/VG50 $ JHEAD(20)=10HVP/VGS1 SIMK1 88
JHEAD(21)=10HVP/WG50 $ JHEAD(22)=10HVP/WGS1 SIMK1 89
C SIMK1 90
C RIGID BODY ATTITUDE COEFF MATRIX NAMES SIMK1 91
C SIMK1 92
JHEAD(23)=10HR/VP0 $ JHEAD(24)=10HR/VP1 SIMK1 93
JHEAD(25)=10HR/R0 $ JHEAD(26)=10HR/R1 SIMK1 94
C SIMK1 95
C RENDING MODE COEFF MATRIX NAMES SIMK1 96
C SIMK1 97
JHEAD(45)=10HUE/VP0 $ JHEAD(46)=10HUE/VP1 SIMK1 98
JHEAD(47)=10HUF/R0 $ JHEAD(48)=10HUE/R1 SIMK1 99
JHEAD(49)=10HUF/UE0 $ JHEAD(50)=10HUE/UE1 SIMK1 100
JHEAD(51)=10HUF/UE2 $ JHEAD(52)=10HUE/DELS0 SIMK1 101
JHEAD(53)=10HUF/DELS1 $ JHEAD(54)=10HUE/DELS2 SIMK1 102
JHEAD(55)=10HUF/UG0 $ JHEAD(56)=10HUE/UG1 SIMK1 103
JHEAD(57)=10HUF/VG0 $ JHEAD(58)=10HUE/VG1 SIMK1 104
JHEAD(59)=10HUF/WG0 $ JHEAD(60)=10HUE/WG1 SIMK1 105
JHEAD(61)=10HUF/UG50 $ JHEAD(62)=10HUE/UGS1 SIMK1 106
JHEAD(63)=10HUE/VG50 $ JHEAD(64)=10HUE/VGS1 SIMK1 107
JHEAD(65)=10HUF/WG50 $ JHEAD(66)=10HUE/WGS1 SIMK1 108
C SIMK1 109
C SENSOR COEFF MATRIX NAMES SIMK1 110
C SIMK1 111
JHEAD(67)=10HT/VP0 $ JHEAD(68)=10HT/VP1 SIMK1 112
JHEAD(69)=10HT/R0 $ JHEAD(70)=10HT/R1 SIMK1 113
JHEAD(71)=10HT/UE0 $ JHEAD(72)=10HT/UE1 SIMK1 114
JHEAD(73)=10HT/UE2 $ JHEAD(74)=10HT/DELS0 SIMK1 115
JHEAD(75)=10HT/DELS1 $ JHEAD(76)=10HT/DELS2 SIMK1 116
JHEAD(77)=10HT/UG0 $ JHEAD(78)=10HT/UG1 SIMK1 117
JHEAD(79)=10HT/VG0 $ JHEAD(80)=10HT/VG1 SIMK1 118
JHEAD(81)=10HT/WG0 $ JHEAD(82)=10HT/WG1 SIMK1 119
JHEAD(83)=10HT/UG50 $ JHEAD(84)=10HT/UGS1 SIMK1 120
JHEAD(85)=10HT/VG50 $ JHEAD(86)=10HT/VGS1 SIMK1 121
JHEAD(87)=10HT/WG50 $ JHEAD(88)=10HT/WGS1 SIMK1 122
C SIMK1 123
C LOADS COEFF MATRIX NAMES SIMK1 124
C SIMK1 125
JHEAD(89)=10HL/VP0 $ JHEAD(90)=10HL/VP1 SIMK1 126
JHEAD(91)=10HL/R0 $ JHEAD(92)=10HL/R1 SIMK1 127
JHEAD(93)=10HL/UE0 $ JHEAD(94)=10HL/UE1 SIMK1 128
JHEAD(95)=10HL/UE2 $ JHEAD(96)=10HL/DELS0 SIMK1 129
JHEAD(97)=10HL/DELS1 $ JHEAD(98)=10HL/DELS2 SIMK1 130
JHEAD(99)=10HL/UG0 $ JHEAD(100)=10HL/UG1 SIMK1 131
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

# Contrails

```
JHEAD(101)=10HL/VG0      $ JHFAD(102)=10HL/VG1      SIMK1132
JHEAD(103)=10HL/WG0      $ JHFAD(104)=10HL/WG1      SIMK1133
JHFAD(105)=10HL/UGS0     $ JHFAD(106)=10HL/UGS1      SIMK1134
JHEAD(107)=10HL/VGS0     $ JHFAD(108)=10HL/VGS1      SIMK1135
JHFAD(109)=10HL/WGS0     $ JHFAD(110)=10HL/WGS1      SIMK1136
C                                                                    SIMK1137
C MISCELLANEOUS MATRIX NAMES                                       SIMK1138
C                                                                    SIMK1139
C JHEAD(111)=10H(BANDING) $ JHFAD(112)=10H*FINISHED*           SIMK1140
C                                                                    SIMK1141
C SET UP MAX ROW AND COL DIMENSIONS FOR SIMULATOR MATRIX DATA   SIMK1142
C                                                                    SIMK1143
C DO 4 I=1,16                                                       SIMK1144
C I1=I                                                              SIMK1145
C I2=16*I                                                           SIMK1146
C I3=32*I                                                           SIMK1147
C I4=48*I                                                           SIMK1148
C I5=64*I                                                           SIMK1149
C IDRM(I1)=NXVPM $ IDRM(I2)=NXRM $ IDRM(I3)=NXUEM                SIMK1150
C IDRM(I4)=NRTM $ IDRM(I5)=NRLM                                    SIMK1151
4 CONTINUE                                                         SIMK1152
C DO 6 I=1,5                                                        SIMK1153
C J=(16*I-16)                                                       SIMK1154
C IDCM(J+1)=NXVPM $ IDCM(J+2)=NXVPM                               SIMK1155
C IDCM(J+3)=NXRM $ IDCM(J+4)=NXRM                                  SIMK1156
C IDCM(J+5)=NXUEM $ IDCM(J+6)=NXUEM $ IDCM(J+7)=NXUEM           SIMK1157
C IDCM(J+8)=NUCM $ IDCM(J+9)=NUCM $ IDCM(J+10)=NUCM              SIMK1158
C IDCM(J+11)=NUGM $ IDCM(J+12)=NUGM $ IDCM(J+13)=NUGM           SIMK1159
C IDCM(J+14)=NUGM $ IDCM(J+15)=NUGM $ IDCM(J+16)=NUGM           SIMK1160
6 CONTINUE                                                         SIMK1161
C IDRM(81)=NRM $ IDCM(81)=NLM                                       SIMK1162
C IDRM(82)=30 $ IDCM(82)=30                                         SIMK1163
C                                                                    SIMK1164
C CHECK IF SCRATCH ARRAY SIZE IS SUFFICIENT                         SIMK1165
C                                                                    SIMK1166
C N=0                                                                SIMK1167
C DO 8 I=1,20                                                       SIMK1168
8 N=N+IDRM(I)*IDCM(I)                                             SIMK1169
C DO 9 I=33,82                                                      SIMK1170
9 N=N+IDRM(I)*IDCM(I)                                             SIMK1171
C IF(N.GT.MS2)                                                       SIMK1172
C ICALL DERRM(MS1,N,MS3,MS4,MS1,MS2,MS3,MS4,1,0,4HSIMK,4H1 ,IW)  SIMK1173
C IF(IPRINT.EQ.6)CALL DFRUG(2,4HSIMK,4H1 ,1,0,IW)                 SIMK1174
C                                                                    SIMK1175
C INITIALIZE THE MEMORY WHERE SIMULATOR MATRIX DATA IS STORED   SIMK1176
C                                                                    SIMK1177
C 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
C READ(IR,16)ICARD                                                  SIMK1184
16 FORMAT(8A10)                                                    SIMK1185
C IF(ICARD(1).EQ.IEND)RETURN                                        SIMK1186
C IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))WRITE(IW,22)                  SIMK1187
C IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))WRITE(IW,24)ICARD            SIMK1188
22 FORMAT(//,20X,27H*** LSA - FLEXSTAR DATA ***,//)             SIMK1189
24 FORMAT(//,1X,8A10,//)                                           SIMK1190
C DO 28 I=1,8                                                       SIMK1191
28 IHEAD(I)=ICARD(I)                                              SIMK1192
C READ(IR,16)ICARD                                                  SIMK1193
C MHEAD=ICARD(1)                                                  SIMK1194
C DECODE(10,30,ICARD(2))NROW,NCOL                                  SIMK1195
30 FORMAT(2I5)                                                     SIMK1196
C IF(ICARD(1).NE.IEND)GO TO 52                                       SIMK1197
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

# Contrails

```

      JR=IR
      IR=4
C
C   LOCATE SIMULATOR DECK DATA
C
32  CONTINUE
    READ(IR,16)ICARD
    IF(EOF(IR))36,44
36  CONTINUE
C
C   PRINT ERROR MESSAGE
C
    WRITE(IW,40)IR
40  FORMAT(1H1,/,/,1X,3RHVEHICLE DATA CANNOT BE FOUND ON FILE= ,I2)
    STOP 111
C
C   READ MATRIX NAME CARD
C
44  CONTINUE
    IF(IHEAD(1).NE.ICARD(1))GO TO 32
    READ(IR,16)ICARD
    MHEAD=ICARD(1)
    DECODE(10,30,(ICARD(2))NROW,NCOL
    GO TO 52
48  CONTINUE
    READ(IR,16)ICARD
    MHEAD=ICARD(1)
    DECODE(10,30,ICARD(2))NROW,NCOL
52  CONTINUE
    IF(IPRINT.EQ.6)CALL DEBUG(3,4HSIMK,4H1 ,1,0,IW)
    DO 54 I=1,112
    IF(MHEAD.EQ.JHFAD(I))GO TO 58
54  CONTINUE
C
C   PRINT ERROR MESSAGE
C
55  CONTINUE
    WRITE(IW,56)
56  FORMAT(1H1,/,/,1X,1QHERROR IN INPUT DATA)
    STOP 111
C
C   READ AND PRINT LSA SIMULATOR DECK DATA
C
58  CONTINUE
    IF(I.GE.112)GO TO 98
C
C   COMPUTE II FROM I SO THAT STEADY GUST COEFF MATRICES ARE
C   STORED IN THE SAME LOCATIONS AS THE GUST COEFF MATRICES
C
    II=I
    IF((I.GT.16).AND.(I.LE.26))II=I-6
    IF((I.GT.26).AND.(I.LE.44))GO TO 55
    IF((I.GT.44).AND.(I.LE.60))II=I-12
    IF((I.GT.60).AND.(I.LE.82))II=I-18
    IF((I.GT.82).AND.(I.LE.104))II=I-24
    IF((I.GT.104).AND.(I.LE.112))II=I-30
C
C   COMPUTE ARRAY START INDEX FOR SIMULATOR MATRIX DATA
C
    N=1
    IM=II-1
    IF(II.FQ.33)IM=20
    IF(IM.EQ.0)GO TO 70
    JMI=IM
    IF(IM.GT.20)JMI=20
    DO 60 J=1,JMI
SIMK1198
SIMK1199
SIMK1200
SIMK1201
SIMK1202
SIMK1203
SIMK1204
SIMK1205
SIMK1206
SIMK1207
SIMK1208
SIMK1209
SIMK1210
SIMK1211
SIMK1212
SIMK1213
SIMK1214
SIMK1215
SIMK1216
SIMK1217
SIMK1218
SIMK1219
SIMK1220
SIMK1221
SIMK1222
SIMK1223
SIMK1224
SIMK1225
SIMK1226
SIMK1227
SIMK1228
SIMK1229
SIMK1230
SIMK1231
SIMK1232
SIMK1233
SIMK1234
SIMK1235
SIMK1236
SIMK1237
SIMK1238
SIMK1239
SIMK1240
SIMK1241
SIMK1242
SIMK1243
SIMK1244
SIMK1245
SIMK1246
SIMK1247
SIMK1248
SIMK1249
SIMK1250
SIMK1251
SIMK1252
SIMK1253
SIMK1254
SIMK1255
SIMK1256
SIMK1257
SIMK1258
SIMK1259
SIMK1260
SIMK1261
SIMK1262
SIMK1263

```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

# Contrails

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	READ AND PRINT SIMULATOR MATRIX DATA	SIMK1269
C	NROWM=IDRM(II) \$ NCOLM=IDCM(II)	SIMK1270
	CALL INPT1(SC(N),NROWM,NCOLM,NROW,NCOL,IR)	SIMK1271
	CALL MPRS1(SC(N),NROWM,NCOLM,NROW,NCOL,MHEAD)	SIMK1272
C	COMPUTE STATE DIMENSION OF SIMULATOR MATRIX DATA READ	SIMK1273
C	IF((II.GT.00).AND.(II.LE.16))NXVP=NROW	SIMK1274
	IF((II.GT.16).AND.(II.LE.32))NXR=NROW	SIMK1275
	IF((II.GT.32).AND.(II.LE.48))NXUE=NROW	SIMK1276
	IF((II.GT.48).AND.(II.LE.64))NRT=NROW	SIMK1277
	IF((II.GT.64).AND.(II.LE.80))NRL=NROW	SIMK1278
C	COMPUTE INPUT DIMENSION OF SIMULATOR MATRIX DATA READ	SIMK1279
C	DO 80 J=1,5	SIMK1280
	JJ=(J-1)*22+I	SIMK1281
	IF(JJ.LT.96)GO TO 80	SIMK1282
	IF(JJ.GT.110)GO TO 80	SIMK1283
	IF(JJ.EQ.96)NUC1=NCOL	SIMK1284
	IF(JJ.EQ.97)NUC2=NCOL	SIMK1285
	IF(JJ.EQ.98)NUC3=NCOL	SIMK1286
	IF(JJ.EQ.99)NUG0=NCOL	SIMK1287
	IF(JJ.EQ.100)NUG1=NCOL	SIMK1288
	IF(JJ.EQ.101)NVG0=NCOL	SIMK1289
	IF(JJ.EQ.102)NVG1=NCOL	SIMK1290
	IF(JJ.EQ.103)NWG0=NCOL	SIMK1291
	IF(JJ.EQ.104)NWG1=NCOL	SIMK1292
	IF(JJ.EQ.105)NUGS0=NCOL	SIMK1293
	IF(JJ.EQ.106)NUGS1=NCOL	SIMK1294
	IF(JJ.EQ.107)NVGS0=NCOL	SIMK1295
	IF(JJ.EQ.108)NVGS1=NCOL	SIMK1296
	IF(JJ.EQ.109)NWGS0=NCOL	SIMK1297
	IF(JJ.EQ.110)NWGS1=NCOL	SIMK1298
80	CONTINUE	SIMK1299
	GO TO 48	SIMK1300
C	PRINT THE LAST CARD READ FROM SIMULATOR MATRIX DATA	SIMK1301
C	98 CONTINUE	SIMK1302
	IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))WRITE(IW,100)MHEAD	SIMK1303
100	FORMAT(//,10X,A10,//)	SIMK1304
	IF(IPRINT.EQ.6)CALL DFBUG(4,4HSIMK,4H1 .1.0,IW)	SIMK1305
C	FORM THE UNITY MATRIX	SIMK1306
C	IF(NXUE.EQ.0)GO TO 134	SIMK1307
	DO 130 I=1,NXUE	SIMK1308
	DO 130 J=1,NXUE	SIMK1309
	UNITY(I,J)=0.0	SIMK1310
130	UNITY(I,I)=1.0	SIMK1311
134	CONTINUE	SIMK1312
C	CHECK FOR DIMENSION ERROR	SIMK1313
C	IF((NXVP.LE.NXVPM).AND.(NXR.LE.NXPM).AND.(NXUE.LE.NXUEM)	SIMK1314
	1.AND.(NUC1.LE.NUCM).AND.(NUC2.LE.NUCM).AND.(NUC3.LE.NUCM)	SIMK1315
	2.AND.(NUG0.LE.NUGM).AND.(NUG1.LE.NUGM)	SIMK1316
	3.AND.(NVG0.LE.NVG1).AND.(NVG1.LE.NVG1)	SIMK1317
		SIMK1318
		SIMK1319
		SIMK1320
		SIMK1321
		SIMK1322
		SIMK1323
		SIMK1324
		SIMK1325
		SIMK1326
		SIMK1327
		SIMK1328
		SIMK1329

Figure 18. Subroutine SIMK1 Program Listing (Continued)

# Contrails

```
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 138 SIMK1335
WRITE(IW,136) SIMK1336
136 FORMAT(1H1,/,1X,43HDIMENSION 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
NR=NRT+NRL SIMK1347
IF(IR.NE.4)GO TO 17 SIMK1348
IR=JR SIMK1349
IF(IPRINT.EQ.6)CALL DFBUG(5,4HSIMK,4H1 .1,0,IW) 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 168 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)

# Contrails

```
      DO 176 K=1,NUC2
      KK=NUC1+K
176  XDOTL(I)=XDOTL(I)+VPDELS1(I,K)*U(KK)
      IF (NUC3.LE.0)GO TO 184
      DO 180 K=1,NUC3
      KK=NUC1+NUC2+K
180  XDOTL(J)=XDOTL(J)+VPDELS2(I,K)*U(KK)
184  CONTINUE
      MU=NUC1+NUC2+NUC3
      II=I
C
C   FROM U-GUST INPUTS
C
      IF (NUG0.LE.0)GO TO 196
      DO 188 K=1,NUG0
      KK=MU+K
188  XDOTL(II)=XDOTL(II)+VPUG0(I,K)*U(KK)
      IF (NUG1.LE.0)GO TO 196
      DO 192 K=1,NUG1
      KK=MU+NUG0+K
192  XDOTL(II)=XDOTL(II)+VPUG1(I,K)*U(KK)
196  CONTINUE
      MU=MU+NUG0+NUG1
C
C   FROM V-GUST INPUTS
C
      IF (NVG0.LE.0)GO TO 208
      DO 200 K=1,NVG0
      KK=MU+K
200  XDOTL(II)=XDOTL(II)+VPVG0(I,K)*U(KK)
      IF (NVG1.LE.0)GO TO 208
      DO 204 K=1,NVG1
      KK=MU+NVG0+K
204  XDOTL(II)=XDOTL(II)+VPVG1(I,K)*U(KK)
208  CONTINUE
      MU=MU+NVG0+NVG1
C
C   FROM W-GUST INPUTS
C
      IF (NWG0.LE.0)GO TO 220
      DO 212 K=1,NWG0
      KK=MU+K
212  XDOTL(II)=XDOTL(II)+VPWG0(I,K)*U(KK)
      IF (NWG1.LE.0)GO TO 220
      DO 216 K=1,NWG1
      KK=MU+NWG0+K
216  XDOTL(II)=XDOTL(II)+VPWG1(I,K)*U(KK)
220  CONTINUE
      MU=MU+NWG0+NWG1
C
C   FROM STEADY U-GUST INPUTS
C
      IF (NUGS0.LE.0)GO TO 232
      DO 224 K=1,NUGS0
      KK=MU+K
224  XDOTL(II)=XDOTL(II)+VPUG0(I,K)*U(KK)
      IF (NUGS1.LE.0)GO TO 232
      DO 228 K=1,NUGS1
      KK=MU+NUGS0+K
228  XDOTL(II)=XDOTL(II)+VPUG1(I,K)*U(KK)
232  CONTINUE
      MU=MU+NUGS0+NUGS1
C
C   FROM STEADY V-GUST INPUTS
C
      IF (NVGS0.LE.0)GO TO 244
```

```
SIMK1396
SIMK1397
SIMK1398
SIMK1399
SIMK1400
SIMK1401
SIMK1402
SIMK1403
SIMK1404
SIMK1405
SIMK1406
SIMK1407
SIMK1408
SIMK1409
SIMK1410
SIMK1411
SIMK1412
SIMK1413
SIMK1414
SIMK1415
SIMK1416
SIMK1417
SIMK1418
SIMK1419
SIMK1420
SIMK1421
SIMK1422
SIMK1423
SIMK1424
SIMK1425
SIMK1426
SIMK1427
SIMK1428
SIMK1429
SIMK1430
SIMK1431
SIMK1432
SIMK1433
SIMK1434
SIMK1435
SIMK1436
SIMK1437
SIMK1438
SIMK1439
SIMK1440
SIMK1441
SIMK1442
SIMK1443
SIMK1444
SIMK1445
SIMK1446
SIMK1447
SIMK1448
SIMK1449
SIMK1450
SIMK1451
SIMK1452
SIMK1453
SIMK1454
SIMK1455
SIMK1456
SIMK1457
SIMK1458
SIMK1459
SIMK1460
SIMK1461
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)



# Contrails

```

DO 236 K=1,NVGS0
KK=MU+K
236 XDOTL (II)=XDOTL (II)+VPVG0 (I,K)*U (KK)
IF (NVGS1.LE.0)GO TO 244
DO 240 K=1,NVGS1
KK=MU+NVGS0+K
240 XDOTL (II)=XDOTL (II)+VPVG1 (I,K)*U (KK)
244 CONTINUE
MU=MU+NVGS0+NVGS1
C
C FROM STEADY W-GUST INPUTS
C
IF (NWGS0.LE.0)GO TO 256
DO 248 K=1,NWGS0
KK=MU+K
248 XDOTL (II)=XDOTL (II)+VPWG0 (I,K)*U (KK)
IF (NWGS1.LE.0)GO TO 256
DO 252 K=1,NWGS1
KK=MU+NWGS0+K
252 XDOTL (II)=XDOTL (II)+VPWG1 (I,K)*U (KK)
256 CONTINUE
MU=MU+NWGS0+NWGS1
260 CONTINUE
264 CONTINUE
C
C DIFFERENTIAL EQUATIONS FOR RIGID BODY ATTITUDES
C
IF (NXP.LE.0)GO TO 280
DO 272 I=1,NXP
II=NXP+I
XDOTL (II)=0.0
IF (NXVP.LE.0)GO TO 272
C
C FROM RIGID BODY VELOCITIES
C
DO 268 K=1,NXVP
268 XDOTL (II)=XDOTL (II)+RVP0 (I,K)*X (K)+RVP1 (I,K)*XDOT (K)
272 CONTINUE
C
C FROM RIGID BODY ATTITUDES
C
DO 276 K=1,NXR
KK=NXP+K
276 XDOTL (II)=XDOTL (II)+RR0 (I,K)*X (KK)+RR1 (I,K)*XDOT (KK)
280 CONTINUE
C
C DIFFERENTIAL EQUATIONS FOR BENDING MODE DISPLACEMENTS AND RATES
C
IF (NXUF.LE.0)GO TO 396
DO 284 I=1,NXUF
II=NXP+NXR+I
XDOTL (II)=0.0
DO 284 K=1,NXUF
KK=NXP+NXR+NXUF+K
284 XDOTL (II)=XDOTL (II)+UNITY (I,K)*X (KK)
DO 392 I=1,NXUF
II=NXP+NXR+NXUF+I
XDOTL (II)=0.0
IF (NXVP.LE.0)GO TO 292
C
C FROM RIGID BODY VELOCITIES
C
DO 288 K=1,NXVP
288 XDOTL (II)=XDOTL (II)+UEVP0 (I,K)*X (K)+UEVP1 (I,K)*XDOT (K)
292 CONTINUE
IF (NXR.LE.0)GO TO 298

```

```

SIMK1462
SIMK1463
SIMK1464
SIMK1465
SIMK1466
SIMK1467
SIMK1468
SIMK1469
SIMK1470
SIMK1471
SIMK1472
SIMK1473
SIMK1474
SIMK1475
SIMK1476
SIMK1477
SIMK1478
SIMK1479
SIMK1480
SIMK1481
SIMK1482
SIMK1483
SIMK1484
SIMK1485
SIMK1486
SIMK1487
SIMK1488
SIMK1489
SIMK1490
SIMK1491
SIMK1492
SIMK1493
SIMK1494
SIMK1495
SIMK1496
SIMK1497
SIMK1498
SIMK1499
SIMK1500
SIMK1501
SIMK1502
SIMK1503
SIMK1504
SIMK1505
SIMK1506
SIMK1507
SIMK1508
SIMK1509
SIMK1510
SIMK1511
SIMK1512
SIMK1513
SIMK1514
SIMK1515
SIMK1516
SIMK1517
SIMK1518
SIMK1519
SIMK1520
SIMK1521
SIMK1522
SIMK1523
SIMK1524
SIMK1525
SIMK1526
SIMK1527

```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

# Contrails

```
C
C FROM RIGID BODY ATTITUDES
C
C DO 296 K=1,NXR
C KK=NXVP+K
296 XDOTL (II)=XDOTL (II)+UFRO (I,K)*X(KK)+UER1 (I,K)*XDOT (KK)
298 CONTINUE
C
C FROM HENDING MODES
C
C DO 300 K=1,NXUE
C KK=NXVP+NXR+K
C KKK=NXVP+NXR+NXUE+K
300 XDOTL (II)=XDOTL (II)+UFUE0 (I,K)*X(KK)+UFUE1 (I,K)*X(KKK)
C 1+UEUE2 (I,K)*XDOT (KKK)
C
C FROM CONTROL SURFACE INPUTS
C
C IF (NUC1.LE.0)GO TO 316
C DO 304 K=1,NUC1
304 XDOTL (II)=XDOTL (II)+UEDELS0 (I,K)*U(K)
C IF (NUC2.LE.0)GO TO 316
C DO 308 K=1,NUC2
C KK=NUC1+K
308 XDOTL (II)=XDOTL (II)+UFDELS1 (I,K)*U(KK)
C IF (NUC3.LE.0)GO TO 316
C DO 312 K=1,NUC3
C KK=NUC1+NUC2+K
312 XDOTL (II)=XDOTL (II)+UEDELS2 (I,K)*U(KK)
316 CONTINUE
C MU=NUC1+NUC2+NUC3
C
C FROM U-GUST INPUTS
C
C IF (NUG0.LE.0)GO TO 328
C DO 320 K=1,NUG0
C KK=MU+K
320 XDOTL (II)=XDOTL (II)+UFUG0 (I,K)*U(KK)
C IF (NUG1.LE.0)GO TO 328
C DO 324 K=1,NUG1
C KK=MU+NUG0+K
324 XDOTL (II)=XDOTL (II)+UFUG1 (I,K)*U(KK)
328 CONTINUE
C MU=MU+NUG0+NUG1
C
C FROM V-GUST INPUTS
C
C IF (NVG0.LE.0)GO TO 340
C DO 332 K=1,NVG0
C KK=MU+K
332 XDOTL (II)=XDOTL (II)+UFVG0 (I,K)*U(KK)
C IF (NVG1.LE.0)GO TO 340
C DO 336 K=1,NVG1
C KK=MU+NVG0+K
336 XDOTL (II)=XDOTL (II)+UFVG1 (I,K)*U(KK)
340 CONTINUE
C MU=MU+NVG0+NVG1
C
C FROM W-GUST INPUTS
C
C IF (NWG0.LE.0)GO TO 352
C DO 344 K=1,NWG0
C KK=MU+K
344 XDOTL (II)=XDOTL (II)+UEWGO (I,K)*U(KK)
C IF (NWG1.LE.0)GO TO 352
C DO 348 K=1,NWG1
```

```
SIMK1528
SIMK1529
SIMK1530
SIMK1531
SIMK1532
SIMK1533
SIMK1534
SIMK1535
SIMK1536
SIMK1537
SIMK1538
SIMK1539
SIMK1540
SIMK1541
SIMK1542
SIMK1543
SIMK1544
SIMK1545
SIMK1546
SIMK1547
SIMK1548
SIMK1549
SIMK1550
SIMK1551
SIMK1552
SIMK1553
SIMK1554
SIMK1555
SIMK1556
SIMK1557
SIMK1558
SIMK1559
SIMK1560
SIMK1561
SIMK1562
SIMK1563
SIMK1564
SIMK1565
SIMK1566
SIMK1567
SIMK1568
SIMK1569
SIMK1570
SIMK1571
SIMK1572
SIMK1573
SIMK1574
SIMK1575
SIMK1576
SIMK1577
SIMK1578
SIMK1579
SIMK1580
SIMK1581
SIMK1582
SIMK1583
SIMK1584
SIMK1585
SIMK1586
SIMK1587
SIMK1588
SIMK1589
SIMK1590
SIMK1591
SIMK1592
SIMK1593
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

# Contrails

	KK=MU+NWG0+K	SIMK1594
	348 XDOTL (II)=XDOTL (II)+UEWGI (I,K)*U(KK)	SIMK1595
	352 CONTINUE	SIMK1596
	MU=MU+NWG0+NWGI	SIMK1597
C		SIMK1598
C	FROM STEADY U-GUST INPUTS	SIMK1599
C		SIMK1600
	IF (NUGS0.LE.0)GO TO 364	SIMK1601
	DO 356 K=1,NUGS0	SIMK1602
	KK=MU+K	SIMK1603
	356 XDOTL (II)=XDOTL (II)+UEUG0 (I,K)*U(KK)	SIMK1604
	IF (NUGS1.LE.0)GO TO 364	SIMK1605
	DO 360 K=1,NUGS1	SIMK1606
	KK=MU+NUGS0+K	SIMK1607
	360 XDOTL (II)=XDOTL (II)+UEUG1 (I,K)*U(KK)	SIMK1608
	364 CONTINUE	SIMK1609
	MU=MU+NUGS0+NUGS1	SIMK1610
C		SIMK1611
C	FROM STEADY V-GUST INPUTS	SIMK1612
C		SIMK1613
	IF (NVGS0.LE.0)GO TO 376	SIMK1614
	DO 368 K=1,NVGS0	SIMK1615
	KK=MU+K	SIMK1616
	368 XDOTL (II)=XDOTL (II)+UEVG0 (I,K)*U(KK)	SIMK1617
	IF (NVGS1.LE.0)GO TO 376	SIMK1618
	DO 372 K=1,NVGS1	SIMK1619
	KK=MU+NVGS0+K	SIMK1620
	372 XDOTL (II)=XDOTL (II)+UEVG1 (I,K)*U(KK)	SIMK1621
	376 CONTINUE	SIMK1622
	MU=MU+NVGS0+NVGS1	SIMK1623
C		SIMK1624
C	FROM STEADY W-GUST INPUTS	SIMK1625
C		SIMK1626
	IF (NWGS0.LE.0)GO TO 388	SIMK1627
	DO 380 K=1,NWGS0	SIMK1628
	KK=MU+K	SIMK1629
	380 XDOTL (II)=XDOTL (II)+UEWGI (I,K)*U(KK)	SIMK1630
	IF (NWGS1.LE.0)GO TO 388	SIMK1631
	DO 384 K=1,NWGS1	SIMK1632
	KK=MU+NWGS0+K	SIMK1633
	384 XDOTL (II)=XDOTL (II)+UEWGI (I,K)*U(KK)	SIMK1634
	388 CONTINUE	SIMK1635
	MU=MU+NWGS0+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)+TVP0 (I,K)*X (K)+TVP1 (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)

# Contrails

```
C      IF(NXUF,LE.0)GO TO 420
C      FROM HFNDING MODES
C      DO 416 K=1,NXUF
      KK=NXVP+NXR*K
      KKK=NXVP+NXR+NXUF*K
416  RL(I)=RL(I)+TUF0(I,K)*X(KK)+TUF1(I,K)*X(KKK)+TUF2(I,K)*XDOT(KKK)
420  CONTINUE
C      FROM CONTROL SURFACE INPUTS
C      IF(NUC1,LE.0)GO TO 436
      DO 424 K=1,NUC1
424  RL(I)=RL(I)+TDFLS0(I,K)*U(K)
      IF(NUC2,LE.0)GO TO 436
      DO 428 K=1,NUC2
      KK=NUC1+K
428  RL(I)=RL(I)+TDFLS1(I,K)*U(KK)
      IF(NUC3,LE.0)GO TO 436
      DO 432 K=1,NUC3
      KK=NUC1+NUC2+K
432  RL(I)=RL(I)+TDFLS2(I,K)*U(KK)
436  CONTINUE
      MU=NUC1+NUC2+NUC3
C      FROM U-GUST INPUTS
C      IF(NUG0,LE.0)GO TO 448
      DO 440 K=1,NUG0
      KK=MU+K
440  RL(I)=RL(I)+TUG0(I,K)*U(KK)
      IF(NUG1,LE.0)GO TO 448
      DO 444 K=1,NUG1
      KK=MU+NUG0+K
444  RL(I)=RL(I)+TUG1(I,K)*U(KK)
448  CONTINUE
      MU=MU+NUG0+NUG1
C      FROM V-GUST INPUTS
C      IF(NVG0,LE.0)GO TO 460
      DO 452 K=1,NVG0
      KK=MU+K
452  RL(I)=RL(I)+TVG0(I,K)*U(KK)
      IF(NVG1,LE.0)GO TO 460
      DO 456 K=1,NVG1
      KK=MU+NVG0+K
456  RL(I)=RL(I)+TVG1(I,K)*U(KK)
460  CONTINUE
      MU=MU+NVG0+NVG1
C      FROM W-GUST INPUTS
C      IF(NWG0,LE.0)GO TO 472
      DO 464 K=1,NWG0
      KK=MU+K
464  RL(I)=RL(I)+TWG0(I,K)*U(KK)
      IF(NWG1,LE.0)GO TO 472
      DO 468 K=1,NWG1
      KK=MU+NWG0+K
468  RL(I)=RL(I)+TWG1(I,K)*U(KK)
472  CONTINUE
      MU=MU+NWG0+NWG1
C      FROM STEADY U-GUST INPUTS
```

```
SIMK1660
SIMK1661
SIMK1662
SIMK1663
SIMK1664
SIMK1665
SIMK1666
SIMK1667
SIMK1668
SIMK1669
SIMK1670
SIMK1671
SIMK1672
SIMK1673
SIMK1674
SIMK1675
SIMK1676
SIMK1677
SIMK1678
SIMK1679
SIMK1680
SIMK1681
SIMK1682
SIMK1683
SIMK1684
SIMK1685
SIMK1686
SIMK1687
SIMK1688
SIMK1689
SIMK1690
SIMK1691
SIMK1692
SIMK1693
SIMK1694
SIMK1695
SIMK1696
SIMK1697
SIMK1698
SIMK1699
SIMK1700
SIMK1701
SIMK1702
SIMK1703
SIMK1704
SIMK1705
SIMK1706
SIMK1707
SIMK1708
SIMK1709
SIMK1710
SIMK1711
SIMK1712
SIMK1713
SIMK1714
SIMK1715
SIMK1716
SIMK1717
SIMK1718
SIMK1719
SIMK1720
SIMK1721
SIMK1722
SIMK1723
SIMK1724
SIMK1725
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

# Contrails

```
IF(NUGS0.LE.0)GO TO 484
DO 476 K=1,NUGS0
KK=MU+K
476 RL(I)=RL(I)+TUG0(I,K)*U(KK)
IF(NUGS1.LE.0)GO TO 484
DO 480 K=1,NUGS1
KK=MU+NUGS0+K
480 RL(I)=RL(I)+TUG1(I,K)*U(KK)
484 CONTINUE
MU=MU+NUGS0+NUGS1
C
C FROM STEADY V-GUST INPUTS
C
IF(NVGS0.LE.0)GO TO 496
DO 488 K=1,NVGS0
KK=MU+K
488 RL(I)=RL(I)+TVG0(I,K)*U(KK)
IF(NVGS1.LE.0)GO TO 496
DO 492 K=1,NVGS1
KK=MU+NVGS0+K
492 RL(I)=RL(I)+TVG1(I,K)*U(KK)
496 CONTINUE
MU=MU+NVGS0+NVGS1
C
C FROM STEADY W-GUST INPUTS
C
IF(NWGS0.LE.0)GO TO 508
DO 500 K=1,NWGS0
KK=MU+K
500 RL(I)=RL(I)+TWG0(I,K)*U(KK)
IF(NWGS1.LE.0)GO TO 508
DO 504 K=1,NWGS1
KK=MU+NWGS0+K
504 RL(I)=RL(I)+TWG1(I,K)*U(KK)
508 CONTINUE
MU=MU+NWGS0+NWGS1
512 CONTINUE
516 CONTINUE
C
C LOAD EQUATIONS
C
IF(NRL.LE.0)GO TO 716
DO 712 I=1,NRL
J=NRT+I
RL(J)=0.0
IF(NXVP.LE.0)GO TO 604
C
C FROM RIGID BODY VELOCITIES
C
DO 600 K=1,NXVP
600 RL(J)=RL(J)+LVP0(I,K)*X(K)+LVP1(I,K)*XDOT(K)
604 CONTINUE
IF(NXR.LE.0)GO TO 612
C
C FROM RIGID BODY ATTITUDES
C
DO 608 K=1,NXR
KK=NXVP+K
608 RL(J)=RL(J)+LR0(I,K)*X(KK)+LR1(I,K)*XDOT(KK)
612 CONTINUE
IF(NXUE.LE.0)GO TO 620
C
C FROM BENDING MODES
C
DO 616 K=1,NXUF
SIMK1726
SIMK1727
SIMK1728
SIMK1729
SIMK1730
SIMK1731
SIMK1732
SIMK1733
SIMK1734
SIMK1735
SIMK1736
SIMK1737
SIMK1738
SIMK1739
SIMK1740
SIMK1741
SIMK1742
SIMK1743
SIMK1744
SIMK1745
SIMK1746
SIMK1747
SIMK1748
SIMK1749
SIMK1749
SIMK1750
SIMK1751
SIMK1752
SIMK1753
SIMK1754
SIMK1755
SIMK1756
SIMK1757
SIMK1758
SIMK1759
SIMK1760
SIMK1761
SIMK1762
SIMK1763
SIMK1764
SIMK1765
SIMK1766
SIMK1767
SIMK1768
SIMK1769
SIMK1770
SIMK1771
SIMK1772
SIMK1773
SIMK1774
SIMK1775
SIMK1776
SIMK1777
SIMK1778
SIMK1779
SIMK1780
SIMK1781
SIMK1782
SIMK1783
SIMK1784
SIMK1785
SIMK1786
SIMK1787
SIMK1788
SIMK1789
SIMK1790
SIMK1791
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

# Contrails

```
      KK=NXVP+NXR*K
      KKK=NXVP+NXR*NXUF+K
616  RL(J)=RL(J)+LDF0(I,K)*X(KK)+LDF1(I,K)*X(KKK)+LDF2(I,K)*X00T(KKK)
620  CONTINUE
C
C      FROM CONTROL SURFACE INPUTS
C
      IF(NUC1.LE.0)GO TO 636
      DO 624 K=1,NUC1
624  RL(J)=RL(J)+LDFLS0(I,K)*U(K)
      IF(NUC2.LE.0)GO TO 636
      DO 628 K=1,NUC2
      KK=NUC1+K
628  RL(J)=RL(J)+LDFLS1(I,K)*U(KK)
      IF(NUC3.LE.0)GO TO 636
      DO 632 K=1,NUC3
      KK=NUC1+NUC2+K
632  RL(J)=RL(J)+LDFLS2(I,K)*U(KK)
636  CONTINUE
      MU=NUC1+NUC2+NUC3
C
C      FROM U-GUST INPUTS
C
      IF(NUG0.LE.0)GO TO 648
      DO 640 K=1,NUG0
      KK=MU+K
640  RL(J)=RL(J)+LUG0(I,K)*U(KK)
      IF(NUG1.LE.0)GO TO 648
      DO 644 K=1,NUG1
      KK=MU+NUG0+K
644  RL(J)=RL(J)+LUG1(I,K)*U(KK)
648  CONTINUE
      MU=MU+NUG0+NUG1
C
C      FROM V-GUST INPUTS
C
      IF(NVG0.LE.0)GO TO 660
      DO 652 K=1,NVG0
      KK=MU+K
652  RL(J)=RL(J)+LVG0(I,K)*U(KK)
      IF(NVG1.LE.0)GO TO 660
      DO 656 K=1,NVG1
      KK=MU+NVG0+K
656  RL(J)=RL(J)+LVG1(I,K)*U(KK)
660  CONTINUE
      MU=MU+NVG0+NVG1
C
C      FROM W-GUST INPUTS
C
      IF(NWG0.LE.0)GO TO 672
      DO 664 K=1,NWG0
      KK=MU+K
664  RL(J)=RL(J)+LWG0(I,K)*U(KK)
      IF(NWG1.LE.0)GO TO 672
      DO 668 K=1,NWG1
      KK=MU+NWG0+K
668  RL(J)=RL(J)+LWG1(I,K)*U(KK)
672  CONTINUE
      MU=MU+NWG0+NWG1
C
C      FROM STEADY U-GUST INPUTS
C
      IF(NUGS0.LE.0)GO TO 684
      DO 676 K=1,NUGS0
      KK=MU+K
676  RL(J)=RL(J)+LUG0(I,K)*U(KK)
```

```
SIMK1792
SIMK1793
SIMK1794
SIMK1795
SIMK1796
SIMK1797
SIMK1798
SIMK1799
SIMK1800
SIMK1801
SIMK1802
SIMK1803
SIMK1804
SIMK1805
SIMK1806
SIMK1807
SIMK1808
SIMK1809
SIMK1810
SIMK1811
SIMK1812
SIMK1813
SIMK1814
SIMK1815
SIMK1816
SIMK1817
SIMK1818
SIMK1819
SIMK1820
SIMK1821
SIMK1822
SIMK1823
SIMK1824
SIMK1825
SIMK1826
SIMK1827
SIMK1828
SIMK1829
SIMK1830
SIMK1831
SIMK1832
SIMK1833
SIMK1834
SIMK1835
SIMK1836
SIMK1837
SIMK1838
SIMK1839
SIMK1840
SIMK1841
SIMK1842
SIMK1843
SIMK1844
SIMK1845
SIMK1846
SIMK1847
SIMK1848
SIMK1849
SIMK1850
SIMK1851
SIMK1852
SIMK1853
SIMK1854
SIMK1855
SIMK1856
SIMK1857
```

Figure 18. Subroutine SIMK1 Program Listing (Continued)

# Contracts

```

      IF (NUGS1.LE.0) GO TO 684
      DO 680 K=1,NUGS1
      KK=MU+NUGS0+K
680  RL(J)=RL(J)+LUG1(I,K)*U(KK)
684  CONTINUE
      MU=MU+NUGS0+NUGS1
C
C   FROM STEADY V-GUST INPUTS
C
      IF (NVGS0.LE.0) GO TO 696
      DO 688 K=1,NVGS0
      KK=MU+K
688  RL(J)=RL(J)+LVG0(I,K)*U(KK)
      IF (NVGS1.LE.0) GO TO 696
      DO 692 K=1,NVGS1
      KK=MU+NVGS0+K
692  RL(J)=RL(J)+LVG1(I,K)*U(KK)
696  CONTINUE
      MU=MU+NVGS0+NVGS1
C
C   FROM STEADY W-GUST INPUTS
C
      IF (NWGS0.LE.0) GO TO 708
      DO 700 K=1,NWGS0
      KK=MU+K
700  RL(J)=RL(J)+LWG0(I,K)*U(KK)
      IF (NWGS1.LE.0) GO TO 708
      DO 704 K=1,NWGS1
      KK=MU+NWGS0+K
704  RL(J)=RL(J)+LWG1(I,K)*U(KK)
708  CONTINUE
      MU=MU+NWGS0+NWGS1
712  CONTINUE
716  CONTINUE
      RETURN
      END
SIMK1858
SIMK1859
SIMK1860
SIMK1861
SIMK1862
SIMK1863
SIMK1864
SIMK1865
SIMK1866
SIMK1867
SIMK1868
SIMK1869
SIMK1870
SIMK1871
SIMK1872
SIMK1873
SIMK1874
SIMK1875
SIMK1876
SIMK1877
SIMK1878
SIMK1879
SIMK1880
SIMK1881
SIMK1882
SIMK1883
SIMK1884
SIMK1885
SIMK1886
SIMK1887
SIMK1888
SIMK1889
SIMK1890
SIMK1891
SIMK1892
SIMK1893
```

Figure 18. Subroutine SIMK1 Program Listing (Concluded)

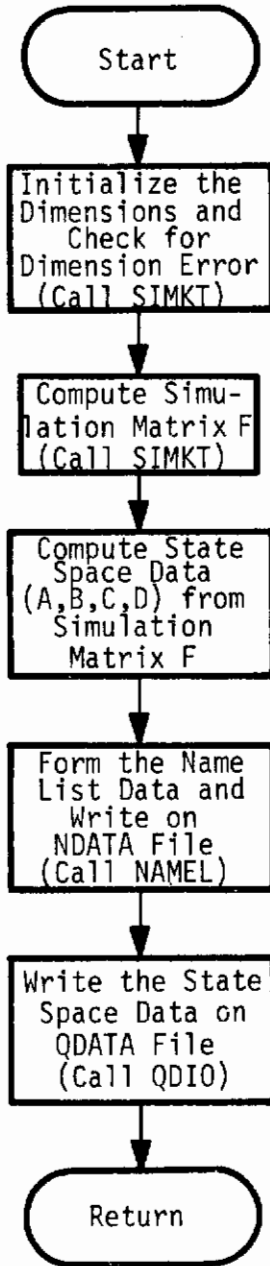


Figure 19. Subroutine STAMK2 Flow Chart



# Contrails

```

SUBROUTINE STAMK2(V,W,F,XDOT,X,RI,UI,J,VNX,NNR,NUU,
1A,B,C,D,AT,BT,CT,DT,P,Q,R,S,PRINT,HS,VNS,VNS,DFSS,UNITS,
2NNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI,
3MAXN,MAXM,NXM,NRM,NUM,NYM,MR,MTFB,MS1,MS2,MS3,MS4,NB)
C
C PURPOSE - TO OBTAIN STATE MODEL FROM TRANSFER FN REPRESENTATION
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C DATE WRITTEN - 1975
C
C SUBPROGRAMS CALLED
C   DE4UG
C   DEPRMS
C   MPRS
C   ODT0
C   TOTNVR
C   DERRM
C   HPP
C   NAMEL
C   SIKKT
C
C ARGUMENTS LIST
C   V          V ARRAY FOR COMPUTING SIMULATION MATRIX
C   W          W ARRAY FOR COMPUTING SIMULATION MATRIX
C   F          SIMULATION MATRIX
C   XDOT      ARRAY FOR STATE DERIVATIVES
C   X          ARRAY FOR STATES
C   RI        ARRAY FOR INTERNAL OUTPUTS
C   UI        ARRAY FOR INTERNAL INPUTS
C   U          ARRAY FOR EXTERNAL INPUTS
C   NNX       ARRAY FOR STORING SYSTEM DIMENSION NX
C   NNR       ARRAY FOR STORING SYSTEM DIMENSION NR
C   NUU       ARRAY FOR STORING SYSTEM DIMENSION NU
C   A          IN/OUT STATE TRANSITION MATRIX
C   B          IN/OUT CONTROL INPUT MATRIX
C   C          IN/OUT STATE OUTPUT MATRIX
C   D          IN/OUT CONTROL OUTPUT MATRIX
C   AT        ARRAY FOR STORING TR FUNCTION QUADRUPLE A
C   BT        ARRAY FOR STORING TR FUNCTION QUADRUPLE B
C   CT        ARRAY FOR STORING TR FUNCTION QUADRUPLE C
C   DT        ARRAY FOR STORING TR FUNCTION QUADRUPLE D
C   P          INTERCONNECTION QUADRUPLE
C   Q          INTERCONNECTION QUADRUPLE
C   R          INTERCONNECTION QUADRUPLE
C   S          INTERCONNECTION QUADRUPLE
C   PRINT     STORES TRANSFER FUNCTION DATA
C   HS        TRANSFER FUNCTION DATA
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
C STAMK2 2
C STAMK2 3
C STAMK2 4
C STAMK2 5
C STAMK2 6
C STAMK2 7
C STAMK2 8
C STAMK2 9
C STAMK210
C STAMK211
C STAMK212
C STAMK213
C STAMK214
C STAMK215
C STAMK216
C STAMK217
C STAMK218
C STAMK219
C STAMK220
C STAMK221
C STAMK222
C STAMK223
C STAMK224
C STAMK225
C STAMK226
C STAMK227
C STAMK228
C STAMK229
C STAMK230
C STAMK231
C STAMK232
C STAMK233
C STAMK234
C STAMK235
C STAMK236
C STAMK237
C STAMK238
C STAMK239
C STAMK240
C STAMK241
C STAMK242
C STAMK243
C STAMK244
C STAMK245
C STAMK246
C STAMK247
C STAMK248
C STAMK249
C STAMK250
C STAMK251
C STAMK252
C STAMK253
C STAMK254
C STAMK255
C STAMK256
C STAMK257
C STAMK258
C STAMK259
C STAMK260
C STAMK261
C STAMK262
C STAMK263
C STAMK264

```

Figure 20. Subroutine STAMK2 Program Listing

# Contrails

```

C      NY      INPUT      MAXIMUM DIMENSION FOR INTERCONN EQUATIONS      STAMK265
C      MB      INPUT      MAXIMUM NO OF SUBSYSTEMS FOR COMBINING      STAMK266
C      MTFB     INPUT      MAX NO OF TRANSFER FN BLOCKS FOR COMBINING      STAMK267
C      MST      INPUT      MAX POWER OF S IN THE TRANSFER FUNCTION      STAMK268
C      MT       INPUT      MAX NO OF TERMS IN THE TRANSFER FUNCTION      STAMK269
C      MS1      INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S1      STAMK270
C      MS2      INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S2      STAMK271
C      MS3      INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S3      STAMK272
C      MS4      INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S4      STAMK273
C      NB       INPUT      MAXIMUM SYSTEM NO - IMPLICIT MODEL      STAMK274
C                                                    STAMK275
COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS      STAMK276
COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)      STAMK277
1,PHEAD(26)      STAMK278
COMMON /SCI/ S1(1)      STAMK279
C      DIMENSION DESS(NX,10,MB),UNITSS(NX,4,MR)      STAMK280
C      DIMENSION DESO(NRM,10,MR),UNITO(NRM,4,MR)      STAMK281
C      DIMENSION DESI(NUM,10,MB),UNITI(NUM,4,MR)      STAMK282
C      DIMENSION NXX(MR),NRR(MR),NIJ(MB)      STAMK283
C      DIMENSION V(MAXN),W(MAXM),F(MAXN,MAXM)      STAMK284
C      DIMENSION XDOT(MST,MTFB),X(MST,MTFB),RI(1,MTFB),UI(1,MTFB)      STAMK285
C      DIMENSION U(NUM),NXX(MTFB),NRR(MTFB),NNU(MTFB)      STAMK286
C      DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM)      STAMK287
C      DIMENSION AT(MST,MST,MTFB),RT(MST,MTFB)      STAMK288
C      DIMENSION CT(1,MST,MTFB),DT(1,1,MTFB)      STAMK289
C      DIMENSION P(MTFB,MTFB),Q(MTFB,NUM),R(NRM,MTFB),S(NRM,NUM)      STAMK290
C      DIMENSION PRINT(2,MT),HS(2,MT,MTFB)      STAMK291
C      DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)      STAMK292
C      DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)      STAMK293
C      DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)      STAMK294
C      IF(IPRINT.EQ.6)CALL DEBUG(1,4HSTAM,4HK2 ,2,0,IW)      STAMK295
C      L1=1 $ L2=L1+NX*MR*10 $ L3=L2+NX*MR*4 $ L4=L3+NRM*MR*10      STAMK296
C      L5=L4+NRM*MR*4 $ L6=L5+NUM*MR*10 $ L7=L6+NUM*MR*4      STAMK297
C      L8=L7+MR $ L9=L8+MR $ L10=L9+MR      STAMK298
C      IF(L1.GT.MS1)      STAMK299
C      1CALL DERRM(L10,MS2,MS3,MS4,MS1,MS2,MS3,MS4,2,0,4HSTAM,4HK2 ,IW)      STAMK100
C      IF(IPRINT.EQ.6)CALL DEBUG(2,4HSTAM,4HK2 ,2,0,IW)      STAMK101
C      NR1=0 $ NR2=0 $ NR3=0 $ NIJ=0 $ NI2=0 $ NU3=0      STAMK102
C      NXA=0 $ NRA=0 $ NUA=0      STAMK103
C      EPSF=.0E-30 $ T=0.0 $ INIT=0 $ NFLAG=0      STAMK104
C      IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))CALL MPR(HEAD,IW)      STAMK105
C                                                    STAMK106
C      INITIALIZING CALL TO SIMKT      STAMK107
C                                                    STAMK108
C      NX=0 $ NY=0 $ NR=0 $ NU=0      STAMK109
C      N1=1 $ N2=N1+NX $ N3=N2+NY      STAMK110
C      CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NXX,NRR,NNU,      STAMK111
C      1AT,RT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MTFB,MST,      STAMK112
C      2MT,NUM,NRM,INIT,T)      STAMK113
C      IF(IPRINT.EQ.6)CALL DEBUG(3,4HSTAM,4HK2 ,2,0,IW)      STAMK114
C                                                    STAMK115
C      CHECK FOR DIMENSION ERROR      STAMK116
C                                                    STAMK117
C      INIT = 1      STAMK118
C      M=2*NX+NY+NIJ      STAMK119
C      N=NX+NY+NR      STAMK120
C      IF((NX.GT.NXM).OR.(NR.GT.NRM).OR.(NU.GT.NUM).OR.(NY.GT.NYM))      STAMK121
C      1CALL DERRMS(NX+NR+NU+NY,NXM,NRM,NUM,NYM,2,0,4HSTAM,4HK2 ,IW)      STAMK122
C      N1=1 $ N2=N1+NX $ N3=N2+NY      STAMK123
C                                                    STAMK124
C      ZERO OUT XDOT,RI,UI,X,U      STAMK125
C                                                    STAMK126
C      DO 10 NN=1,NMAX      STAMK127
C      MX=NN*(NN)      STAMK128
C      DO 10 J=1,MX      STAMK129
C      XDOT(J,NN)=0.0      STAMK130

```

Figure 20. Subroutine STAMK2 Program Listing (Continued)

# Contrails

```

10  X(J,NN)=0.0
    DO 11 NN=1,NMAX
    MX=NN*(NN)
    DO 12 J=1,MX
12  RI(J,NN)=0.0
    MX=NN*(NN)
    DO 13 J=1,MX
13  UI(J,NN)=0.0
11  CONTINUE
    DO 14 I=1,NU
14  U(I)=0.0
C
C  COMPUTE PARTIALS WPT STATE DERIVATIVES
C
    JJ=0
    DO 50 NN=1,NMAX
    MX=NN*(NN)
    DO 50 J=1,MX
    JJ=JJ+1
    XDOT(J,NN)=1.
    CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NNU,
1AT,RT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MFB,MST,
2MT,NUM,NRM,INIT,T)
    XDOT(J,NN)=0.
    DO 50 I=1,N
50  F(I,J)=V(I)
C
C  COMPUTE PARTIALS WPT INTERNAL OUTPUTS
C
    DO 100 NN=1,NMAX
    MX=NN*(NN)
    DO 100 J=1,MX
    JJ=JJ+1
    RI(J,NN)=1.
    CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NNU,
1AT,RT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MFB,MST,
2MT,NUM,NRM,INIT,T)
    RI(J,NN)=0.
    DO 100 I=1,N
100 F(I,J)=V(I)
C
C  COMPUTE PARTIALS WPT INTERNAL INPUTS
C
    DO 150 NN=1,NMAX
    MX=NN*(NN)
    DO 150 J=1,MX
    JJ=JJ+1
    UI(J,NN)=1.
    CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NNU,
1AT,RT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MFB,MST,
2MT,NUM,NRM,INIT,T)
    UI(J,NN)=0.
    DO 150 I=1,N
150 F(I,J)=V(I)
    IF(IPRINT.EQ.6)CALL DERUG(4,4HSTAM,4HK2 ,2.0,(W)
C
C  COMPUTE PARTIALS WPT STATES
C
    DO 200 NN=1,NMAX
    MX=NN*(NN)
    DO 200 J=1,MX
    JJ=JJ+1
    X(J,NN)=1.
    CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NNU,
1AT,RT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MFB,MST,
2MT,NUM,NRM,INIT,T)

```

Figure 20. Subroutine STAMK2 Program Listing (Continued)

# Contrails

	X(J,NX)=0.	STAMK197
	DO 201 I=1,N	STAMK198
201	F(I,J)=V(I)	STAMK199
	IF(IPRINT.EQ.6)CALL DEFBIG(5,4HSTAM,4HK2 ,2,0,TW)	STAMK200
C		STAMK201
C	COMPUTE PARTIALS WRT EXTERNAL INPUTS	STAMK202
C		STAMK203
	DO 251 J=1,NU	STAMK204
	JJ=JJ+1	STAMK205
	U(J)=!.	STAMK206
	CALL SIMKT(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,NNX,NNR,NNU,	STAMK207
	IAT,BT,CT,DT,PRINT,HS,P,Q,R,S,NX,NY,NR,NU,NMAX,MTR,MST,	STAMK208
	2MT,NUM,NRM,INIT,T)	STAMK209
	U(J)=!.	STAMK210
	DO 251 I=1,N	STAMK211
251	F(I,J)=V(I)	STAMK212
8002	CONTINUE	STAMK213
C		STAMK214
C	COMPUTE THE SIMULATION MATRIX	STAMK215
C		STAMK216
	NV=NX+NY	STAMK217
	IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,N,M,T,4HSIM )	STAMK218
	DO 51 I=1,NV	STAMK219
	DO 52 J=1,NV	STAMK220
52	F(I,J)=-F(I,J)	STAMK221
51	F(I,I)=F(I,I)+1.	STAMK222
	CALL TDINVR(ISOL,IOSOL,NV,-M,F,MAXN,KDUM,DET)	STAMK223
	IR=NV+1	STAMK224
	IE=NV+NR	STAMK225
	JB=IB	STAMK226
	JE=M	STAMK227
	DO 53 I=IR,IE	STAMK228
	DO 53 J=JB,JE	STAMK229
	DO 53 K=1,NV	STAMK230
53	F(I,J)=F(I,J)+F(I,K)*F(K,J)	STAMK231
	DO 53 I=1,IE	STAMK232
	DO 53 J=1,JE	STAMK233
	IF(ABS(F(I,J)).LE.FPSF) F(I,J) = 0.0	STAMK234
530	CONTINUE	STAMK235
	IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,N,M,T,4HSMI)	STAMK236
C		STAMK237
C	FORM A,B,C,D MATRICES	STAMK238
C		STAMK239
	J1=NV+1	STAMK240
	J2=NV+NX	STAMK241
	J3=J1+NX	STAMK242
	J4=J2+NU	STAMK243
	I1=NV+1	STAMK244
	I2=NV+NR	STAMK245
	DO 6001 I=1,NX	STAMK246
	DO 6001 J=J1,J2	STAMK247
	JJ=J-1+1	STAMK248
6001	A(I,J)=F(I,J)	STAMK249
	DO 6002 I=1,NX	STAMK250
	DO 6002 J=J3,J4	STAMK251
	JJ=J-1+1	STAMK252
6002	B(I,J)=F(I,J)	STAMK253
	DO 6003 I=I1,I2	STAMK254
	II=I-1+1	STAMK255
	DO 6003 J=J1,J2	STAMK256
	JJ=J-1+1	STAMK257
6003	C(II,J)=F(I,J)	STAMK258
	DO 6004 I=I1,I2	STAMK259
	II=I-1+1	STAMK260
	DO 6004 J=J3,J4	STAMK261
	JJ=J-1+1	STAMK262

Figure 20. Subroutine STAMK2 Program Listing (Continued)

# Contrails

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

Figure 20. Subroutine STAMK2 Program Listing (Concluded)

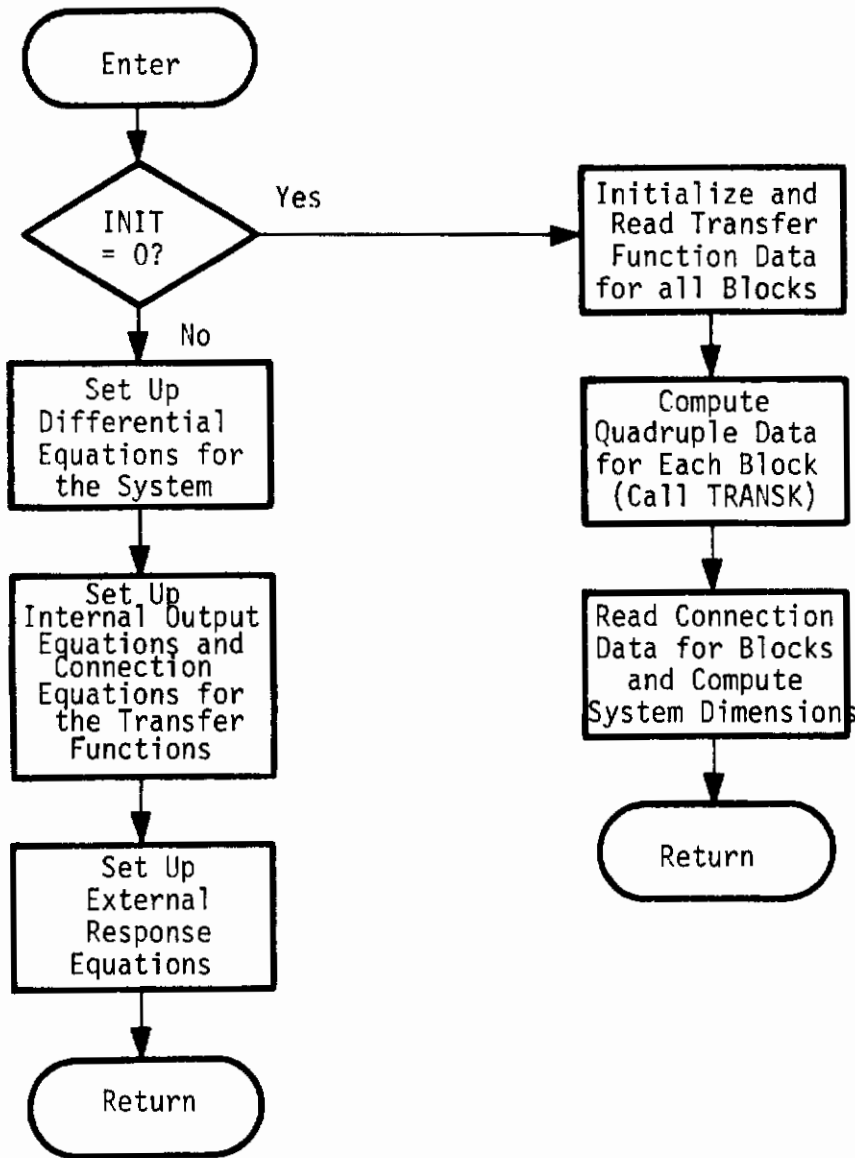


Figure 21. Subroutine SIMKT Flow Chart

# Contracts

```

SUBROUTINE SIMKT(XDOTL,YL,RL,XDOT,X,RI,UI,U,NNX,NNR,NUU,
1AT,BT,CT,DT,PRINT,MS,P,Q,R,S,NX,NY,NR,NU,NMAX,MTFB,MST,
2MT,NUM,NRM,INIT,T)
C
C PURPOSE - TO COMBINE SUBSYSTEM BLOCKS WHICH ARE DESCRIBED BY
C RATIONAL TRANSFER FUNCTIONS
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C DATE WRITTEN - 1975
C
C SUBPROGRAMS CALLED
C   DERUG
C   INPT
C   PHERR
C   TRANSK
C   DFN
C   MPRS
C   TPR
C   ZERO
C
C ARGUMENTS LIST
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   NMAX    INPUT    BLOCK NO OF TRANSFER FUNCTION
C   INIT    INPUT    INITIAL MODE FLAG
C   T       OUTPUT   SAMPLE TIME
C
C OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
DIMENSION XDOTL(NX),YL(NY),RL(NR)
DIMENSION XDOT(MST,MTFB),X(MST,MTFB),RI(1,MTFB),UI(1,MTFB)
DIMENSION U(NUM),NNX(MTFB),NNR(MTFB),NUU(MTFB)
DIMENSION AT(MST,MST,MTFB),AT(MST,1,MTFB)
DIMENSION CT(1,MST,MTFB),DT(1,1,MTFB)
DIMENSION P(MTFB,MTFB),Q(MTFB,NUM),R(NRM,MTFB),S(NRM,NUM)
DIMENSION PRINT(2,MT),MS(2,MT,MTFB)
DIMENSION CARD(20),IHEAD(3)
REAL IHEAD
DATA MC,MEND,MBLOC,МК,MDELA,HY/1MC,4HEND ,4MBLOC,1HK,4MDELA,1HY/
DATA HUIR,HI,HUIU,HRI,HRU/4HUI/R,4HI ,4HUI/U,4HR/R,4HR/U /
IF(IPRINT.EQ.6)CALL DEBUG(1,4MS1MK,4MT ,2,0,1W)
IF(INIT.NE.0) GO TO 100
C
C INITIALIZE
C
NBK=0
T=0.0
DO 1020 I=1,2
DO 1020 J=1,MT
DO 1020 K=1,MTFB
1020 HS(I,J,K)=0.0
CALL ZERO(P,MTFB,MTFB)
CALL ZERO(Q,MTFB,NUM)
CALL ZERO(R,NRM,MTFB)
CALL ZERO(S,NRM,NUM)
IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))WRITE(IW,1030)
1030 FORMAT(///,20X,4H*** TRANSFER FUNCTION DATA FOR BLOCKS ***,/)
1040 CONTINUE
READ(IR,1060)CARD

```

```

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

```

Figure 22. Subroutine SIMKT Program Listing

# Contrails

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 1480	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.HDELA).AND.(BKD2.EQ.HY))GO TO 1300	SIMKT 82
	IF(IPRINT.EQ.6)CALL DEBUG(2,4HSIMK,4HT ,2,0,IW)	SIMKT 83
C		SIMKT 84
C	READ RATIONAL TRANSFER FUNCTION DATA	SIMKT 85
C		SIMKT 86
	CALL ZERO(PRINT,2,MT)	SIMKT 87
	CALL INPT(PRINT,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 MESSEGE	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(PRINT,NN1,MT,IHEAD,T,IW)	SIMKT116
	GO TO 1040	SIMKT117
1300	CONTINUE	SIMKT118
	IF(IPRINT.EQ.6)CALL DEBUG(3,4HSIMK,4HT ,2,0,IW)	SIMKT119
C		SIMKT120
C	READ TIME DELAY SPECIFICATION	SIMKT121
C		SIMKT122
	READ(IW,1320)TD,XX,XR,UU,OMEGM,DELPHM,ND,NN	SIMKT123
1320	FORMAT(6E12,6,2I2)	SIMKT124
	IF(TD.GT.0.0)GO TO 1300	SIMKT125
	IF(UU.EQ.0.0)GO TO 1340	SIMKT126
	TD=(XX+XR)/UU	SIMKT127
	IF(TD.LT.0.0)GO TO 1340	SIMKT128
	GO TO 1300	SIMKT129
1340	CONTINUE	SIMKT130

Figure 22. Subroutine SIMKT Program Listing (Continued)



# Contrails

```
C
C PRINT ERROR MESSEGE
C
WRITE(IW,1360)
1360 FORMAT(1H1,/,/,1X,30HTIME DELAY SPECIFICATION ERROR)
STOP 111
1380 CONTINUE
IF(OMEGM.NE.0.0)GO TO 1400
IF(ND.EQ.0).OR.(NN.EQ.0)GO TO 1340
CALL OFN(HS,MT,MTFB,ND,NN,N,TD,IPRINT,IW)
NNX(N)=ND-1
GO TO 1200
1400 CONTINUE
IF(DELPHM.LE.0.0)GO TO 1340
DO 1420 ND=2,5
NNX(N)=ND-1
NDM=ND
IF(ND.EQ.5)NDM=4
DO 1420 NN=1,NDM
CALL OFN(HS,MT,MTFB,ND,NN,N,TD,IPRINT,IW)
CALL PHERR(HS,MT,MTFB,ND,N,OMEGM,TD,DELPH,IPRINT,IW)
IF(DELPH.LE.DELPHM)GO TO 1200
1420 CONTINUE
WRITE(IW,1430)DELPHM,DELPH
1430 FORMAT(1H1,/,/,1X,38HTIME DELAY SPECIFICATION CANNOT BE MET,/,
11X,20HALLOWED PHASE ERROR=,E12.6,/,
21X,20FACTUAL PHASE ERROR =,E12.6,/)
GO TO 1200
1440 CONTINUE
C
C PRINT ERROR MESSEGE
C
WRITE(IW,1460)
1460 FORMAT(1H1,/,/,1X,37HDATA CONTROL CARD SPECIFICATION ERROR)
STOP 111
1480 CONTINUE
NMAX=NBK
IF(IPRINT.EQ.6)CALL DEBUG(4.4HSIMK,4HT ,2.0,IW)
C
C COMPUTE QUADRUPLES FOR ALL BLOCKS
C
DO 1540 N=1,NMAX
CALL TRANSK(NNX,NNR,NNU,AT,BT,CT,DT,PRINT,HS,
IMST,MT,MTFB,N,NUM,NRM,IPRINT,IW)
1540 CONTINUE
NX=0
DO 1560 N=1,NMAX
NX=NX+NNX(N)
NNU(N)=1
1560 NNR(N)=1
NY=2*NMAX
C
C READ INTERCONNECTION QUADRUPLES AND PRINT THEM
C
1580 CONTINUE
READ(IR,1060)CARD
DECODE(4,1100,CARD(1))CC=DUMMY
IF(CC.EQ.HC)GO TO 1580
IF(CARD(1).EQ.HEND)GO TO 1600
IF((CARD(1).EQ.HUIR).AND.(CARD(2).EQ.HI))CALL INPT(P,MTFB,MTFB)
IF((CARD(1).EQ.HUIR).AND.(CARD(2).EQ.HI))GO TO 1580
IF(CARD(1).EQ.HUIU)CALL INPT(O,MTFB,NUM)
IF(CARD(1).EQ.HUIU)GO TO 1580
IF(CARD(1).EQ.HRRI)CALL INPT(R,NRM,MTFB)
IF(CARD(1).EQ.HRRI)GO TO 1580
IF(CARD(1).EQ.HRU)CALL INPT(S,NRM,NUM)
SIMKT131
SIMKT132
SIMKT133
SIMKT134
SIMKT135
SIMKT136
SIMKT137
SIMKT138
SIMKT139
SIMKT140
SIMKT141
SIMKT142
SIMKT143
SIMKT144
SIMKT145
SIMKT146
SIMKT147
SIMKT148
SIMKT149
SIMKT150
SIMKT151
SIMKT152
SIMKT153
SIMKT154
SIMKT155
SIMKT156
SIMKT157
SIMKT158
SIMKT159
SIMKT160
SIMKT161
SIMKT162
SIMKT163
SIMKT164
SIMKT165
SIMKT166
SIMKT167
SIMKT168
SIMKT169
SIMKT170
SIMKT171
SIMKT172
SIMKT173
SIMKT174
SIMKT175
SIMKT176
SIMKT177
SIMKT178
SIMKT179
SIMKT180
SIMKT181
SIMKT182
SIMKT183
SIMKT184
SIMKT185
SIMKT186
SIMKT187
SIMKT188
SIMKT189
SIMKT190
SIMKT191
SIMKT192
SIMKT193
SIMKT194
SIMKT195
SIMKT196
```

Figure 22. Subroutine SIMKT Program Listing (Continued)

# Contrails

```

        IF(CARD(1).EQ.HRU)GO TO 158)
        GO TO 1440
1600 CONTINUE
        IF(IPRINT.EQ.6)CALL DFRUG(5.4HSIMK.4HT ,2.0,IW)
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(1H1,/,/,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,MFB,MFB,NMAX,NMAX,T,4HP )
        CALL MPRS(Q,MFB,NUM,NMAX,NU,T,4HQ )
        CALL MPRS(R,NRM,MFB,NR,NMAX,T,4HR )
        CALL MPRS(S,NRM,NUM,NR,NU,T,4HS )
1860 CONTINUE
        RETURN
100 CONTINUE
C
C COMPUTE SUBSYSTEM STATES XDOT(N)=AN*KN +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)
        DO201 J=1,NUX
201 XDOTL(II)=XDOTL(II)+BT(I,J,N)*UI(J,N)
        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=NNR(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)*U(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)

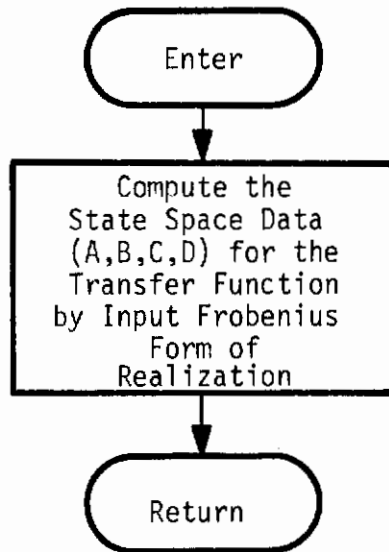


Figure 23. Subroutine TRANSK Flow Chart

# Contrails

```

SUBROUTINE TRANSK(NNX,NNR,NNU,AT,RT,CT,DT,PRINT,HS,
1MST,MT,MTRF,N,NUM,NRM,IPRINT,IW)
C
C   PURPOSE - TO COMPUTE QUADRUPLES FOR RATIONAL TRANSFER FUNCTION
C   ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C   DATE WRITTEN - 1976
C
C   SUBPROGRAMS CALLED
C   DEBUG
C
C   ARGUMENTS LIST
C   N           INPUT      TRANSFER FN BLOCK NO
C   IPRINT      INPUT      PRINT CONTROL FLAG
C   IW          INPUT      FILE NO FOR LINE PRINTED
C   OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
C   DIMENSION NNX(MTRF),NNR(MTRF),NNU(MTRF)
C   DIMENSION HS(2,MT,MTRF),PRINT(2,MT)
C   DIMENSION AT(MST,MST,MTRF),RT(MST,1,MTRF)
C   DIMENSION CT(1,MST,MTRF),DT(1,1,MTRF)
C
C   ZERO OUT STORAGE SPACE
C
C   IF (IPRINT.EQ.6) CALL DEBUG(1,4,TRAN,4H$K  *2,0,IW)
MX=NNX(N)
MXM1=NX(N)-1
MXP1=NX(N)+1
DO 1 I=1,MX
DO 1 J=1,MX
1  AT(I,1,N)=0.0
DO 2 I=1,MX
J=1
2  RT(I,1,N)=0.0
DO 3 I=1,MX
I=1
3  CT(I,1,N)=0.0
DT(1,1,N)=0.0
C
C   COMPUTE AT,RT
C
C   SET OFF DIAGONAL TERMS IN A TO UNITY
C
DO 50 I=1,MXM1
50  AT(I,I+1,N)=1.
RT(NNX(N),1,N)=1./HS(2,1,N)
C
C   COMPUTE LAST ROW OF A
C
C   IF (IPRINT.EQ.6) CALL DEBUG(2,4,TRAN,4H$K  *2,0,IW)
DO 10 J=1,MX
100  AT(NNX(N),J,N)=-HS(2,NNX(N)+2-J,N)*RT(NNX(N),1,N)
C
C   COMPUTE CT,DT
C
DO 20 J=1,MX
200  CT(1,1,N)=HS(1,NNX(N)+2-J,N)
IF (HS(1,1,N).EQ.0.0) GO TO 40
DO 30 J=1,MX
300  CT(1,1,N)=CT(1,J,N)+AT(NNX(N),J,N)*HS(1,1,N)
DT(1,1,N)=RT(NNX(N),1,N)*HS(1,1,N)
400  CONTINUE
IF (IPRINT.EQ.6) CALL DEBUG(3,4,TRAN,4H$K  *2,0,IW)
RETURN
END

```

Figure 24. Subroutine TRANSK Program Listing

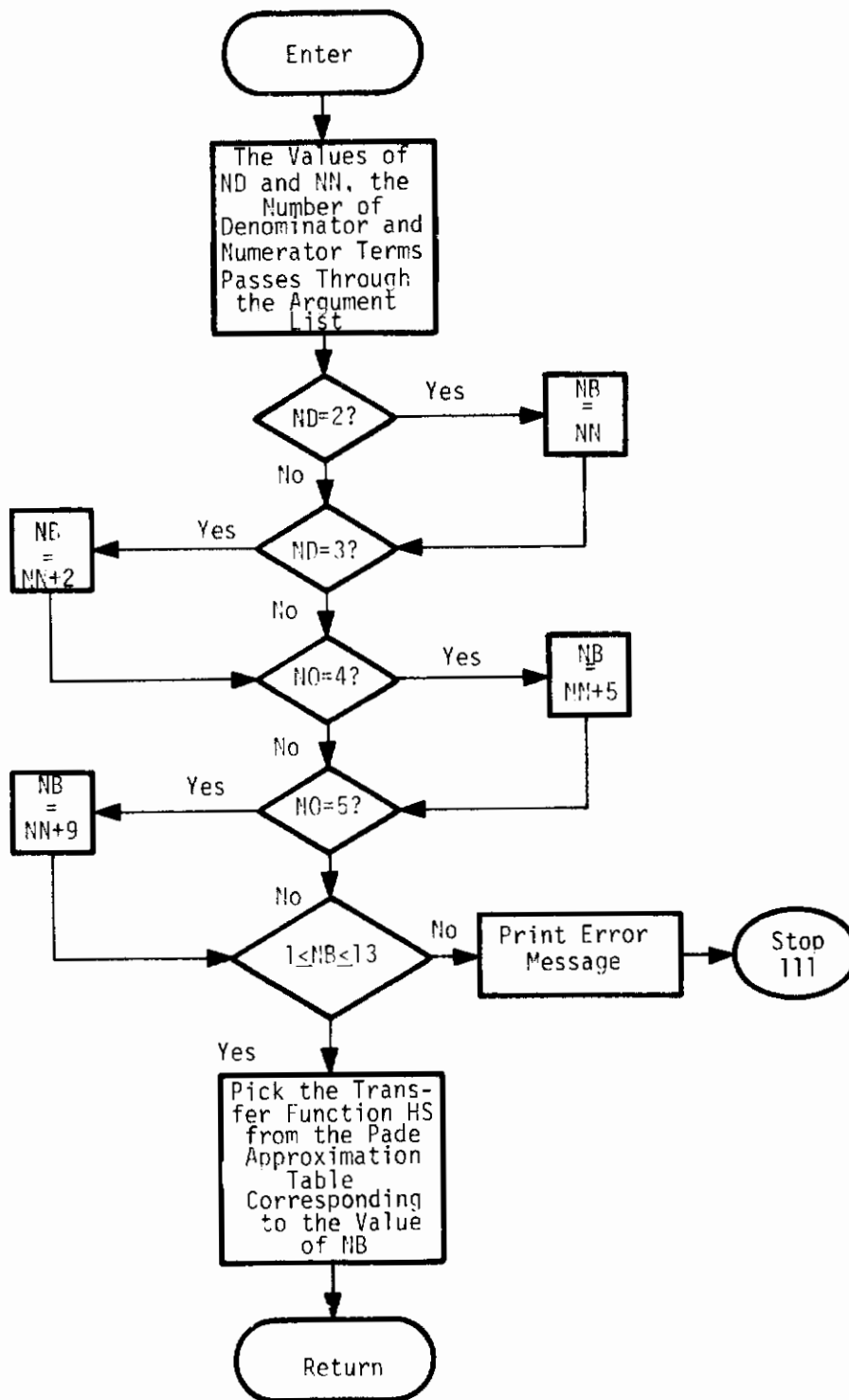


Figure 25. Subroutine DFN Flow Chart

# Contrails

```

SUBROUTINE DFN(HS,MT,MTR,N,NN,N,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 HONEYWELL 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.6)CALL DEBUG(1,4HDFN ,4H      +2,0,1W)   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 DEBUG(2,4HDFN ,4H      +2,0,1W)   DFN 24
C GO TO (510,520,530,540,550,560,570,580,590,600,610,620,630)NR DFN 25
C                                                     DFN 26
C FIRST ORDER PADE APPROXIMATIONS                      DFN 27
C                                                     DFN 28
C 510 CONTINUE                                         DFN 29
C   HS(1,2,N)=1.0                                       DFN 30
C   HS(2,1,N)=TD                                         DFN 31
C   HS(2,2,N)=1.0                                       DFN 32
C   GO TO 650                                           DFN 33
C 520 CONTINUE                                         DFN 34
C   HS(1,1,N)=-TD/2.0                                    DFN 35
C   HS(1,2,N)=1.0                                       DFN 36
C   HS(2,1,N)=TD/2.0                                    DFN 37
C   HS(2,2,N)=1.0                                       DFN 38
C   GO TO 650                                           DFN 39
C                                                     DFN 40
C SECOND ORDER PADE APPROXIMATIONS                    DFN 41
C                                                     DFN 42
C 530 CONTINUE                                         DFN 43
C   HS(1,2,N)=1.0                                       DFN 44
C   HS(2,1,N)=TD*TD/2.0                                   DFN 45
C   HS(2,2,N)=TD                                         DFN 46
C   HS(2,3,N)=1.0                                       DFN 47
C   GO TO 650                                           DFN 48
C 540 CONTINUE                                         DFN 49
C   HS(1,2,N)=-TD/3.0                                    DFN 50
C   HS(1,3,N)=1.0                                       DFN 51
C   HS(2,1,N)=TD*TD/6.0                                   DFN 52
C   HS(2,2,N)=2.0*TD/3.0                                  DFN 53
C   HS(2,3,N)=1.0                                       DFN 54
C   GO TO 650                                           DFN 55
C 550 CONTINUE                                         DFN 56
C   HS(1,1,N)=TD*TD/12.0                                  DFN 57
C   HS(1,2,N)=-TD/2.0                                    DFN 58
C   HS(1,3,N)=1.0                                       DFN 59
C   HS(2,1,N)=TD*TD/12.0                                  DFN 60
C   HS(2,2,N)=TD/2.0                                     DFN 61
C   HS(2,3,N)=1.0                                       DFN 62
C   GO TO 650                                           DFN 63
C                                                     DFN 64

```

Figure 26. Subroutine DFN Program Listing

# Contrails

```
C      THIRD ORDER PADE APPROXIMATIONS
C
560 CONTINUE
   HS(1.4,N)=1.0
   HS(2.1,N)=TD**3/6.0
   HS(2.2,N)=TD*TD/2.0
   HS(2.3,N)=TD
   HS(2.4,N)=1.0
   GO TO 650
570 CONTINUE
   HS(1.3,N)=-TD/4.0
   HS(2.1,N)=TD**3/24.0
   HS(1.4,N)=1.0
   HS(2.2,N)=TD*TD/4.0
   HS(2.3,N)=TD*3.0/4.0
   HS(2.4,N)=1.0
   GO TO 650
580 CONTINUE
   HS(1.2,N)=TD*TD/20.0
   HS(1.3,N)=-2.0*TD/5.0
   HS(1.4,N)=1.0
   HS(2.1,N)=TD**3/60.0
   HS(2.2,N)=3.0*TD*TD/20.0
   HS(2.3,N)=TD*3.0/5.0
   HS(2.4,N)=1.0
   GO TO 650
590 CONTINUE
   HS(1.1,N)=-TD**3/120.0
   HS(1.2,N)=TD*TD/10.0
   HS(1.3,N)=-TD/2.0
   HS(1.4,N)=1.0
   HS(2.1,N)=TD**3/120.0
   HS(2.2,N)=TD*TD/10.0
   HS(2.3,N)=TD/2.0
   HS(2.4,N)=1.0
   GO TO 650
C
C      FOURTH ORDER PADE APPROXIMATIONS
C
600 CONTINUE
   HS(1.5,N)=1.0
   HS(2.1,N)=TD**4/24.0
   HS(2.2,N)=TD**3/6.0
   HS(2.3,N)=TD*TD/2.0
   HS(2.4,N)=TD
   HS(2.5,N)=1.0
   GO TO 650
610 CONTINUE
   HS(1.4,N)=-TD/5.0
   HS(1.5,N)=1.0
   HS(2.1,N)=TD**4/120.0
   HS(2.2,N)=2.0*TD**3/30.0
   HS(2.3,N)=TD*TD*3.0/10.0
   HS(2.4,N)=TD*4.0/5.0
   HS(2.5,N)=1.0
   GO TO 650
620 CONTINUE
   HS(1.3,N)=TD*TD/30.0
   HS(1.4,N)=-TD/3.0
   HS(1.5,N)=1.0
   HS(2.1,N)=TD**4/360.0
   HS(2.2,N)=TD**3/30.0
   HS(2.3,N)=TD*TD*2.0/10.0
   HS(2.4,N)=TD*2.0/3.0
   HS(2.5,N)=1.0
   GO TO 650
DFN 65
DFN 66
DFN 67
DFN 68
DFN 69
DFN 70
DFN 71
DFN 72
DFN 73
DFN 74
DFN 75
DFN 76
DFN 77
DFN 78
DFN 79
DFN 80
DFN 81
DFN 82
DFN 83
DFN 84
DFN 85
DFN 86
DFN 87
DFN 88
DFN 89
DFN 90
DFN 91
DFN 92
DFN 93
DFN 94
DFN 95
DFN 96
DFN 97
DFN 98
DFN 99
DFN 100
DFN 101
DFN 102
DFN 103
DFN 104
DFN 105
DFN 106
DFN 107
DFN 108
DFN 109
DFN 110
DFN 111
DFN 112
DFN 113
DFN 114
DFN 115
DFN 116
DFN 117
DFN 118
DFN 119
DFN 120
DFN 121
DFN 122
DFN 123
DFN 124
DFN 125
DFN 126
DFN 127
DFN 128
DFN 129
DFN 130
```

Figure 26. Subroutine DFN Program Listing (Continued)



# Contrails

```
630 CONTINUE                                DFN 131
      HS(1.2,N)=-TD**3/210.0                 DFN 132
      HS(1.3,N)=TD*TD/14.0                  DFN 133
      HS(1.4,N)=-3.0*TD/7.0                 DFN 134
      HS(1.5,N)=1.0                          DFN 135
      HS(2.1,N)=TD**4/84.0                   DFN 136
      HS(2.2,N)=4.0*TD**3/210.0             DFN 137
      HS(2.3,N)=TD*TD*2.0/14.0              DFN 138
      HS(2.4,N)=TD*4.0/7.0                  DFN 139
      HS(2.5,N)=1.0                          DFN 140
650 CONTINUE                                DFN 141
      IF (JPRINT.EQ.6)CALL DEBUG(3,4HDFN ,4H   *2.0*IW) DFN 142
      RETURN                                  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,43HDIMENSIONS FOR TIME DELAY EXCEEDS THE LIMIT) DFN 149
      STOP 111                                 DFN 150
      END                                       DFN 151
```

Figure 26. Subroutine DFN Program Listing (Concluded)

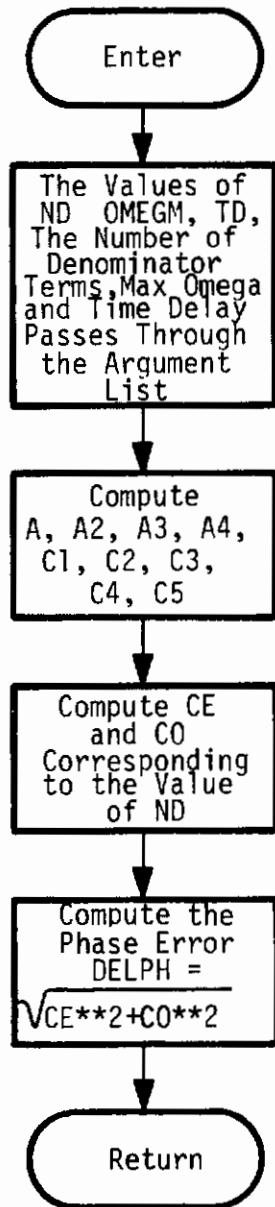


Figure 27. Subroutine PHERR Flow Chart

# Contrails

```

SUBROUTINE PHERR(HS,MT,MTR,ND,N,OMEGM,TD,DELPH,IPRINT,IW)          PHERR  2
C                                                                    PHERR  3
C PURPOSE - TO COMPUTE PHASE ERROR OF PADE APPROXIMATION TO TIME DEL PHERR  4
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC          PHERR  5
C DATE WRITTEN - 1975                                          PHERR  6
C                                                                    PHERR  7
C SUBPROGRAMS CALLED                                           PHERR  8
C   DEBUG                                                       PHERR  9
C                                                                    PHERR 10
C ARGUMENTS LIST                                               PHERR 11
C   ND           INPUT           NO OF DENOMINATOR TERMS IN THE TR FN PHERR 12
C   N            INPUT           TRANSFER FN BLOCK NO          PHERR 13
C   OMEGM        INPUT           MAXIMUM FREQUENCY FOR COMPUTING PHASE ERROR PHERR 14
C   TD           INPUT           TIME OR TRANSPORT DELAY          PHERR 15
C   DELPH        OUTPUT          PHASE ERROR                      PHERR 16
C   IPRINT       INPUT           PRINT CONTROL FLAG              PHERR 17
C   IW           INPUT           FILE NO FOR LINE PRINTER        PHERR 18
C OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM              PHERR 19
C                                                                    PHERR 20
C DIMENSION HS(2,MT,MTR)                                       PHERR 21
C IF (IPRINT.EQ.6)CALL DEBUG(1,4,PHERR,4HR   *2,0,IW)          PHERR 22
C A=OMEGM*TD                                                    PHERR 23
C A2=A**2                                                       PHERR 24
C A3=A**3                                                       PHERR 25
C A4=A**4                                                       PHERR 26
C C1=HS(2,1,N)*A-HS(1,1,N)                                     PHERR 27
C C2=HS(2,2,N)*A-HS(1,2,N)                                     PHERR 28
C C3=HS(2,3,N)*A-HS(1,3,N)                                     PHERR 29
C C4=HS(2,4,N)*A-HS(1,4,N)                                     PHERR 30
C C5=HS(2,5,N)*A-HS(1,5,N)                                     PHERR 31
C NO=ND-1                                                       PHERR 32
C IF (IPRINT.EQ.6)CALL DEBUG(2,4,PHERR,4HR   *2,0,IW)          PHERR 33
C GO TO(110,120,130,140)NO                                       PHERR 34
110 CONTINUE                                                    PHERR 35
C CE=C2                                                         PHERR 36
C CO=C1*A                                                       PHERR 37
C GO TO 150                                                       PHERR 38
120 CONTINUE                                                    PHERR 39
C CE=C3-C1*A2                                                   PHERR 40
C CO=C2*A                                                       PHERR 41
C GO TO 150                                                       PHERR 42
130 CONTINUE                                                    PHERR 43
C CE=C4-C2*A2                                                   PHERR 44
C CO=C3*A-C1*A3                                               PHERR 45
C GO TO 150                                                       PHERR 46
140 CONTINUE                                                    PHERR 47
C CE=C5-C3*A2+C1*A4                                           PHERR 48
C CO=C4*A-C2*A3                                               PHERR 49
150 CONTINUE                                                    PHERR 50
C CE2=CE**2                                                    PHERR 51
C CO2=CO**2                                                    PHERR 52
C DELPH=SQR(CE2+CO2)                                           PHERR 53
C IF (IPRINT.EQ.6)CALL DEBUG(3,4,PHERR,4HR   *2,0,IW)          PHERR 54
C RETURN                                                       PHERR 55
C END                                                            PHERR 56

```

Figure 28. Subroutine PHERR Program Listing

# Contrails

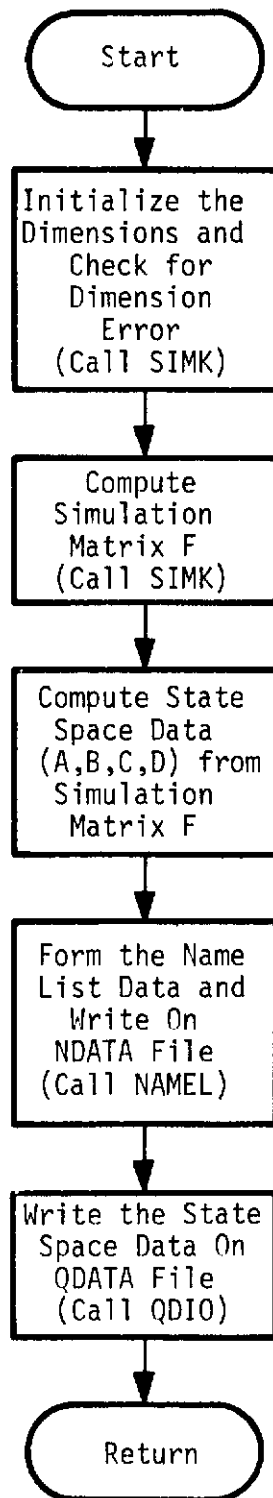


Figure 29. Subroutine STAMK3 Flow Chart

# Contrails

```
SUBROUTINE STAMK3(V,W,F,XDOT,X,RI,UI,J,RIN,NNX,NNR,NNU,
1A,B,C,D,NNS,VNS,DESS,UNITS,VNO,VNO,DESO,UNITO,
2NNI,VNI,DESI,UNITI,MAXN,MAXM,NXM,NRM,NUM,NYM,
3MH,MN,MM,MP,MQ,MR,MS1,MS2,MS3,MS4,NB,VRMMR)
C
C PURPOSE - TO OBTAIN STATE SPACE MODEL FROM INTERCONNECTION
C DATA FOR SUBSYSTEMS OR TO READ DIRECTLY THE STATE SPACE DATA
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C DATE WRITTEN - 1975
C
C SUBPROGRAMS CALLED
C DEPRM
C HPD
C NAMEL
C QUADK
C TDINVR
C DEGRMS
C MPQS
C QDID
C SI *K
C
C ARGUMENTS LIST
C V V ARRAY FOR COMPUTING SIMULATION MATRIX
C W W ARRAY FOR COMPUTING SIMULATION MATRIX
C F SIMULATION MATRIX
C XDOT ARRAY FOR STATE DERIVATIVES
C X ARRAY FOR STATES
C RI ARRAY FOR INTERNAL OUTPUTS
C UI ARRAY FOR INTERNAL INPUTS
C U ARRAY FOR EXTERNAL INPUTS
C RIN ARRAY FOR INTERNAL OUTPUTS FOR ALL SYSTEMS
C NNX ARRAY FOR STORING SYSTEM DIMENSION NX
C NNR ARRAY FOR STORING SYSTEM DIMENSION NR
C NNU ARRAY FOR STORING SYSTEM DIMENSION NU
C A IN/OUT STATE TRANSITION MATRIX
C B 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 VNO IN/OUT NUMBER ARRAY FOR OUTPUT
C DESO IN/OUT VARIABLE NAME 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 MN INPUT MM*MR
C MM INPUT MAXIMUM OF (NR,NUM)
C MP INPUT MAXIMUM DIMENSION FOR P ARRAY
C MQ INPUT MAXIMUM DIMENSION FOR Q ARRAY
C MR INPUT MAXIMUM DIMENSION FOR R ARRAY
C MS1 INPUT MAXIMUM DIMENSION FOR SCRATCH ARRAY S1
C STAMK3 2
C STAMK3 3
C STAMK3 4
C STAMK3 5
C STAMK3 6
C STAMK3 7
C STAMK3 8
C STAMK3 9
C STAMK310
C STAMK311
C STAMK312
C STAMK313
C STAMK314
C STAMK315
C STAMK316
C STAMK317
C STAMK318
C STAMK319
C STAMK320
C STAMK321
C STAMK322
C STAMK323
C STAMK324
C STAMK325
C STAMK326
C STAMK327
C STAMK328
C STAMK329
C STAMK330
C STAMK331
C STAMK332
C STAMK333
C STAMK334
C STAMK335
C STAMK336
C STAMK337
C STAMK338
C STAMK339
C STAMK340
C STAMK341
C STAMK342
C STAMK343
C STAMK344
C STAMK345
C STAMK346
C STAMK347
C STAMK348
C STAMK349
C STAMK350
C STAMK351
C STAMK352
C STAMK353
C STAMK354
C STAMK355
C STAMK356
C STAMK357
C STAMK358
C STAMK359
C STAMK360
C STAMK361
C STAMK362
C STAMK363
C STAMK364
```

Figure 30. Subroutine STAMK3 Program Listing

# Contrails

```

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      NH       INPUT      MAXIMUM SYSTEM NO - IMPLICIT MODEL        STAMK368
C      NRM*MB   INPUT      MAXIMUM DIMENSION FOR RIN                  STAMK369
C                                                    STAMK370
COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS      STAMK371
COMMON /SYS/ SCODE,SDEF(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20)         STAMK372
I,PHEAD(20)                                                            STAMK373
COMMON /SC1/ S1(1)                                                    STAMK374
C      DIMENSION DESS(NX*10,MB),UNITSS(NX*4,4,MB)                    STAMK375
C      DIMENSION DESO(NRM*10,MB),UNITO(NRM*4,MB)                    STAMK376
C      DIMENSION DESI(NUM*10,MB),UNITI(NUM*4,MB)                     STAMK377
C      DIMENSION NXX(MB),NRR(MB),NUU(MB)                              STAMK378
COMMON /SC2/ S2(1)                                                    STAMK379
C      DIMENSION ATC(NX*NX*MB),RTC(NX*NUM,MB)                        STAMK380
C      DIMENSION CTC(NRM*NX*MB),DTC(NRM,NUM,MB)                     STAMK381
C      DIMENSION PC(MN,MN),QC(MN,NJM),RC(NRM,MN),SC(NM*NUM)         STAMK382
COMMON /SC3/ S3(1)                                                    STAMK383
C      DIMENSION PP(MP,MM,MM),QQ(MQ*MM,NUM),RR(MR,NRM,MM)           STAMK384
C      DIMENSION NSP(MP),NSQ(MQ),NSR(MR)                              STAMK385
C      DIMENSION V(MAXN),W(MAXN),F(MAXN*MAXN)                        STAMK386
C      DIMENSION XDOT(NX*MB),X(NX*MB),RI(NRM,MB),UI(NUM,MB)         STAMK387
C      DIMENSION RIN(NRM*MB),II(NUM),NXX(MB),NRR(MB),NNI(MB)        STAMK388
C      DIMENSION A(NX*NX),R(NX*NUM),C(NRM,NX),D(NRM,NUM)            STAMK389
C      DIMENSION NNS(NX),VNS(NX*2),DESS(NX*10),UNITS(NX*4)          STAMK390
C      DIMENSION NNO(NRM),VNO(NRM*2),DESO(NRM*10),UNITO(NRM*4)     STAMK391
C      DIMENSION NNI(NUM),VNI(NUM*2),DESI(NUM*10),UNITI(NUM*4)      STAMK392
C      DATA HDINT/4HSINT/                                            STAMK393
C                                                    STAMK394
C      PRINT SYSTEM DIMENSIONS IF NEEDED                               STAMK395
C                                                    STAMK396
C      IF(IPRINT.EQ.6)WRITE(IW,165)MS1,MS2,MS3,MS4,MAXN,MAXM        STAMK397
1,NX,NRM,NRM*NUM,NYM,MM,MP,MQ,MR,MR,NR,MS,MM                          STAMK398
165 FORMAT(1X,15(I5,1X))                                             STAMK399
C                                                    STAMK100
C      COMPUTE ARRAY START INDEXES                                    STAMK101
C                                                    STAMK102
C      FOR DESS,UNITSS,DESO,UNITO,DESI,UNITI,NXX,NRR,NUU           STAMK103
C                                                    STAMK104
C      L1=1 $ L2=L1+NX*MB*10 $ L3=L2+NX*MB*4 $ L4=L3+NRM*MB*10     STAMK105
C      L5=L4+NRM*MB*4 $ L6=L5+NUM*MB*10 $ L7=L6+NUM*MR*4           STAMK106
C      L8=L7+MB $ L9=L8+MR $ L10=L9+MB                                STAMK107
C                                                    STAMK108
C      FOR ATC,RTC,CTC,DTC,PC,QC,RC,SC                               STAMK109
C                                                    STAMK110
C      M1=1 $ M2=M1+NX*NX*MB $ M3=M2+NX*NUM*MB $ M4=M3+NRM*NX*MB   STAMK111
C      M5=M4+NRM*NUM*MB $ M6=M5+MN*MN $ M7=M6+NM*NUM $ M8=M7+NRM*MY STAMK112
C      M9=M8+NRM*NUM                                                 STAMK113
C                                                    STAMK114
C      FOR PP,QQ,RR,NSP,NSQ,NSR                                       STAMK115
C                                                    STAMK116
C      K1=1 $ K2=K1+MP*MM*MM $ K3=K2+MQ*MM*NUM $ K4=K3+MR*NRM*MM   STAMK117
C      K5=K4+MP $ K6=K5+MQ $ K7=K6+MR                                STAMK118
C      IF(IPRINT.EQ.6)WRITE(IW,165)L1,L2,L3,L4,L5,L6,L7,L8,L9,L10    STAMK119
C      IF(IPRINT.EQ.6)WRITE(IW,165)M1,M2,M3,M4,M5,M6,M7,M8,M9       STAMK120
C      IF(IPRINT.EQ.6)WRITE(IW,165)K1,K2,K3,K4,K5,K6,K7             STAMK121
C                                                    STAMK122
C      CHECK IF SCRATCH AFRAY SIZES ARE SUFFICIENT                  STAMK123
C                                                    STAMK124
C      IF((L10.GT.MS1).OR.(M9.GT.MS2).OR.(K7.GT.MS3))              STAMK125
C      ICALL DERRM(L10,M9,K7,MS4,MS1,MS2,MS3,MS4,3,0,4HSTAM,4HK3 ,IW) STAMK126
C      IF(SCODE.EQ.HDINT)GO TO 5                                       STAMK127
C      CALL QUADK(A,B,C,D,NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,    STAMK128
C      INNT,VNT,DESI,UNITI,S1(L1),S1(L2),S1(L3),S1(L4),S1(L5),S1(L6), STAMK129
C      STAMK130

```

Figure 30. Subroutine STAMK3 Program Listing (Continued)

# Contrails

```

2S1(L7),S1(L8),S1(L9),NXM,NRM,NUM,MB,NB1
RETURN
5 CONTINUE
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 $ INIT=0 $ NFLAG=1
IF((IPRINT.EQ.3).OR.(IPRINT.GT.4))CALL HPR(HEAD,IW)
C
C INITIALIZING CALL TO SUBROUTINE SIMK
C
NX=0 $ NR=0 $ NU=0 $ NY=0
N1=1 $ N2=N1+NX $ N3=N2+NY
CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NUU,
1S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
2S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,
3MB,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,NXA,NRA,NUA,NB,NRMMB)
C
C CHECK FOR DIMENSION ERROR
C
INIT = 1
M=2*NX+NY+NU
N=NX+NY+NR
IF((NX.GT.NXM).OR.(NR.GT.NRM).OR.(NU.GT.NUM).OR.(NY.GT.NYM))
1CALL DERRMS(NX,NR,NU,NY,NXM,NRM,NUM,NYM,3,0,4HSTAM,4HKJ ,IW)
N1=1 $ N2=N1+NX $ N3=N2+NY
C
C ZERO OUT XDOT,RI,UI,X,U
C
DO 10 NN=1,NMAX
MX=NNX(NN)
DO 10 J=1,MX
XDOT(J,NN)=0.0
10 X(J,NN)=0.0
DO 11 NN=1,NMAX
MX=NNR(NN)
DO 12 J=1,MX
12 RI(J,NN)=0.0
MX=NNU(NN)
DO 13 J=1,MX
13 UI(J,NN)=0.0
11 CONTINUE
DO 14 I=1,NU
14 U(I)=0.0
C
C COMPUTE PARTIALS WRT STATE DERIVATIVES
C
45 CONTINUE
JJ=0
DO 50 NN=1,NMAX
MX=NNX(NN)
DO 50 J=1,MX
JJ=JJ+1
XDOT(J,NN)=1.
CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NUU,
1S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
2S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,
3MB,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,NXA,NRA,NUA,NB,NRMMB)
XDOT(J,NN)=0.
DO 50 I=1,N
50 F(I,JJ)=V(I)
C
C COMPUTE PARTIALS WRT INTERNAL OUTPUTS
C
DO 100 NN=1,NMAX
MX=NNR(NN)
DO 100 J=1,MX

```

Figure 30. Subroutine STAMK3 Program Listing (Continued)

# Contrails

```

JJ=JJ+1
RI(J,NN)=1.
CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,
1S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
2S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,
3MB,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,XXA,NRA,NIJA,NR,NRMMB)
RI(J,NN)=0.
DO 10 I=1,N
100 F(I,J)=V(I)
C
C COMPUTE PARTIALS WPT INTERNAL INPUTS
C
DO 15 NN=1,NMAX
MX=NN*(NN)
DO 15 J=1,MX
JJ=JJ+1
UI(J,NN)=1.
CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,
1S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
2S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,
3MB,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,XXA,NRA,NIJA,NB,NRMMB)
UI(J,NN)=0.
DO 15 I=1,M
150 F(I,J)=V(I)
C
C COMPUTE PARTIALS WPT STATES
C
DO 201 NN=1,NMAX
MX=NN*(NN)
DO 201 J=1,MX
JJ=JJ+1
X(J,NN)=1.
CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,
1S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
2S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,
3MB,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,XXA,NRA,NIJA,NB,NRMMB)
X(J,NN)=0.
DO 201 I=1,N
201 F(I,J)=V(I)
C
C COMPUTE PARTIALS WPT EXTERNAL INPUTS
C
DO 251 J=1,NU
JJ=JJ+1
U(J)=1.
CALL SIMK(V(N1),V(N2),V(N3),XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,
1S2(M1),S2(M2),S2(M3),S2(M4),S2(M5),S2(M6),S2(M7),S2(M8),
2S3(K1),S3(K2),S3(K3),S3(K4),S3(K5),S3(K6),NX,NY,NR,NU,NMAX,
3MB,MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,XXA,NRA,NIJA,NB,NRMMB)
U(J)=0.
DO 251 I=1,N
251 F(I,J)=V(I)
8002 CONTINUE
C
C COMPUTE THE SIMULATION MATRIX
C
NV=NX*NY
IF(IPPINT.EQ.6)CALL MPRS(F,MAXN,MAXM,N,M,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.
C
C XDOT ARRAY IS BEING USED AS A SCRATCH ARRAY IN TDINVR
C
CALL TDINVR(ISOL,INSOL,NV,-M,F,MAXN,XDOT,DET)

```

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 30. Subroutine STAMK3 Program Listing (Continued)



# Contracts

```

IR=NV+1
IF=NV+NR
JR=IR
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
530 CONTINUE
IF (IPRINT.EQ.6) CALL MPRS(F,MAXN,MAXM,N,M,T,4HSIMI)
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 6001 I=1,NX
DO 6001 J=J1,J2
JJ=J-J1+1
6001 A(I,J)=F(I,J)
DO 6002 I=1,NX
DO 6002 J=J3,J4
JJ=J-J3+1
6002 B(I,J)=F(I,J)
DO 6003 I=I1,I2
II=I-I1+1
DO 6003 J=J1,J2
JJ=J-J1+1
6003 C(II,JJ)=F(I,J)
DO 6004 I=I1,I2
II=I-I1+1
DO 6004 J=J3,J4
JJ=J-J3+1
6004 D(II,JJ)=F(I,J)
C
C UPDATE NAME LIST DATA
C
KR=NMAX
CALL NAMEL(NNS,VNS,DESS,UNITS,NN0,VN0,DES0,UNIT0,NNI,VNI,
1DES1,UNIT1,S1(L1),S1(L2),S1(L3),S1(L4),S1(L5),S1(L6),
2S1(L7),S1(L8),S1(L9),NXM,NRM,NUM,NX,NR,NU,NFLAG,MR,KB,NB)
C
C WRITE QUADRUPLE DATA ON FILE QDATA
C
IQ=0
MFLAG=2
CALL QDIO(A,B,C,D,S1,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA,
1NR1,NR2,NR3,NU1,NU2,NU3,T,IQ,IPRINT,IW,JQ,HEAD,MARK,
2LOCATE, NULL,INSFRT,MFLAG)
RETURN
END
STAMK263
STAMK264
STAMK265
STAMK266
STAMK267
STAMK268
STAMK269
STAMK270
STAMK271
STAMK272
STAMK273
STAMK274
STAMK275
STAMK276
STAMK277
STAMK278
STAMK279
STAMK280
STAMK281
STAMK282
STAMK283
STAMK284
STAMK285
STAMK286
STAMK287
STAMK288
STAMK289
STAMK290
STAMK291
STAMK292
STAMK293
STAMK294
STAMK295
STAMK296
STAMK297
STAMK298
STAMK299
STAMK300
STAMK301
STAMK302
STAMK303
STAMK304
STAMK305
STAMK306
STAMK307
STAMK308
STAMK309
STAMK310
STAMK311
STAMK312
STAMK313
STAMK314
STAMK315
STAMK316
STAMK317
STAMK318
STAMK319

```

Figure 30. Subroutine STAMK3 Program Listing (Concluded)

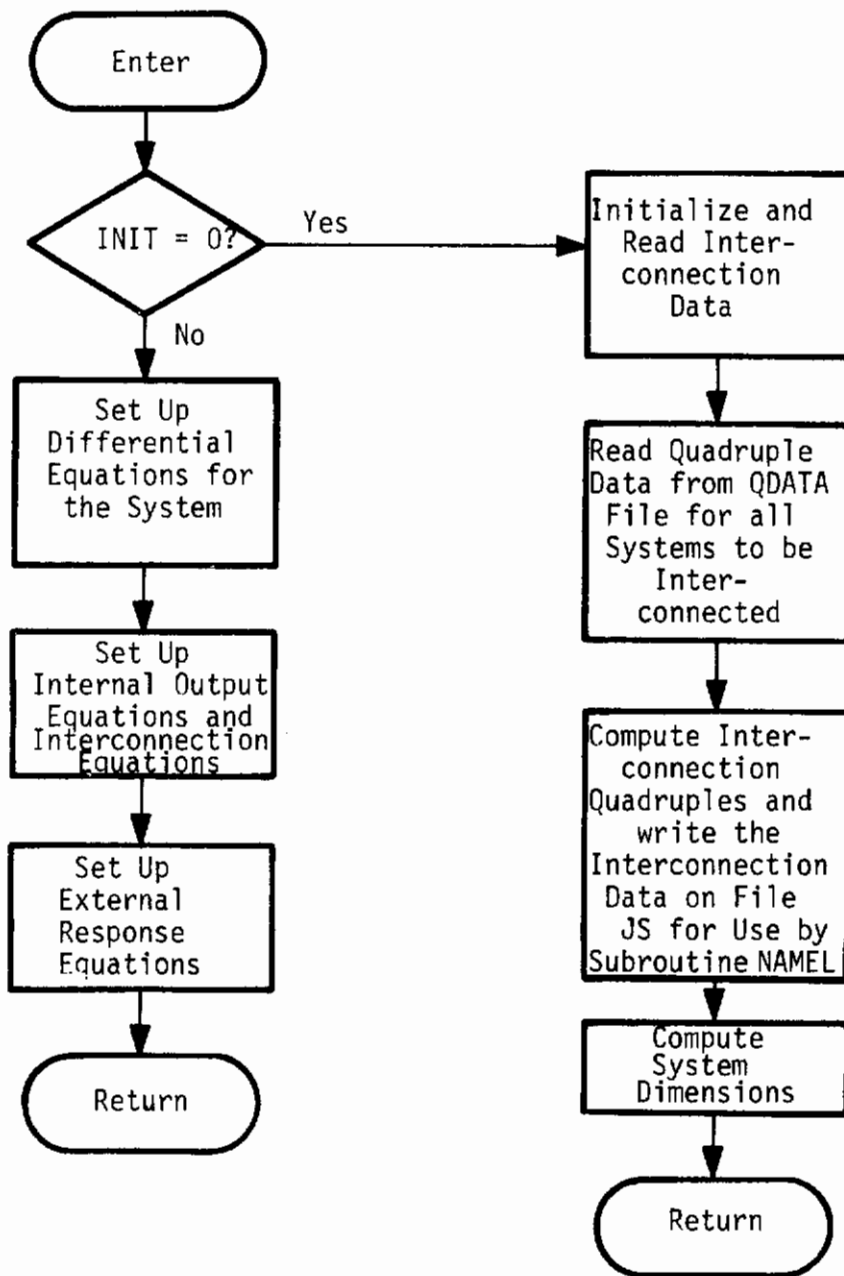


Figure 31. Subroutine SIMK Flow Chart

# Contrails

```

SUBROUTINE SIMK(XDOTL,YL,RL,XDOT,X,RI,UI,U,RIN,NNX,NNR,NNU,      SIMK  2
1AT,BT,CT,DT,P,Q,R,S,PP,QQ,RR,NSP,NSQ,NSR,NX,NY,NR,NU,KB,MR,    SIMK  3
2MM,MP,MQ,MR,NXM,NUM,NRM,MN,INIT,T,NXA,NRA,NUA,NB,NRMMB)        SIMK  4
C                                                                    SIMK  5
C   PURPOSE - TO COMBINE SYSTEMS DESCRIBED BY QUADRUPLES        SIMK  6
C   ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC        SIMK  7
C   DATE WRITTEN - 1975                                          SIMK  8
C                                                                    SIMK  9
C   SUBPROGRAMS CALLED                                           SIMK 10
C   FILE                                                         SIMK 11
C   MPDS                                                         SIMK 12
C   INPT                                                         SIMK 13
C   ZEP0                                                         SIMK 14
C                                                                    SIMK 15
C   ARGUMENTS LIST                                              SIMK 16
C   XDOTL      OUTPUT      ARRAY FOR DERIVATIVE OF STATE        SIMK 17
C   YL         OUTPUT      ARRAY FOR Y EQUATION VARIABLES        SIMK 18
C   RL         OUTPUT      ARRAY FOR EXTERNAL RESPONSE VARIABLES SIMK 19
C   AT         OUTPUT      ARRAY FOR STORING SUBSYSTEM QUADRUPLE A SIMK 20
C   BT         OUTPUT      ARRAY FOR STORING SUBSYSTEM QUADRUPLE B SIMK 21
C   CT         OUTPUT      ARRAY FOR STORING SUBSYSTEM QUADRUPLE C SIMK 22
C   DT         OUTPUT      ARRAY FOR STORING SUBSYSTEM QUADRUPLE D SIMK 23
C   P          OUTPUT      INTERCONNECTION QUADRUPLE             SIMK 24
C   Q          OUTPUT      INTERCONNECTION QUADRUPLE             SIMK 25
C   R          OUTPUT      INTERCONNECTION QUADRUPLE             SIMK 26
C   S          OUTPUT      INTERCONNECTION QUADRUPLE             SIMK 27
C   PP         OUTPUT      ARRAY FOR INTERMEDIATE INTERCONN DATA SIMK 28
C   QQ         OUTPUT      ARRAY FOR INTERMEDIATE INTERCONN DATA SIMK 29
C   RR         OUTPUT      ARRAY FOR INTERMEDIATE INTERCONN DATA SIMK 30
C   NSP        OUTPUT      ARRAY FOR INTERMEDIATE INTERCONN DATA SIMK 31
C   NSQ        OUTPUT      ARRAY FOR INTERMEDIATE INTERCONN DATA SIMK 32
C   NSR        OUTPUT      ARRAY FOR INTERMEDIATE INTERCONN DATA SIMK 33
C   NX         OUTPUT      NUMBER OF STATES                       SIMK 34
C   NY         OUTPUT      NUMBER OF Y EQUATIONS                 SIMK 35
C   NR         OUTPUT      NUMBER OF OUTPUTS                     SIMK 36
C   NU         OUTPUT      NUMBER OF INPUTS                       SIMK 37
C   INIT       INPUT       INITIAL MODE FLAG                     SIMK 38
C   T         OUTPUT      SAMPLE TIME                             SIMK 39
C   NXA        OUTPUT      NO OF STATES WITHOUT THE IMPLICIT MODEL SIMK 40
C   NRA        OUTPUT      NO OF OUTPUTS WITHOUT THE IMPLICIT MODEL SIMK 41
C   NUA        OUTPUT      NO OF INPUTS WITHOUT THE IMPLICIT MODEL SIMK 42
C   OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM              SIMK 43
C                                                                    SIMK 44
C   COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS SIMK 45
C   COMMON /SYS/  SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20) SIMK 46
C   I,PHEAD(20)                                                 SIMK 47
C   DIMENSION XDOTL(NX),YL(NY),RL(NR)                           SIMK 48
C   DIMENSION XDOT(NXM,MB),X(NXM,MB),RI(NRM,MB),UI(NUM,MB)     SIMK 49
C   DIMENSION RIN(NRMMB),U(NUM),NNX(MB),NNR(MB),NNU(MB)         SIMK 50
C   DIMENSION AT(NXM,NXM,MB),BT(NXM,NUM,MB)                     SIMK 51
C   DIMENSION CT(NRM,NXM,MR),DT(NRM,NUM,MB)                     SIMK 52
C   DIMENSION P(MN,MN),Q(MN,NUM),R(NRM,MN),S(NRM,NUM)          SIMK 53
C   DIMENSION PP(MP,MM,MM),QQ(MQ,MM,NUM),RR(MR,NRM,MM)         SIMK 54
C   DIMENSION NSP(MP),NSQ(MQ),NSR(MR)                           SIMK 55
C   DIMENSION CARD(20)                                           SIMK 56
C   DATA HC,HEND,HRI,HRS,HUI,HU/1HC,4HEND,2HRI,2HR/,2HUI,1HU/ SIMK 57
C   DATA HOUTP,HINPU/4HOUTP,4HINPU/                             SIMK 58
C   IF(INIT.NE.0) GO TO 100                                       SIMK 59
C                                                                    SIMK 60
C   INITIALIZE                                                  SIMK 61
C                                                                    SIMK 62
C   KP=0 & KQ=0 & KR=0                                          SIMK 63
C   KR=0                                                         SIMK 64

```

Figure 32. Subroutine SIMK Program Listing

# Contrails

T=0.0	SIMK 65
NX1=0 \$ NR1=0 \$ NU1=0	SIMK 66
IF(IP>INT.EQ.6)WRITE(1W,900)KR,MB,MY,MP,MQ,MR,MS,NUM,NRM,MN,NB	SIMK 67
900 FORMAT(15(1X,12))	SIMK 68
DO 96 I=1,NB	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 1000 K=1,MM	SIMK 82
DO 1000 I=1,MP	SIMK 83
1000 PP(J,I,K)=P(J,K)	SIMK 84
DO 1010 K=1,NUM	SIMK 85
DO 1010 I=1,MQ	SIMK 86
1010 QQ(I,I,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,I,K)=R(J,K)	SIMK 91
C	SIMK 92
C READ INTERCONNECTION DATA	SIMK 93
C	SIMK 94
1040 CONTINUE	SIMK 95
READ(1R,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 1480	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,KR	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.HR1)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.HR1)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)

# Contrails

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,NUMMY	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 1260 I=1,NRM	SIMK 142
	DO 1260 J=1,MM	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,MM	SIMK 163
	DO 1340 J=1,MM	SIMK 164
1340	PP(KP,I,J)=P(I,J)	SIMK 165
	IF(I.PRINT.LT.6)GO TO 1040	SIMK 166
	WRITE(IW,900)KP,NSY,NSP	SIMK 167
	CALL MPRS(P,MN,MN,MM,MM,0.0,0.4HPP )	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,MM	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(1H1,///,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(1H1,///,1X,30HTOO 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)

# Contrails

```

1475 FORMAT(1H1,/,/,1X,39HTOO MANY INTERCONNECTIONS FOR COMBINING)
      STOP 111
C
C   OBTAIN QUADRUPLE DATA FOR SUBSYSTEMS FROM Q DATA FILE
C
1480 CONTINUE
      DO 1490 I=1,20
1490 CARD(I)=HEAD(I)
      DO 1500 N=1,KB
          NSY=NSYS(N)
          DO 1510 I=1,20
              HEAD(I)=SHEAD(NSY,I)
1500 CONTINUE
          CALL FILE(JQ,LOCATE,HEAD)
          READ(JQ)T,NNXN,NNRN,NNUN,
1              ((AT(I,J,N),I=1,NNXN),J=1,NNXN),
2              ((BT(I,J,N),I=1,NNXN),J=1,NNUN),
3              ((CT(I,J,N),I=1,NNRN),J=1,NNXN),
4              ((DT(I,J,N),I=1,NNRN),J=1,NNUN)
          NNX(N)=NNXN
          NNR(N)=NNRN
          NNU(N)=NNUN
C
C   STORE THE IMPLICIT MODEL SYSTEM DIMENSIONS SEPARATELY
C
      IF(NSY.NE.NB)GO TO 1510
      NX1=NNXN
      NR1=NNRN
      NU1=NNUN
1510 CONTINUE
1520 CONTINUE
      DO 1530 I=1,20
1530 HEAD(I)=CARD(I)
C
C   FORM INTERCONNECTION QUADRUPLES
C
      CALL ZERO(P,MN,MN)
      CALL ZERO(Q,MN,NUM)
      CALL ZERO(R,NRM,MN)
C
C   FORM R MATRIX (R/R1)
C
      KYOUT=0
      NM1=1
      NM2=0
      DO 1565 M=1,KB
          KYOUT=KYOUT+NNR(M)
          IF(M.GT.1)NM1=NM1+NNR(M-1)
          NM2=NM2+NNR(M)
          NSY2=NSYS(M)
          DO 1533 KR=1,MR
              IF(NSR(KR).EQ.NSY2)GO TO 1536
1533 CONTINUE
              GO TO 1545
1536 CONTINUE
              DO 1540 I=1,NRM
                  DO 1540 J=NM1,NM2
                      JJ=J-NM1+1
1540 R(I,J)=RR(KR,I,JJ)
1545 CONTINUE
C
C   FORM P MATRIX (UI/R1)
C
      NN1=1
      NN2=0
      DO 1562 N=1,KB

```

```

SIMK 197
SIMK 198
SIMK 199
SIMK 200
SIMK 201
SIMK 202
SIMK 203
SIMK 204
SIMK 205
SIMK 206
SIMK 207
SIMK 208
SIMK 209
SIMK 210
SIMK 211
SIMK 212
SIMK 213
SIMK 214
SIMK 215
SIMK 216
SIMK 217
SIMK 218
SIMK 219
SIMK 220
SIMK 221
SIMK 222
SIMK 223
SIMK 224
SIMK 225
SIMK 226
SIMK 227
SIMK 228
SIMK 229
SIMK 230
SIMK 231
SIMK 232
SIMK 233
SIMK 234
SIMK 235
SIMK 236
SIMK 237
SIMK 238
SIMK 239
SIMK 240
SIMK 241
SIMK 242
SIMK 243
SIMK 244
SIMK 245
SIMK 246
SIMK 247
SIMK 248
SIMK 249
SIMK 250
SIMK 251
SIMK 252
SIMK 253
SIMK 254
SIMK 255
SIMK 256
SIMK 257
SIMK 258
SIMK 259
SIMK 260
SIMK 261
SIMK 262

```

Figure 32. Subroutine SIMK Program Listing (Continued)

```

IF(N.GT.1)NN1=NN1+NNU(N-1)
NN2=NN2+NNU(N)
NSY1=NSYS(N)
NSY=NR*(NSY1-1)+NSY2
DO 1540 KP=1,MP
IF(NSP(KP).EQ.NSY)GO TO 1555
1550 CONTINUE
GO TO 1562
1555 CONTINUE
DO 1540 I=NN1,NN2
II=I-NN1+1
DO 1540 J=NM1,NM2
JJ=J-NM1+1
1560 P(I,J)=PP(KP,II,JJ)
1562 CONTINUE
IF(IPRINT.LT.6)GO TO 1565
WRITE(IW,900)KP,NSY,NSP
NMP=NN2-NN1+1
NMP=NMP-NM1+1
CALL MPRS(P,MN,MN,NMP,NMP,0.0,4HPP )
1565 CONTINUE
C
C FORM Q MATRIX (UI/U)
C
KYIN=0
NN1=1
NN2=0
DO 1600 N=1,KR
KYIN=KYIN+NNU(N)
IF(N.GT.1)NN1=NN1+NNU(N-1)
NN2=NN2+NNU(N)
NSY1=NSYS(N)
DO 1570 KQ=1,MQ
IF(NSQ(KQ).EQ.NSY1)GO TO 1575
1570 CONTINUE
GO TO 1600
1575 CONTINUE
DO 1580 I=NN1,NN2
II=I-NN1+1
DO 1580 J=1,NUM
1580 Q(I,J)=QQ(KQ,II,J)
1600 CONTINUE
IF(IPRINT.NE.6)GO TO 1610
CALL MPRS(P,MN,MN,KYIN,KYOUT,T,4HPR )
CALL MPRS(Q,MN,NUM,KYIN,NUM,T,4HQR )
CALL MPRS(R,NRM,MN,NRM,KYOUT,T,4HRR )
CALL MPRS(S,NRM,NUM,NRM,NUM,T,4HRS )
1610 CONTINUE
C
C CALCULATE NR AND NU BY USING Q, R AND S MATRICES
C
DO 1640 J=1,NUM
DO 1620 I=1,KYIN
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 1740 J=1,KYOUT

```

Figure 32. Subroutine SIMK Program Listing (Continued)

# Contrails

	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
	1820 CONTINUE	SIMK 349
	NX=0	SIMK 350
	DO 1840 N=1,KR	SIMK 351
	NX=NX+NNX(N)	SIMK 352
1840	CONTINUE	SIMK 353
	NY=KYIN+KYOUT	SIMK 354
	IF ((IPRINT.NE.3).AND.(IPRINT.LT.5))GO TO 1880	SIMK 355
	WRITE (IW,1860)	SIMK 356
1860	FORMAT (/,/,20X,28H*** INTERCONNECTION DATA ***,/)	SIMK 357
	CALL MPRS (P,MN,MN,KYIN,KYOUT,T,4HP )	SIMK 358
	CALL MPRS (Q,MN,NUM,KYIN,NU,T,4HQ )	SIMK 359
	CALL MPRS (R,NRM,MN,NR,KYOUT,T,4HR )	SIMK 360
	CALL MPRS (S,NRM,NUM,NR,NU,T,4HS )	SIMK 361
		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=N1-NU1	SIMK 369
C		SIMK 370
C	WRITE INTERCONNECTION DATA ON SCRATCH FILE FOR NAME1	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 NAME1,/,)	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		SIMK 392
C	DO 2020 II=1,NR	SIMK 393
C	IF (II.EQ.I)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)



# Contrails

C2020	CONTINUE	SIMK 395
	DO 2040 JJ=NNRKP+NNRK	SIMK 396
	IF(JJ,EQ,J)GO TO 2040	SIMK 397
	IF(R(I,JJ).NE.0.0)GO TO 2100	SIMK 398
2040	CONTINUE	SIMK 399
	NNRKK=0	SIMK 400
	NNRKK=1	SIMK 401
	DO 2070 KK=1,KR	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,JJ).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(I?,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)=MINPU	SIMK 422
	WRITE(JS,1060)CARD	SIMK 423
	IF(IPRINT,EQ,6)WRITE(IW,2000)CARD	SIMK 424
	NNUK=1	SIMK 425
	NNUKP=1	SIMK 426
	DO 2200 K=1,KR	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
	NNUKK=1	SIMK 442
	DO 2170 KK=1,KR	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)

# Contrails

```

RETURN;
C
C COMPUTE SUBSYSTEM STATES XDOT(N)=AN*XN +BN*UN
C
150 CONTINUE
II=0
DO 25 N=1,KR
NNXN=NX(N)
DO 20 I=1,NNXN
II=II+1
XDOTL(II)=0.0
NNUN=NU(N)
DO 201 J=1,NNUN
251 XDOTL(II)=XDOTL(II)+BT(I,J,N)*UI(J,N)
DO 20 J=1,NNXN
250 XDOTL(II)=XDOTL(II)+AT(I,J,N)*X(J,N)
251 CONTINUE
C
C INTERCONNECTION EQUATIONS
C
C INTERNAL OUTPUTS RIN=CN*XN+DN*UN
C
II=0
DO 35 N=1,KR
NNRN=NR(N)
DO 30 I=1,NNRN
II=II+1
YL(II)=0.0
NNXN=NX(N)
DO 301 J=1,NNXN
301 YL(II)=YL(II)+CT(I,J,N)*X(J,N)
NNUN=NU(N)
DO 30 J=1,NNUN
300 YL(II)=YL(II)+DT(I,J,N)*UI(J,N)
350 CONTINUE
C
C INTERNAL INPUTS
C
J=0
DO 22 N=1,KR
NNRN=NR(N)
DO 22 I=1,NNRN
J=J+1
220 RIN(J)=RI(I,N)
DO 24 I=1,KYIN
II=II+1
YL(II)=0.0
DO 23 J=1,KYOUT
230 YL(II)=YL(II)+P(I,J)*RIN(J)
DO 24 J=1,NU
240 YL(II)=YL(II)+Q(I,J)*U(J)
C
C EXTERNAL RESPONSE EQUATIONS
C
II=0
DO 28 I=1,NR
II=II+1
RL(II)=0.0
DO 27 J=1,KYOUT
270 RL(II)=RL(II)+R(I,J)*RIN(J)
DO 28 J=1,NU
280 RL(II)=RL(II)+S(I,J)*U(J)
RETURN;
END
SIMK 461
SIMK 462
SIMK 463
SIMK 464
SIMK 465
SIMK 466
SIMK 467
SIMK 468
SIMK 469
SIMK 470
SIMK 471
SIMK 472
SIMK 473
SIMK 474
SIMK 475
SIMK 476
SIMK 477
SIMK 478
SIMK 479
SIMK 480
SIMK 481
SIMK 482
SIMK 483
SIMK 484
SIMK 485
SIMK 486
SIMK 487
SIMK 488
SIMK 489
SIMK 490
SIMK 491
SIMK 492
SIMK 493
SIMK 494
SIMK 495
SIMK 496
SIMK 497
SIMK 498
SIMK 499
SIMK 500
SIMK 501
SIMK 502
SIMK 503
SIMK 504
SIMK 505
SIMK 506
SIMK 507
SIMK 508
SIMK 509
SIMK 510
SIMK 511
SIMK 512
SIMK 513
SIMK 514
SIMK 515
SIMK 516
SIMK 517
SIMK 518
SIMK 519
SIMK 520
SIMK 521
SIMK 522
SIMK 523
SIMK 524

```

Figure 32. Subroutine SIMK Program Listing (Concluded)

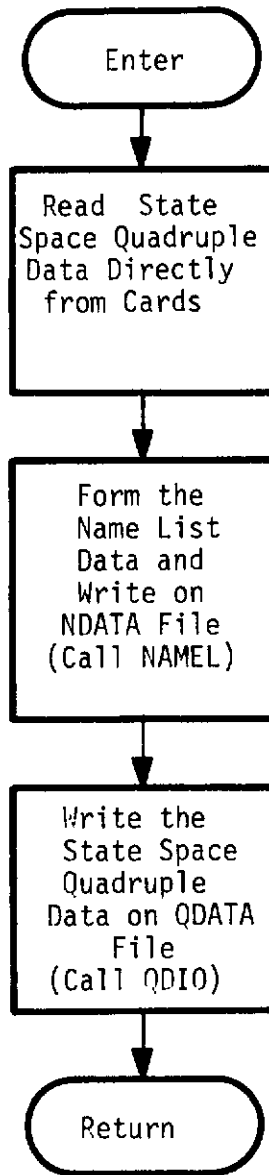


Figure 33. Subroutine QUADK Flow Chart

# Contrails

```

      SUPROUTINE QUADK (A,B,C,D,NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,
      INNI,VNI,DESI,UNITI,DESSS,UNITSS,DESOO,UNITOO,DESII,UNITII,
      2NXX,NXP,NUU,NXM,NRM,NUM,MR,MB)
C
C   PURPOSE - TO READ DIRECTLY THE STATE MODEL
C   ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C   DATE WRITTEN - 1975
C
C   SUBPROGRAMS CALLED
C     INPT
C     QDIO
C     NAMEL
C     ZERO
C
C   ARGUMENTS LIST
C     DESSS          STATE DESCRIPTION ARRAY FOR ALL SUBSYSTEMS
C     UNITSS        STATE UNIT ARRAY FOR ALL SUBSYSTEMS
C     DESOO         OUTPUT DESCRIPTION ARRAY FOR ALL SUBSYSTEMS
C     UNITOO        OUTPUT UNIT ARRAY FOR ALL SUBSYSTEMS
C     DESII         INPUT DESCRIPTION ARRAY FOR ALL SUBSYSTEMS
C     UNITII        INPUT UNIT ARRAY FOR ALL SUBSYSTEMS
C     NXX           NO OF STATE ARRAY FOR ALL SUBSYSTEMS
C     NRP           NO OF OUTPUT ARRAY FOR ALL SUBSYSTEMS
C     NUP           NO OF INPUT ARRAY FOR ALL SUBSYSTEMS
C   OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
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,PHAD(20)
C   DIMENSION A(NXM,NXM),B(NAM,NUM),C(NRM,NXM),D(NPM,NUM)
C   DIMENSION NNS(NXM),VNS(NAM,2),DESS(NXM,10),UNITS(NXM,4)
C   DIMENSION NNO(NPM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
C   DIMENSION INNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
C   DIMENSION DESSS(NXM,10,MB),UNITSS(NAM,4,MB)
C   DIMENSION DESOO(NRM,10,MB),UNITOO(NRM,4,MB)
C   DIMENSION DESII(NUM,10,MB),UNITII(NUM,4,MB)
C   DIMENSION NXX(MR),NRR(MR),NUU(MR)
C   DIMENSION CARD(20)
C   DATA HC,HSAMP,HXDOT,HXJ,HC,4HSAMP,4HXDOT,4HX /
C   DATA H5U,HR5X,HR5U,HEND/4H/U,4HR/X,4HR/U,4HEND /
C   T=0.0 $ NX=0 $ NR=0 $ NU=0
C   NR1=0 $ NR2=0 $ NR3=0 $ NU1=0 $ NU2=0 $ NU3=0
C   NXA=0 $ NRA=0 $ NUA=0
C   CALL ZERO(A,NXM,NXM)
C   CALL ZERO(B,NXM,NUM)
C   CALL ZERO(C,NRM,NXM)
C   CALL ZERO(D,NRM,NUM)
C
C   READ QUADRUPLE DATA FROM CARDS
C
C 45 CONTINUE
C 100 CONTINUE
C   READ(IR,120)CAPD
C 120 FORMAT(2JA4)
C   DECODE(4,140,CARD(1))CC,DUMMY
C 140 FORMAT(A),A3)
C   IF(CC.EQ.HC)GO TO 100
C   IF(CAPD(1).NE.HSAMP)GO TO 200
C   ENCODE(10,160,TW)CARD(4),CARD(5),CARD(6)
C 160 FORMAT(A4,A4,A?)
C   DECODE(10,180,TW)T
C 180 FORMAT(F10.6)
C 200 CONTINUE
```

Figure 34. Subroutine QUADK Program Listing

# Contrails

	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(AI,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
	CALL INPT(C,NRM,NXM)	QUADK 90
	GO TO 100	QUADK 91
260	CONTINUE	QUADK 92
	IF(CARD(1).NE.HRSU)GO TO 280	QUADK 93
C		QUADK 94
C	READ D MATRIX (R/U)	QUADK 95
C		QUADK 96
	DECODE(4,210,CARD(4))NR,DUMMY	QUADK 97
	DECODE(4,230,CARD(5))DUMMY,NU	QUADK 98
	CALL INPT(D,NRM,NUM)	QUADK 99
	GO TO 100	QUADK 100
280	CONTINUE	QUADK 101
	IF(CARD(1).NE.HEND)GO TO 320	QUADK 102
C		QUADK 103
C	READ AND UPDATE NAME LIST DATA	QUADK 104
C		QUADK 105
	NFLAG=0	QUADK 106
	CALL NAMEL(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,MNI,VNI, IDESI,UNITI,DESS,UNITSS,DESO,UNITOD,DESI,UNITII,NXX,NRR,NUU, 2NXM,NRM,NUM,NX,NR,NU,NFLAG,4B,KB,NB)	QUADK 107
C		QUADK 108
C	WRITE QUADRUPLE DATA ON FILE QDATA	QUADK 109
C		QUADK 110
	IQ=0	QUADK 111
	MFLAG=?	QUADK 112
	NXA=NX \$ NRA=NR \$ NUA=NU	QUADK 113
	CALL QDIO(A,B,C,D,DESS,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA, 1NR1,NR2,NR3,NU1,NU2,NU3,T,IO,[PRINT,I#,JQ,HEAD,MARK, 2LOCATE,NULL,INSERT,MFLAG)	QUADK 114
	RETURN	QUADK 115
C		QUADK 116
C	PRINT ERROR MESSEGE	QUADK 117
C		QUADK 118
	320 CONTINUE	QUADK 119
	WRITE(IW,340)	QUADK 120
340	FORMAT(1H1,/,1X,37HDATA CONTROL CARD SPECIFICATION ERROR)	QUADK 121
	STOP 111	QUADK 122
	END	QUADK 123
		QUADK 124
		QUADK 125
		QUADK 126
		QUADK 127

Figure 34. Subroutine QUADK Program Listing (Concluded)

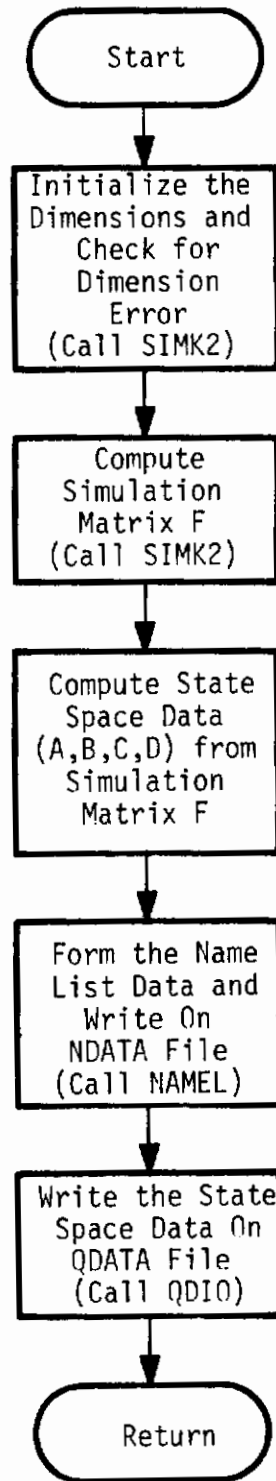


Figure 35. Subroutine STAMK4 Flow Chart

# Contrails

```

SUBROUTINE STAMK4(V,W,F,U,A,B,C,D,NNS,VNS,DESS,UNITS,
INNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI,MAXN,MAXM,
2NXM,NUM,NUM,NYM,MR,MS1,MS2,MS3,MS4,NB)
STAMK4 2
STAMK4 3
STAMK4 4
STAMK4 5
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
C SUBPROGRAMS CALLED
C DERRM
C MPDS
C QDIO
C TDINVR
C DERRMS
C NAMEL
C SIMK2
C
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 B 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
COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),MSYS(9),SHEAD(9,20)
1,PHEAD(20)
DIMENSION V(MAXN),W(MAXM),F(MAXN,MAXM)
DIMENSION U(NUM)
DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
DIMENSION NNS(NXM),VNS(NXM,?),DESS(NXM,10),UNITS(NXM,4)
DIMENSION NNO(NRM),VNO(NRM,?),DESO(NRM,10),UNITO(NRM,4)
DIMENSION NNI(NUM),VNI(NUM,?),DESI(NUM,10),UNITI(NUM,4)
COMMON /SC1/ S1(1)
C
DIMENSION DESS(NXM,10,MB),UNITSS(NXM,4,MB)
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

# Contrails

```

C      DIMENSION DES00(NRM,1),MB),(INIT00(NRM,4,MR)          STAMK465
C      DIMENSION DES11(NUM,10,MB),UNIT11(NUM,4,MR)          STAMK466
C      DIMENSION NXX(MH),MRR(MR),NUU(MB)                    STAMK467
      L1=1 $ L2=L1+NXM*MR*10 $ L3=L2+NXM*MR*4 $ L4=L3+NRM*MB*10 STAMK468
      L5=L4+NRM*MR*4 $ L6=L5+NUM*MB*10 $ L7=L6+NUM*MR*4      STAMK469
      L8=L7+MH $ L9=L8+MR $ L10=L9+MR                        STAMK470
      IF(L1.GT.MS1)                                          STAMK471
1CALL QERRM(L10,MS2,MS3,MS4,MS1,MS2,MS3,MS4,4,0,4HSTAM,4HK4 ,IW) STAMK472
      NR1=0 $ NR2=0 $ NR3=0 $ NU1=0 $ NU2=0 $ NU3=0          STAMK473
      NXA=0 $ NRA=0 $ NUA=0                                  STAMK474
      EPSF=1.0E-30 $ T=0.0 $ NFLAG=0                       STAMK475
C      INITIALIZING CALL TO SUBROUTINE SIMK2                 STAMK476
C      INIT=0                                                STAMK477
C      NX=0 $ NY=0 $ NR=0 $ NU=0                             STAMK478
      N1=1 $ N2=N1+NX $ N3=N2+NY $ N4=N3+NX                  STAMK479
      CALL SIMK2(W(N1),W(N2),W(N3),W(N4),V(N1),V(N2),V(N3),   STAMK480
      INX,NY,NR,NU,INIT,T)                                   STAMK481
C      CHECK FOR DIMENSION ERROR                             STAMK482
C      INIT = 1                                              STAMK483
      M=2*NX+NY+NU                                            STAMK484
      N=NX+NY+NR                                              STAMK485
      IF((NX.GT.NXM).OR.(NR.GT.NRM).OR.(NU.GT.NUM).OR.(NY.GT.NYM)) STAMK486
1CALL QERRMS(NX,NR,NU,NY,NXM,NRM,NUM,NYM,1,0,4HSTAM,4HK1 ,IW) STAMK487
      N1=1 $ N2=N1+NX $ N3=N2+NY $ N4=N3+NX                  STAMK488
      DO 101 J=1,M                                           STAMK489
101  W(J)=0.0                                                 STAMK490
      DO 501 J=1,M                                           STAMK491
      W(J)=1.0                                                STAMK492
      CALL SIMK2(W(N1),W(N2),W(N3),W(N4),V(N1),V(N2),V(N3),   STAMK493
      INX,NY,NR,NU,INIT,T)                                   STAMK494
      W(J)=0.0                                                STAMK495
      DO 501 I=1,N                                           STAMK496
501  F(I,J)=V(I)                                             STAMK497
C      COMPUTE THE SIMULATION MATRIX                          STAMK498
C      NV=NX*NY                                              STAMK499
      IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,V,M,T,4HSIM )     STAMK100
      DO 51 I=1,NV                                            STAMK101
      DO 52 J=1,NV                                            STAMK102
52  F(I,J)=0.0                                               STAMK103
51  F(I,I)=F(I,I)+1.0                                        STAMK104
      CALL TDINVR(ISOL,INDOL,NV,-M,F,MAXN,KDUM,DET)          STAMK105
      IB=NV+1                                                 STAMK106
      IE=NV+NR                                                STAMK107
      JB=IB                                                    STAMK108
      JE=M                                                    STAMK109
      DO 53 I=IB,IE                                           STAMK110
      DO 53 J=JB,JE                                           STAMK111
      DO 53 K=1,NV                                            STAMK112
53  F(I,J)=F(I,J)+F(I,K)*F(K,J)                             STAMK113
      DO 53 I=1,IE                                           STAMK114
      DO 53 J=1,JE                                           STAMK115
      IF(ABS(F(I,J)).LE.EPSF) F(I,J) = 0.0                  STAMK116
530  CONTINUE                                               STAMK117
      IF(IPRINT.EQ.6)CALL MPRS(F,MAXN,MAXM,V,M,T,4HSIM1)     STAMK118
C      FORM A,B,C,D MATRICES                                  STAMK119
C      J1=NV+1                                               STAMK120
      J2=NV+NX                                                STAMK121
      J3=J1+NX                                                STAMK122
      J4=J3+NX                                                STAMK123
      J5=J4+NX                                                STAMK124
      J6=J5+NX                                                STAMK125
      J7=J6+NX                                                STAMK126
      J8=J7+NX                                                STAMK127
      J9=J8+NX                                                STAMK128
      J10=J9+NX                                               STAMK129
      J11=J10+NX                                              STAMK130

```

Figure 36. Subroutine STAMK4 Program Listing (Continued)



	J4=J2+NU	STAMK131
	I1=NV+1	STAMK132
	I2=NV+NR	STAMK133
	DO 6001 I=1,NX	STAMK134
	DO 6001 J=J1,J2	STAMK135
	JJ=J-J1+1	STAMK136
6001	A(I,J)=F(I,J)	STAMK137
	DO 6002 I=1,NX	STAMK138
	DO 6002 J=J3,J4	STAMK139
	JJ=J-J3+1	STAMK140
6002	B(I,J)=F(I,J)	STAMK141
	DO 6003 I=I1,I2	STAMK142
	II=I-I1+1	STAMK143
	DO 6003 J=J1,J2	STAMK144
	JJ=J-J1+1	STAMK145
6003	C(II,JJ)=F(I,J)	STAMK146
	DO 6004 I=I1,I2	STAMK147
	II=I-I1+1	STAMK148
	DO 6004 J=J3,J4	STAMK149
	JJ=J-J3+1	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(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,NNI,VNI,	STAMK156
	IDESI,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=2	STAMK163
	NXA=NX \$ NRA=NR \$ NUA=NU	STAMK164
	CALL QDIO(A,B,C,D,A,NX,NR,NU,NXM,NRM,NUM,NXA,NPA,NUA,	STAMK165
	INR1,NR2,NR3,NU1,NU2,NU3,T,IQ,IPRINT,IN,JG,HEAD,MARK,	STAMK166
	2LOCATE=NULL,INSERT,MFLAG)	STAMK167
	RETURN	STAMK168
	END	STAMK169

Figure 36. Subroutine STAMK4 Program Listing (Concluded)

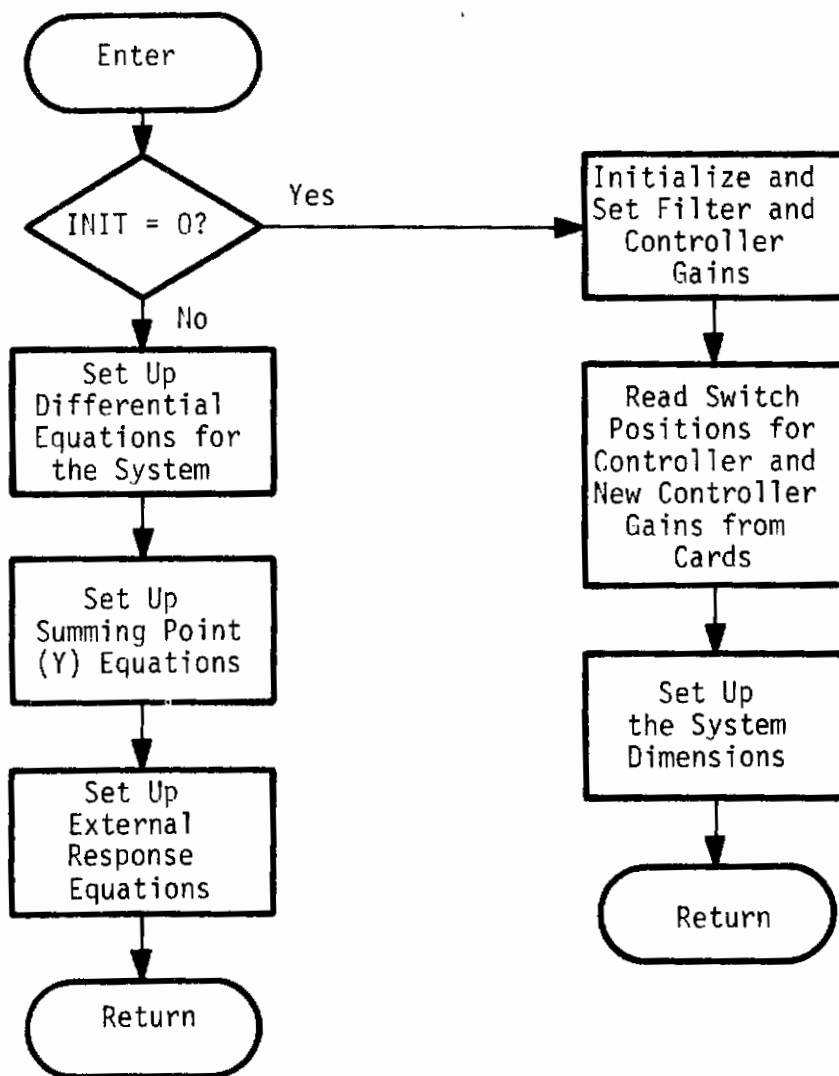


Figure 37. Subroutine SIMK2 Flow Chart

# Contrails

```

SUBROUTINE SIMK2(XDOT,Y,X,U,XDOTL,YL,RL,NX,NY,NR,NU,INIT,T)      SIMK2  2
                                                                    SIMK2  3
C   PURPOSE - TO IMPLEMENT SIMULATION EQUATIONS FOR CSA CONTROLLER  SIMK2  4
C   ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC          SIMK2  5
C   DATE WRITTEN - 1975                                           SIMK2  6
C                                                                    SIMK2  7
C   ARGUMENTS LIST                                               SIMK2  8
C   XDOT      ARRAY FOR STATE DERIVATIVES                       SIMK2  9
C   Y         ARRAY FOR Y EQUATIONS                             SIMK2 10
C   X         ARRAY FOR STATES                                 SIMK2 11
C   U         ARRAY FOR EXTERNAL INPUTS                       SIMK2 12
C   XDOTL     OUTPUT    ARRAY FOR DERIVATIVE OF STATE         SIMK2 13
C   YL        OUTPUT    ARRAY FOR Y EQUATION VARIABLES        SIMK2 14
C   RL        OUTPUT    ARRAY FOR EXTERNAL RESPONSE VARIABLES  SIMK2 15
C   NX        OUTPUT    NUMBER OF STATES                       SIMK2 16
C   NY        OUTPUT    NUMBER OF Y EQUATIONS                  SIMK2 17
C   NR        OUTPUT    NUMBER OF OUTPUTS                      SIMK2 18
C   NU        OUTPUT    NUMBER OF INPUTS                       SIMK2 19
C   INIT      INPUT     INITIAL MODE FLAG                      SIMK2 20
C   T         OUTPUT    SAMPLE TIME                            SIMK2 21
C                                                                    SIMK2 22
C   DIMENSION XDOT(NX),Y(NY),X(NX),U(NU),XDOTL(NX),YL(NY),RL(NR)  SIMK2 23
C   COMMON /INOUT/ IR,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS  SIMK2 24
C   DIMENSION CARD(20)                                           SIMK2 25
C   REAL KM1,KM2,KAF,KQ,KP,KNF,MLC1,MLC2                         SIMK2 26
C   DATA HENDB,HWITC,HAINB/4HEND ,4HWITC,4HAIN /                SIMK2 27
C   DATA HMLC1,HMLC2,HSASB,HALDC/4HMLC1,4HMLC2,4HSAS ,4HALDC/   SIMK2 28
C   DATA HKM1B,HKM2B,HKAFB,HKQBB/4HKM1 ,4HKM2 ,4HKAF ,4HKQ /    SIMK2 29
C   DATA HKPBB,HKNFB/4HKP ,4HKNF /                               SIMK2 30
C                                                                    SIMK2 31
C   CHECK IF INITIALIZATION MODE                                 SIMK2 32
C                                                                    SIMK2 33
C   IF(INIT.NE.0) GO TO 100                                       SIMK2 34
C                                                                    SIMK2 35
C   SET FILTER GAINS                                             SIMK2 36
C                                                                    SIMK2 37
C   AP=-.1 $ BP=.22361E-03                                       SIMK2 38
C   ANF=-6.0                                                       SIMK2 39
C   AF=-.02                                                         SIMK2 40
C   AM1=-.01                                                        SIMK2 41
C   AM2=-.01                                                        SIMK2 42
C   AHF=-1. $ BHF=-1.                                             SIMK2 43
C   ATF=-4.0 $ BTF=4.0                                            SIMK2 44
C                                                                    SIMK2 45
C   SET CONTROLLER SWITCHES                                       SIMK2 46
C                                                                    SIMK2 47
C   SAS=0.0 $ ALDCS=0.0 $ MLC1=0.0 $ MLC2=0.0                    SIMK2 48
C                                                                    SIMK2 49
C   SET CONTROLLER GAINS                                          SIMK2 50
C                                                                    SIMK2 51
C   KM1=1.0/0.26                                                   SIMK2 52
C   KM2=1.0/0.05591                                               SIMK2 53
C   KAF=36.0*0.26                                                 SIMK2 54
C   KQ=0.5                                                         SIMK2 55
C   KP=0.3068                                                      SIMK2 56
C   KNF=-0.09                                                      SIMK2 57
C                                                                    SIMK2 58
C   READ CONTROLLER SWITCHES ON AND CONTROLLER GAIN VALUES     SIMK2 59
C                                                                    SIMK2 60
C   10 CONTINUE                                                    SIMK2 61
C   READ(IR,20)CARD                                               SIMK2 62
C   20 FORMAT(20A4)                                               SIMK2 63
C   IF(CARD(1).EQ.HENDR)GO TO 80                                  SIMK2 64

```

Figure 38. Subroutine SIMK2 Program Listing

# Contrails

	IF (CARD(4).NE.HWITC)GO TO 40	SIMK2 65
C	READ CONTROLLER SWITCHES ON	SIMK2 66
C		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.HKPBR)READ (IR,60)KP	SIMK2 98
	IF (CARD(1).EQ.HKPBR)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)=AHF*X(6)+RHF*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)

# Contrails

```
YL(3)=KP*X(1)+X(6)+U(2)+KNF*U(7)
YL(4)=KM2*U(5)-U(9)
YL(5)=ALDCS*X(5)+X(1)+SAS*KQ*U(8)
C
C  RESPONSE EQUATIONS
C
RL(1)=U(1)
RL(2)=Y(5)
RL(3)=X(1)
C
C  RETURN
C
C  RETURN
END
```

SIMK2131  
SIMK2132  
SIMK2133  
SIMK2134  
SIMK2135  
SIMK2136  
SIMK2137  
SIMK2138  
SIMK2139  
SIMK2140  
SIMK2141  
SIMK2142  
SIMK2143  
SIMK2144

Figure 38. Subroutine SIMK2 Program Listing (Concluded)

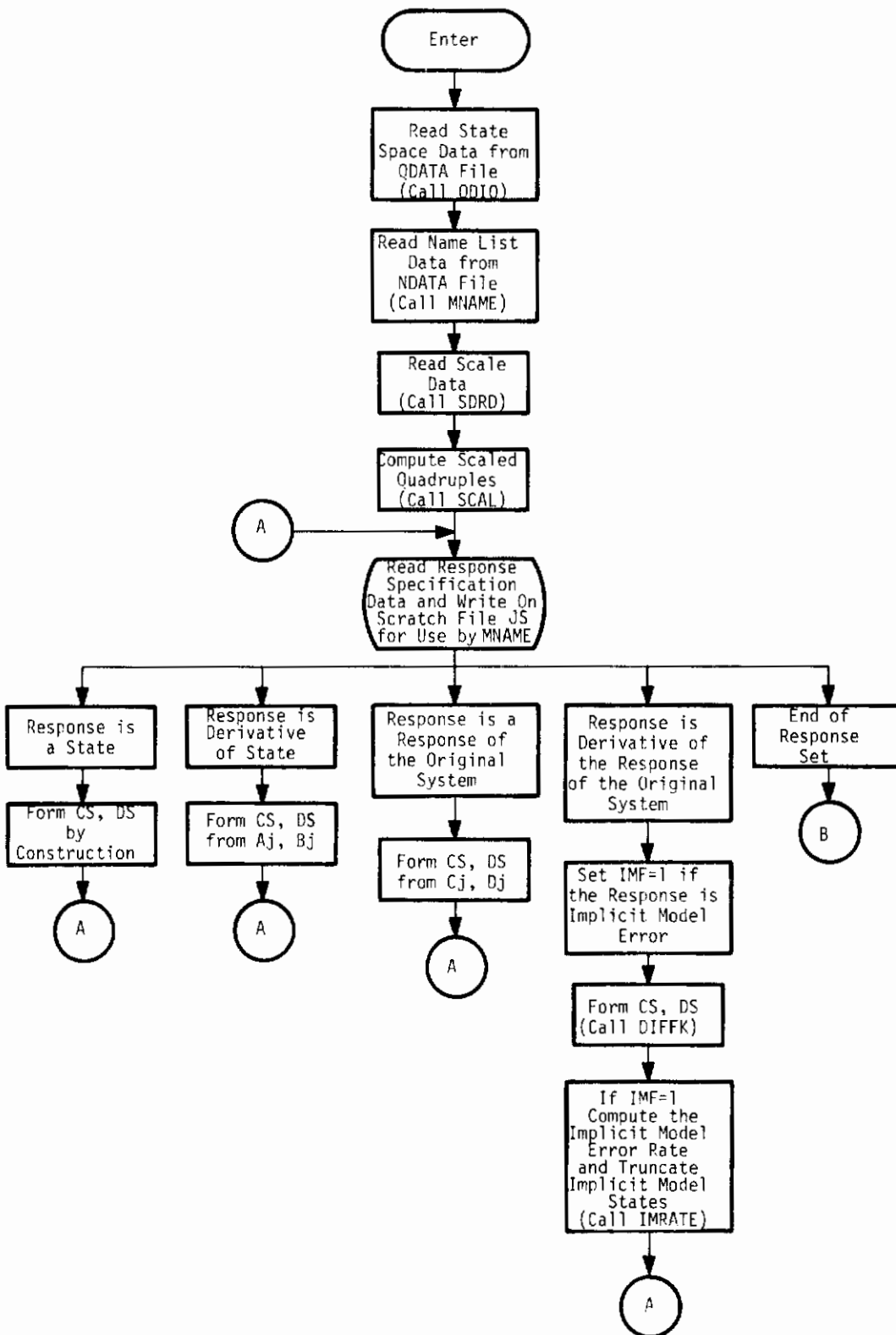


Figure 39. Subroutine CONDK Flow Chart

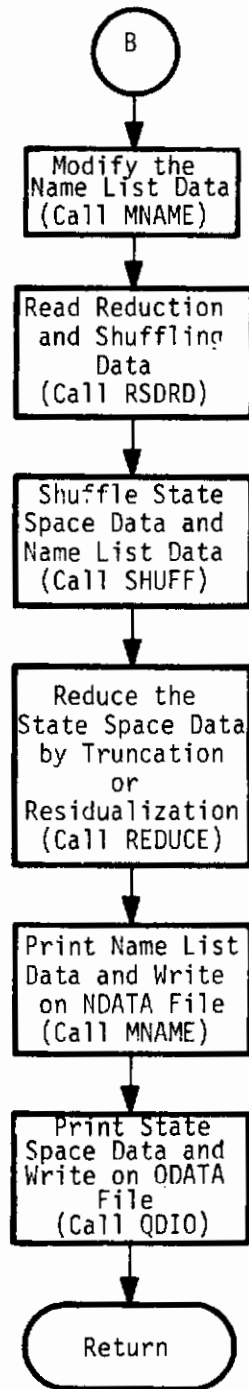


Figure 39. Subroutine CONDK Flow Chart (Concluded)

# Contrails

```

SUBROUTINE CONDK(A,B,C,D,CM,DM,NNS,VNS,DESS,UNITS,NNO,VNO,DESO,
1UNITO,NNI,VNI,DESI,UNITI,NNNS,VNNS,DESNS,UNITNS,NNNO,VNNO,DESN0,
2UNITNO,NNNI,VNNI,DESN1,UNITNI,DUMMY1,DUMMY2,DUMMY3,ES,ER,NSHUF5,
3NSHUF0,NSHUF1,CS,DS,CW,DW,IPS,Q,NXM,NRM,NUM,NDM11,NDM12,NDM21,
4NDM22)
CONDK 2
C
C
C PURPOSE - TO CONDITION THE STATE SPACE QUADRUPLE DATA
CONDK 3
C (RESPONSE SPECIFICATIONS, TRUNCATION AND RESIDUALIZATION
CONDK 4
C AND SHUFFLING)
CONDK 5
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
CONDK 6
C DATE WRITTEN - 1975
CONDK 7
C NOTE - PREVIOUS NAME OF THE SUBROUTINE IS RESPK
CONDK 8
C
CONDK 9
C SUBPROGRAMS CALLED
CONDK 10
C QDIO
CONDK 11
C DERUG
CONDK 12
C MNAME
CONDK 13
C ERPM
CONDK 14
C SDPD
CONDK 15
C SCAL
CONDK 16
C DIFFK
CONDK 17
C IMPATE
CONDK 18
C RSDRD
CONDK 19
C SHUFF
CONDK 20
C REDUCE
CONDK 21
C HPR
CONDK 22
C
CONDK 23
C ARGUMENTS LIST
CONDK 24
C A IN/OUT STATE TRANSITION MATRIX
CONDK 25
C B IN/OUT CONTROL INPUT MATRIX
CONDK 26
C C IN/OUT STATE OUTPUT MATRIX
CONDK 27
C D IN/OUT CONTROL OUTPUT MATRIX
CONDK 28
C CM IN/OUT MODIFIED STATE OUTPUT MATRIX
CONDK 29
C DM IN/OUT MODIFIED CONTROL OUTPUT MATRIX
CONDK 30
C NNS IN/OUT OLD NUMBER ARRAY FOR STATE
CONDK 31
C VNS IN/OUT OLD VARIABLE NAME ARRAY FOR STATE
CONDK 32
C DESS IN/OUT OLD DESCRIPTION ARRAY FOR STATE
CONDK 33
C UNITS IN/OUT OLD UNIT ARRAY FOR STATE
CONDK 34
C NNO IN/OUT OLD NUMBER ARRAY FOR OUTPUT
CONDK 35
C VNO IN/OUT OLD VARIABLE NAME ARRAY FOR OUTPUT
CONDK 36
C DESO IN/OUT OLD DESCRIPTION ARRAY FOR OUTPUT
CONDK 37
C UNITO IN/OUT OLD UNIT ARRAY FOR OUTPUT
CONDK 38
C NNI IN/OUT OLD NUMBER ARRAY FOR INPUT
CONDK 39
C VNI IN/OUT OLD VARIABLE NAME ARRAY FOR INPUT
CONDK 40
C DESI IN/OUT OLD DESCRIPTION ARRAY FOR INPUT
CONDK 41
C UNITI IN/OUT OLD UNIT ARRAY FOR INPUT
CONDK 42
C NNNS IN/OUT NEW NUMBER ARRAY FOR STATE
CONDK 43
C VNNS IN/OUT NEW VARIABLE NAME ARRAY FOR STATE
CONDK 44
C DESNS IN/OUT NEW DESCRIPTION ARRAY FOR STATE
CONDK 45
C UNITNS IN/OUT NEW UNIT ARRAY FOR STATE
CONDK 46
C NNNO IN/OUT NEW NUMBER ARRAY FOR OUTPUT
CONDK 47
C VNNO IN/OUT NEW VARIABLE NAME ARRAY FOR OUTPUT
CONDK 48
C DESNO IN/OUT NEW DESCRIPTION ARRAY FOR OUTPUT
CONDK 49
C UNITNO IN/OUT NEW UNIT ARRAY FOR OUTPUT
CONDK 50
C NNNI IN/OUT NEW NUMBER ARRAY FOR INPUT
CONDK 51
C VNNI IN/OUT NEW VARIABLE NAME ARRAY FOR INPUT
CONDK 52
C DESNI IN/OUT NEW DESCRIPTION ARRAY FOR INPUT
CONDK 53
C UNITNI IN/OUT NEW UNIT ARRAY FOR INPUT
CONDK 54
C DUMMY1 SCRATCH ARRAY
CONDK 55
C DUMMY2 SCRATCH ARRAY
CONDK 56
C DUMMY3 SCRATCH ARRAY
CONDK 57
C ES OUTPUT STATE RESIDUALIZATION ERROR MATRIX
CONDK 58
C ER OUTPUT OUTPUT RESIDUALIZATION ERROR MATRIX
CONDK 59
CONDK 60
CONDK 61
CONDK 62
CONDK 63
CONDK 64

```

Figure 40. Subroutine CONDK Program Listing



# Contrails

```

C      NSHUFFS   INPUT      SHUFFLING ARRAY FOR STATE          CONDK 65
C      NSHUFFO   INPUT      SHUFFLING APRAY 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      NUM       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
COMMON /INOUT/ IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS CONDK 82
COMMON /SYS/ SCODE,SDES(5),MSYS,HEAD(20),NSYS(9),SHEAD(9,20) CONDK 83
1,PHEAD(20) CONDK 84
DIMENSION A(NXM,NXM),B(NXM,NUM),C(NRM,NXM),D(NRM,NUM) CONDK 85
DIMENSION CM(NRM,NXM),DM(NRM,NUM) CONDK 86
DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4) CONDK 87
DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4) CONDK 88
DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4) CONDK 89
DIMENSION NNNS(NXM),VNNNS(NXM,2),DESNS(NXM,10),UNITNS(NXM,4) CONDK 90
DIMENSION NNNO(NRM),VNNNO(NRM,2),DESN0(NRM,10),UNITNO(NRM,4) CONDK 91
DIMENSION NNNI(NUM),VNNI(NUM,2),DESN1(NUM,10),UNITNI(NUM,4) CONDK 92
DIMENSION DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22),DUMMY3(NUM) CONDK 93
DIMENSION ES(NXM,NUM),EP(NRM,NUM) CONDK 94
DIMENSION NSHUFFS(NXM),NSHUFFO(NRM),NSHUFFI(NUM) CONDK 95
DIMENSION CS(NRM,NXM),DS(NRM,NUM),CW(NRM,NXM),DW(NRM,NUM) CONDK 96
DIMENSION IRS(NRM),Q(NRM,NRM) CONDK 97
DIMENSION CARD(20) CONDK 98
EQUIVALENCE (NU1,NCL),(NU2,NGT),(NU3,VCD),(NR1,NDR),(NR2,NPR), CONDK 99
1(NR3,NSR) CONDK 100
DATA HENDB,HRBBB,HRDOT,HXDOT/4HEND ,4HR ,4HRDOT,4HXDOT/ CONDK 101
DATA HXRBB,HSCAL/4HX ,4HSCAL/ CONDK 102
DATA HCONS,HSELE/4HCONS,4HSFLE/ CONDK 103
DATA HSIGN,HRFOR,HENSO/4HSIGN,4HRFOR,4HENSO/ CONDK 104
DATA HONTR,HUSTB,HOMMA/4HONTR,4HUST ,4HOMMA/ CONDK 105
C READ QUADRUPLE DATA CONDK 106
C CONDK 107
C CONDK 108
IQ=0 CONDK 109
NFLAG=1 CONDK 110
CALL QDIO(A,B,C,D,Q,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA, CONDK 111
1NR1,NR2,NR3,NU),NU2,NU3,T,IQ,IPRINT,IW,JQ,PHEAD,MARK, CONDK 112
2LOCATE,NULL,INSERT,NFLAG) CONDK 113
IF(IPRINT.EQ.6)CALL DEBUG(1,4HRESP,4HK ,5,0,IW) CONDK 114
C CONDK 115
C READ NAME LIST DATA CONDK 116
C CONDK 117
MFLAG=1 CONDK 118
CALL MNAME(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO, CONDK 119
1NNI,VNI,DESI,UNITI,NNNS,VNNNS,DESNS,UNITNS,NNNO,VNNO,DESN0, CONDK 120
2UNITNO,NNNI,VNNI,DESN1,UNITNI,NX,NR,NU,NXM,NRM,NUM, CONDK 121
3NU1,NU2,NU3,NR1,NR2,NR3,MFLAG) CONDK 122
IF(IPRINT.EQ.6)CALL DEBUG(2,4HRESP,4HK ,5,0,IW) CONDK 123
REWIND JS CONDK 124
READ(IR,140)CARD CONDK 125
IF(CARD(1).EQ.HENDR)GO TO 110 CONDK 126
IF(CARD(1).NE.HSCAL)CALL ERRM(1,4HRESP,4HK ,5,0,IW) CONDK 127
C CONDK 128
C READ SCALE AND NEW UNIT DATA CONDK 129
C CONDK 130

```

Figure 40. Subroutine CONDK Program Listing (Continued)

# Contrails

```

CALL SDRD(DUMMY1,UNITNS,UNITS,DUMMY2,UNITNO,UNITO,DUMMY3,UNITNI,
UNITE,NX,NR,NU,NXM,NRM,NUM,IR,IW,IPRINT)
C
C COMPUTE SCALED QUADRUPLES
C
CALL SCAL(A,B,C,D,DUMMY1,DUMMY2,DUMMY3,NX,NR,NU,NXM,NRM,NUM)
C
C READ RESPONSE SPECIFICATION DATA AND WRITE IT
C ON A SCRATCH FILE JS FOR USE BY SUBROUTINE MNAME
C
110 CONTINUE
NXO=NX $ NRO=NR
IFLAG=0
IRR=0 $ K=0 $ NDR=0 $ NPR=0 $ NSR=0
JU=0 $ NCL=0 $ NGT=0 $ NCD=0
IRESP=0
120 CONTINUE
READ(IR,140)CARD
140 FORMAT(20A4)
WRITE(JS,140)CARD
IF(CARD(1).EQ.HENDR)GO TO 570
IRESP=1
IF((CARD(1).NE.HCONS).AND.(CARD(1).NE.HSELE))
CALL ERRH(2,4HRESP,4HK ,5,0,IW)
IF((CARD(3).NE.HONTR).AND.(CARD(3).NE.HUSTB).AND.(CARD(3).NE.
IHOMMA))GO TO 240
C
C READ INPUT SPECIFICATION AND MODIFY B AND D MATRICES
C
IF(CARD(3).EQ.HONTR)IUU=1
IF(CARD(3).EQ.HUSTR)IUU=2
IF(CARD(3).EQ.HOMMA)IUU=3
160 CONTINUE
READ(IR,140)CARD
WRITE(JS,140)CARD
IF(CARD(1).EQ.HENDR)GO TO 270
JU=JU+1
DECODE(4,340,CARD(2))DI,K,D?
DO 180 I=1,NX
180 DUMMY1(I,JU)=B(I,K)
DO 200 I=1,NR
200 DUMMY2(I,JU)=D(I,K)
GO TO 160
220 CONTINUE
IF(IUU.EQ.1)NCL=JU
IF(IUU.EQ.2)NGT=JU-NCL
IF(IUU.EQ.3)NCD=JU-NCL-NGT
GO TO 120
240 CONTINUE
IF(IFLAG.EQ.1)GO TO 300
NUN=NCL+NGT+NCD
IF(NUN.EQ.0)GO TO 300
NU=NUN
DO 260 I=1,NX
DO 260 J=1,NU
260 B(I,J)=DUMMY1(I,J)
DO 280 I=1,NR
DO 280 J=1,NU
280 D(I,J)=DUMMY2(I,J)
IFLAG=1
C
C READ OUTPUT SPECIFICATION AND COMPUTE C AND D MATRICES
C
300 CONTINUE
IF(CARD(4).EQ.HSIGN)IRR=1
IF(CARD(4).EQ.HRFOR)IRR=2

```

Figure 40. Subroutine CONDK Program Listing (Continued)

# Contrails

```
IF(CARD(3).EQ.HENSO)IPR=3
IM=0 * IMF=0 $ J=0
320 CONTINUE
READ(ITR,140)CARD
WRITE(JS,140)CARD
IF(CARD(1).EQ.HENDR)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.HXRRR)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 380 L=1,NX
380 CS(J,L)=C(K,L)
DO 400 L=1,NU
400 DS(J,L)=D(K,L)
GO TO 320
420 CONTINUE
DO 425 L=1,NX
425 CS(J,L)=0.0
CS(J,K)=1.0
DO 430 L=1,NU
430 DS(J,L)=0.0
GO TO 320
440 CONTINUE
IF(IX.EQ.1)GO TO 500
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 443 L=1,NX
443 CW(IM,L)=C(K,L)
DO 446 L=1,NU
446 DW(IM,L)=D(K,L)
450 CONTINUE
IF(IPRINT.EQ.6)CALL DEBAG(4,4HRESP,4HK ,5,0,IW)
C
C COMPUTE DERIVATIVES OF RESPONSES
C
CALL DIFFK(A,R,C,D,DUMMY1,DJMY2,NX,NR,NU,
INXM,NPM,NUM,NRS,IRS,ID,IW,IPRINT,NDM11,NDM12,NDM21,NDM22)
DO 460 L=1,NX
460 CS(J,L)=C(NR,L)
DO 480 L=1,NU
480 DS(J,L)=D(NR,L)
NR=NR-1
GO TO 320
500 CONTINUE
DO 520 L=1,NX
520 CS(J,L)=A(K,L)
DO 540 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
550 CONTINUE
NR=NJ
IF (IMF.EQ.0) GO TO 560
C
C   COMPUTE IMPLICIT MODEL ERROR RATES AND TRUNCATE THE
C   IMPLICIT MODEL
C
NXR=NX-NXA
IF (IM.NE.NXR) CALL FRRM(4.4HRESP,4HK .5,0,1W)
IF (IPRINT.EQ.6) CALL DEBUG(5.4HRESP,4HK .5,0,1W)
CALL IMRATE(CS,DS,CW,OW,DUMMY1,DUMMY2,NX,NR,NU,
INXM,NPM,NUM,NXA,NRA,NJA,IW,IPRINT,NOM11,NOM12,NOM21,NOM22)
560 CONTINUE
C
C   COMPUTE NEW C AND D MATRICES (OUTPUTS)
C
NXN=NXA
K=0
IF (IR0.EQ.0) NNR=NR
IF (IR0.EQ.1) NDR=NR
IF (IR0.EQ.2) NPR=NR
IF (IR0.EQ.2) K=NDR
IF (IRR.EQ.3) NSR=NR
IF (IR0.EQ.3) K=NDR+NPR
DO 564 I=1,NR
IK=I+K
DO 562 J=1,NX
562 CM(I,K)=CS(I,J)
DO 564 J=1,NU
564 DM(I,K)=DS(I,J)
NX=NX0 $ NR=NRO
GO TO 120
570 CONTINUE
IF (IRFSP.EQ.0) GO TO 595
IF (IPRINT.EQ.6) CALL DEBUG(6.4HRESP,4HK .5,0,1W)
IF (IR0.EQ.0) NR=NNR
IF (IRR.EQ.0) GO TO 575
NX=NXN $ NR=NDR+NPR+NSR
575 CONTINUE
DO 590 I=1,NR
DO 580 J=1,NX
580 C(I,J)=CM(I,J)
DO 590 J=1,NU
590 D(I,J)=DM(I,J)
C
C   MODIFY NAME LIST DATA
C
MFLAG=?
CALL MNAME(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,
INNI,VNI,DESI,UNITI,NNNS,VNNS,DESNS,UNITNS,NNNO,VNNO,DESN0,
ZUNITNO,NNNI,VNNI,DESN1,UNITN1,NX,NR,NJ,NXM,NRM,NUM,
3NU1,NU2,NU3,NR1,NR2,NR3,MFLAG)
IF (IPRINT.EQ.6) CALL DEBUG(7.4HRESP,4HK .5,0,1W)
595 CONTINUE
C
C   READ SHUFFLING AND REDUCTION DATA
C
CALL RSDRD(DUMMY1,NSHUF5,DUMMY2,NSHUF0,DUMMY3,NSHUF1,
INX,NR,NU,NXRN,NXN,NXR,NRN,NRR,NRT,NUN,NXM,NRM,NUM,
2IR,IW,IPRINT,IRSS)
IF (IRSS.EQ.0) GO TO 600
IF (IPRINT.EQ.6) CALL DEBUG(8.4HRESP,4HK .5,0,1W)
IF (IPRINT.LT.6) GO TO 598
WRITE (IW,596) NX,NR,NU,NXM,NRM,NUM,NDM
WRITE (IW,596) NSHUF5,NSHUF0,NSHUF1

```

Figure 40. Subroutine CONDK Program Listing (Continued)

# Contrails

```

596 FORMAT(1X,(20I2,1X))
598 CONTINUE
C
C   SHUFFLE QUADRUPLE DATA AND NAME LIST DATA
C
C   CALL SHUFF(A,B,C,D,NNS,VNS,DESNS,UNITNS,NNO,
1VNO,DESNO,UNITNO,NNI,VNI,DESNI,UNITNI,
2NSHUF5,NSHUF0,NSHUF1,DUMMY1,DUMMY2,NX,NR,NU,NXM,NRM,NUM,
3NDM11,NDM12,NDM21,NDM22)
C
C   TRUNCATE THE SYSTEM VARIABLES AS SPECIFIED
C
C   NX=NXRN
C   NR=NRN
C   IF(NRN.EQ.0)NR=NRT+NRR
C   NU=NUN
C   IF(NXR.LE.0)GO TO 600
C
C   REDUCE THE QUADRUPLE DATA
C
C   IF(IPRINT.EQ.5)CALL HPR(HEAD,IW)
C   CALL REDUCE(A,B,C,D,DUMMY1,DUMMY2,ES,ER,
1NX,NR,NU,NXR,NRR,NRT,NXM,NRM,NUM,T,IW,IPRINT,
2NDM11,NDM12,NDM21,NDM22)
C   IF(IPRINT.EQ.6)CALL DEBUG(9,4HRESP,4HK  ,5.0,IW)
C
C   WRITE NAME LIST DATA
C
C   600 CONTINUE
C   MFLAG=3
C   CALL MNAME(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,
1NNI,VNI,DESI,UNITI,NNNS,VNNS,DESNS,UNITNS,NNNO,VNNO,DESNO,
2UNITNO,NNNI,VNNI,DESNI,UNITNI,NX,NR,NU,NXM,NRM,NUM,
3NU1,NU2,NU3,NR1,NR2,NR3,MFLAG)
C   IF(IPRINT.EQ.6)CALL DEBUG(10,4HRESP,4HK  ,5.0,IW)
C
C   WRITE QUADRUPLE DATA
C
C   IQ=0 $ NFLAG=2
C   NXA=NX $ NRA=NR $ NUA=NU
C   CALL QDIO(A,B,C,D,Q,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA,
1NR1,NR2,NR3,NU1,NU2,NU3,T,IQ,IPRINT,1#,JQ,HEAD,MARK,
2LOCATF,NULL,INSERT,NFLAG)
C   IF(IPRINT.EQ.6)CALL DEBUG(11,4HRESP,4HK  ,5.0,IW)
C   RETURN
C   END

```

Figure 40. Subroutine CONDK Program Listing (Concluded)

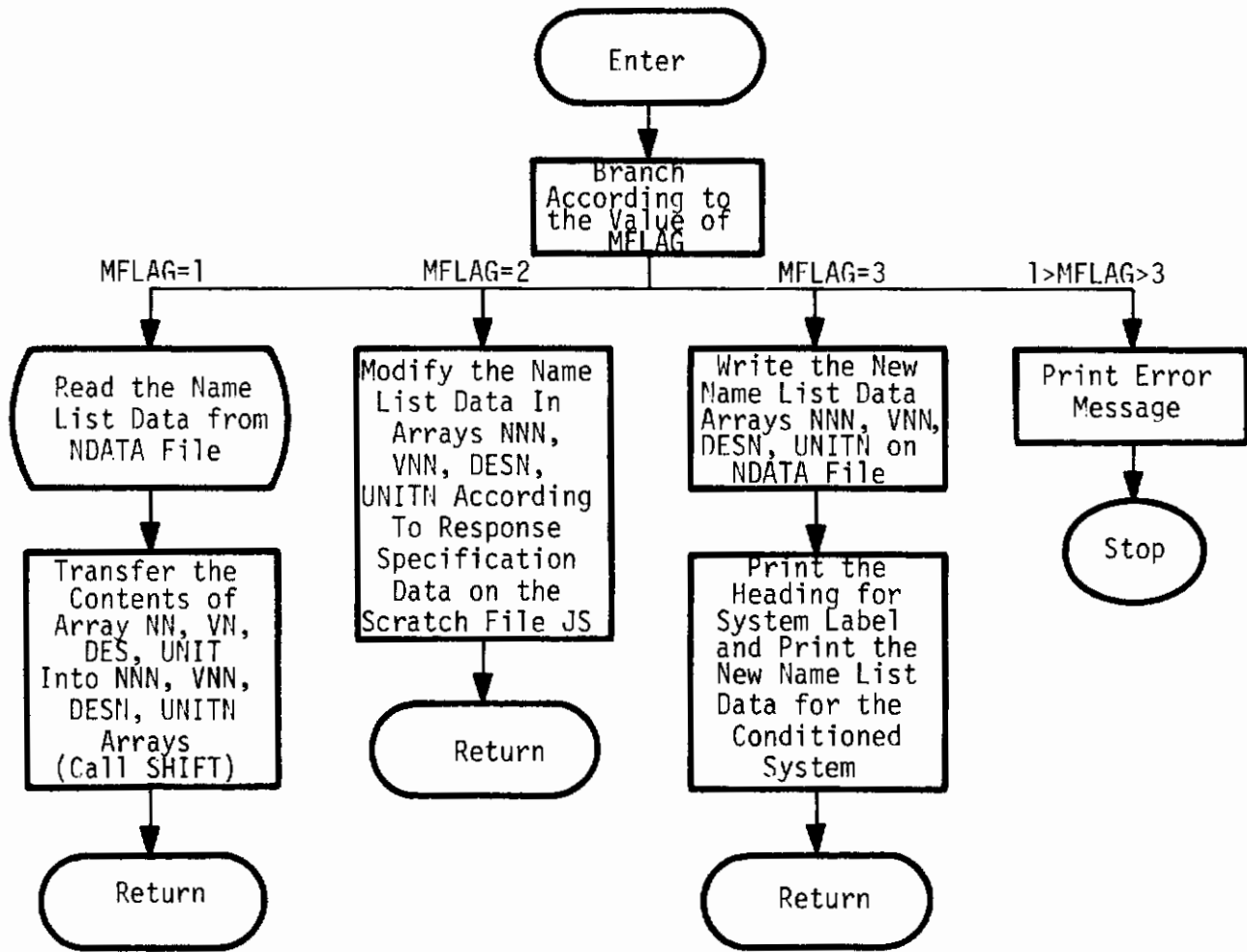


Figure 41. Subroutine MNAME Flow Chart

# Contrails

```

SUBROUTINE MNAME(NNS,VNS,DESS,UNITS,NNO,VNO,DESO,UNITO,
1NNI,VNI,DESI,UNITI,NNNS,VNNS,DESNS,UNITNS,NMNO,VNNO,DESN,
2UNITNO,NNNI,VNNI,DESN,UNITNI,NXN,NRN,NUN,NXM,NRM,NUM,
3NCL,NGT,NCD,NDR,NPR,NSR,MFLAG)
MNAME 2
MNAME 3
MNAME 4
MNAME 5
MNAME 6
C
C PURPOSE - TO READ, MANIPULATE AND PRINT NAME LIST TABLE
MNAME 7
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
MNAME 8
C DATE WRITTEN - 1975
MNAME 9
C
C SUBPROGRAMS CALLED
MNAME 10
C FILE
MNAME 11
C ERRM
MNAME 12
C SHIFT
MNAME 13
C DEBUG
MNAME 14
C HPP
MNAME 15
C
C ARGUMENTS LIST
MNAME 16
C NXN OUTPUT NUMBER OF STATES
MNAME 17
C NRN OUTPUT NUMBER OF OUTPUTS
MNAME 18
C NUN OUTPUT NUMBER OF INPUTS
MNAME 19
C NCI OUTPUT NO OF CONTROL INPUTS
MNAME 20
C NGT OUTPUT NO OF GUST INPUTS
MNAME 21
C NCD OUTPUT NO OF COMMAND INPUTS
MNAME 22
C NDR OUTPUT NO OF DESIGN OUTPUTS
MNAME 23
C NPR OUTPUT NO OF PERFORMANCE OUTPUTS
MNAME 24
C NSR OUTPUT NO OF SENSOR OUTPUTS
MNAME 25
C MFLAG INPUT CONTROLS ENTRY POINT IN THE SUBROUTINE
MNAME 26
C OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
MNAME 27
C
C COMMON /SYS/ SCODE,SDES(5),MSYS,HFAD(20),NSYS(9),SHEAD(9,20)
MNAME 28
1,PHEAD(20)
MNAME 29
C COMMON /INOUT/ IR,JW,IPRINT,INSERT,LOCATE,NULL,MARK(20),JN,JQ,JS
MNAME 30
C DIMENSION NNS(NXM),VNS(NXM,2),DESS(NXM,10),UNITS(NXM,4)
MNAME 31
C DIMENSION NNO(NRM),VNO(NRM,2),DESO(NRM,10),UNITO(NRM,4)
MNAME 32
C DIMENSION NNI(NUM),VNI(NUM,2),DESI(NUM,10),UNITI(NUM,4)
MNAME 33
C DIMENSION NNNS(NXM),VNNS(NXM,2),DESNS(NXM,10),UNITNS(NXM,4)
MNAME 34
C DIMENSION NNNI(NUM),VNNI(NUM,2),DESN(NUM,10),UNITNI(NUM,4)
MNAME 35
C DIMENSION CARD(20)
MNAME 36
C DATA HONTR,HUSTR,HOMMA/4HONTR,4HUST,4HOMMA/
MNAME 37
C DATA HPRBB,HRRBP,HRDES/4H( ,4H ),4H DES/
MNAME 38
C DATA HRBBB,HENDR/4H ,4HEND /
MNAME 39
C DATA HXP,HRP,HUP,HP/2HX(,2HR(,2HU(,4H) /
MNAME 40
C DATA HOSDT,HOF,HSSFC/4HD/DT,4H OF ,4H/SEC/
MNAME 41
C DATA HXBBB,HRBBB,HXDOT,HRDOT/4HX ,4HR ,4HXDOT,4HRDOT/
MNAME 42
C REWIN) JS
MNAME 43
C
C READ THE OLD NAME LIST TABLE OFF THE DATA FILE
MNAME 44
C AND TRANSFER INTO NEW NAME LIST TABLE
MNAME 45
C
C IF(MFLAG.NE.1)GO TO 160
MNAME 46
C IF(IPRINT.EQ.6)CALL DEBUG(1,4HMNAM,4HE ,5,0,1W)
MNAME 47
C IF(IPRINT.EQ.6)WRITE(IW,150)PHEAD
MNAME 48
150 FORMAT(1X,20A4)
MNAME 49
C CALL FILE(JN,LOCATE,PHEAD)
MNAME 50
C READ(IN)NX,NR,NU
MNAME 51
1 (NNS(I),(VNS(I,J),J=1,2),
MNAME 52
2 (DESS(I,J),J=1,10),(UNITS(I,J),J=1,4),I=1,NX),
MNAME 53
3 (NNO(I),(VNO(I,J),J=1,2),
MNAME 54
4 (DESO(I,J),J=1,10),(UNITO(I,J),J=1,4),I=1,NR),
MNAME 55
5 (NNI(I),(VNI(I,J),J=1,2),
MNAME 56
6 (DESI(I,J),J=1,10),(UNITI(I,J),J=1,4),I=1,NU)
MNAME 57
C CALL SHIFT(NNS,VNS,DESS,UNITS,NNNS,VNNS,DESNS,UNITNS,NX,NXM)
MNAME 58
MNAME 59
MNAME 60
MNAME 61
MNAME 62
MNAME 63
MNAME 64
```

Figure 42. Subroutine MNAME Program Listing

# Contrails

```
      IIW,IP,INT)
      CALL SHIFT(MNO,VNO,DFSO,UNITO,NNNO,VNNO,DESN,UNITNO,VR,NRM,
      IIW,IP,INT)
      CALL SHIFT(NNI,VNI,DESI,UNITI,NNNI,VNVI,DESVI,UNITVI,VU,NUM,
      IIW,IP,INT)
      IF(IP,INT.EQ.6)CALL DEBUG(2,4HMNAM,4HE .5,0,1W)
      RETURN
C
C   MODIFY THE NAME LIST TABLE FOR INPUT VARIABLES
C   IF THE INPUT SPACE IS EXPANDED
C
160 CONTINUE
   IF(MFLAG.NE.2)GO TO 760
   IF(IP,INT.EQ.6)CALL DEBUG(3,4HMNAM,4HE .5,0,1W)
   IF(NUM.LT.NU)CALL FRRM(1,4HMNAM,4HE .6,0,1W)
   IF(NUM.EQ.NU)GO TO 280
   NUS=NU
180 CONTINUE
   DO 260 I=1,NUS
     II=NU-I
     NNNI(I)=II
     ENCODE(4,200,VNNI(II,1))HUP,II
200  FORMAT(A2,I2)
     VNNI(I,2)=HP
     DO 220 J=1,4
220  UNITNI(II,J)=UNITI(I,J)
     DO 240 J=1,10
240  DESNI(II,J)=DESI(I,J)
260  CONTINUE
     NU=NU+NUS
     IF(NUM.LT.NU)CALL FRRM(2,4HMNAM,4HE .6,0,1W)
     IF(NUM.EQ.NU)GO TO 280
     GO TO 180
C
C   MODIFY THE NAME LIST FOR OUTPUT VARIABLES AND INPUT VARIABLES
C
280 CONTINUE
   IF(IP,INT.EQ.6)CALL DEBUG(4,4HMNAM,4HE .5,0,1W)
   IFLAG=0
   J=0 $ IRR=0 $ JU=0 $ JUU=0
340 CONTINUE
   READ(JS,480)CARD
   IF(CARD(1).EQ.HENDR)RETURN
   IF((CARD(3).NE.HONTR).AND.(CARD(3).NE.HUSTB).AND.(CARD(3).NE.
1HOMMA))GO TO 460
C
C   OBTAIN NAME LIST DATA FOR SPECIFIED INPUTS
C
360 CONTINUE
   READ(JS,480)CARD
   IF(CARD(1).EQ.HENDR)GO TO 340
   JU=JU+1
   DECODE(4,500,CARD(2))DI,K,D2
   NNNI(JU)=JU
   NNI(JU)=JU
   ENCODE(4,200,VNNI(JU,1))HUP,JU
   VNNI(JU,2)=HP
   VNI(JU,1)=VNNI(JU,1)
   VNI(JU,2)=VNNI(JU,2)
   DO 380 L=1,10
380  DESNI(JU,L)=DESI(K,L)
   DO 400 L=1,4
400  UNITNI(JU,L)=UNITI(K,L)
   GO TO 360
C
C   OBTAIN NAME LIST DATA FOR SPECIFIED OUTPUTS
C
```

Figure 42. Subroutine MNAME Program Listing (Continued)



# Contrails

```
C
460 CONTINUE
    READ (JS,480) CARD
480 FORMAT(20A4)
    IF (CARD(1).EQ.HENDR) GO TO 340
    J=J+1
    DECODE (4,5)0,CARD(2)D)K,02
500 FORMAT(A1,I2,A1)
    ID=0 * IX=0
    IF (CARD(1).EQ.HRRRR) GO TO 520
    ID=0 * IX=1
    IF (CARD(1).EQ.HXRRR) GO TO 520
    ID=1 * IX=0
    IF (CARD(1).EQ.HRDOT) GO TO 520
    ID=1 * IX=1
    IF (CARD(1).EQ.HXDUT) GO TO 520
    CALL FRRM(3,4HMNAM,4HE .6,0,IW)
520 CONTINUE
    NNNO(I)=J
    NNO(J)=J
    ENCODE (4,200,VNNO(I,1))HRP,J
    VNNO(I,2)=HP
    VNO(J,1)=VNNO(J,1)
    VNO(J,2)=VNNO(J,2)
    IF (ID.EQ.1) GO TO 640
    IF (IX.EQ.1) GO TO 520
    DO 540 L=1,10
540 DESNO(J,L)=DESO(K,L)
    DO 560 L=1,4
560 UNITNO(J,L)=UNITO(K,L)
    GO TO 460
580 CONTINUE
    DO 600 L=1,10
600 DESNO(J,L)=DESS(K,L)
    DO 620 L=1,4
620 UNITNO(J,L)=UNITS(K,L)
    GO TO 460
C
C   FORM NAME LIST DATA FOR DERIVATIVES OF SPECIFIED STATES
C   OR OUTPUTS
C
640 CONTINUE
    DESNO(J,1)=HDSOT
    DESNO(J,2)=HOF
    DESNO(J,3)=HPRRR
    DESNO(J,10)=HRRRP
    UNITNO(J,4)=HSSEC
    IF (IX.EQ.1) GO TO 740
    DO 660 L=1,6
    LL=3+L
660 DESNO(J,LL)=DESO(K,L)
    DO 680 L=1,3
680 UNITNO(J,L)=UNITO(K,L)
    GO TO 460
700 CONTINUE
    DO 720 L=1,6
    LL=3+L
720 DESNO(J,LL)=DESS(K,L)
    DO 740 L=1,3
740 UNITNO(J,L)=UNITS(K,L)
    GO TO 460
760 CONTINUE
    IF (IPRINT.EQ.6) CALL DERUG(5,4HMNAM,4HE .5,0,IW)
C
C   WRITE THE NEW NAME LIST TABLE ON THE DATA FILE
C
```

MNAME131  
MNAME132  
MNAME133  
MNAME134  
MNAME135  
MNAME136  
MNAME137  
MNAME138  
MNAME139  
MNAME140  
MNAME141  
MNAME142  
MNAME143  
MNAME144  
MNAME145  
MNAME146  
MNAME147  
MNAME148  
MNAME149  
MNAME150  
MNAME151  
MNAME152  
MNAME153  
MNAME154  
MNAME155  
MNAME156  
MNAME157  
MNAME158  
MNAME159  
MNAME160  
MNAME161  
MNAME162  
MNAME163  
MNAME164  
MNAME165  
MNAME166  
MNAME167  
MNAME168  
MNAME169  
MNAME170  
MNAME171  
MNAME172  
MNAME173  
MNAME174  
MNAME175  
MNAME176  
MNAME177  
MNAME178  
MNAME179  
MNAME180  
MNAME181  
MNAME182  
MNAME183  
MNAME184  
MNAME185  
MNAME186  
MNAME187  
MNAME188  
MNAME189  
MNAME190  
MNAME191  
MNAME192  
MNAME193  
MNAME194  
MNAME195  
MNAME196

Figure 42. Subroutine MNAME Program Listing (Continued)

# Contrails

```
IF (MFLAG.EQ.3) CALL ERRM(4,4)MNAME,4HE .6,0,IW)
CALL FILE(JN,INSEPT,HEAD)
WRITE(IW)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,NXN),
3 (NNNO(I),(VNNO(I,J),J=1,2),
4 (DESN(I,J),J=1,10),(UNITNO(I,J),J=1,4),I=1,NRN),
5 (NNNI(I),(VNMI(I,J),J=1,2),
6 (DESN(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 (IPPRINT.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,6H 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,NXN)
800 FORMAT(1X,I2,6X,2A4,4X,10A4,4X,4A4)
IF (NDR.EQ.0) GO TO 040
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),(DESN(I,J),J=1,10),
1 (UNITNO(I,J),J=1,4),I=1,NDR)
820 CONTINUE
NDPR=NDR+NPR
IF (NPR.EQ.0) GO TO 840
C
C PRINT NAME LIST DATA FOR PERFORMANCE OUTPUTS
C
WRITE(IW,830)
830 FORMAT(//,1X,18HPERFORMANCE OUTPUT,/)
NDPR1=NDR+1
WRITE(IW,800) (NNNO(I),(VNNO(I,J),J=1,2),(DESN(I,J),J=1,10),
1 (UNITNO(I,J),J=1,4),I=NDPR1,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,/)
NDPR1=NDR+1
WRITE(IW,800) (NNNO(I),(VNNO(I,J),J=1,2),(DESN(I,J),J=1,10),
1 (UNITNO(I,J),J=1,4),I=NDPR1,NRN)
860 CONTINUE
IF (NCI.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)

# Contrails

```
      870 FORMAT(/,1X,13#CONTROL INPUT,/)
      WRITE(IW,886) (NINI(I), (VNNI(I,J),J=1,2), (DESNI(I,J),J=1,10),
1          (UNITNI(I,J),J=1,4), I=1, NCL)
      880 CONTINUE
      NCG=NCL+NGT
      IF(NGT.EQ.0)GO TO 900
C
C   PRINT NAME LIST DATA FOR GUST INPUTS
C
      WRITE(IW,890)
      890 FORMAT(/,1X,10#GUST INPUT,/)
      NCLP1=NCL+1
      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, NCG)
      900 CONTINUE
      IF(NCG.EQ.0)GO TO 920
C
C   PRINT NAME LIST DATA FOR COMMAND INPUTS
C
      WRITE(IW,910)
      910 FORMAT(/,1X,13#COMMAND INPUT,/)
      NCGP1=NCG+1
      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=NCGP1, NUN)
      920 CONTINUE
      IF(IPRINT.EQ.6)CALL DFRUG(6,4#MNAM,4HE .5,0, IW)
      RETURN
      940 CONTINUE
C
C   PRINT NAME LIST DATA FOR OUTPUTS
C
      WRITE(IW,950)
      950 FORMAT(/,1X,6#OUTPUT,/)
      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, VRN)
C
C   PRINT NAME LIST DATA FOR INPUTS
C
      WRITE(IW,960)
      960 FORMAT(/,1X,5#INPUT,/)
      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 DFRUG(7,4#MNAM,4HE .5,0, IW)
      RETURN
      END
```

Figure 42. Subroutine MNAME Program Listing (Concluded)

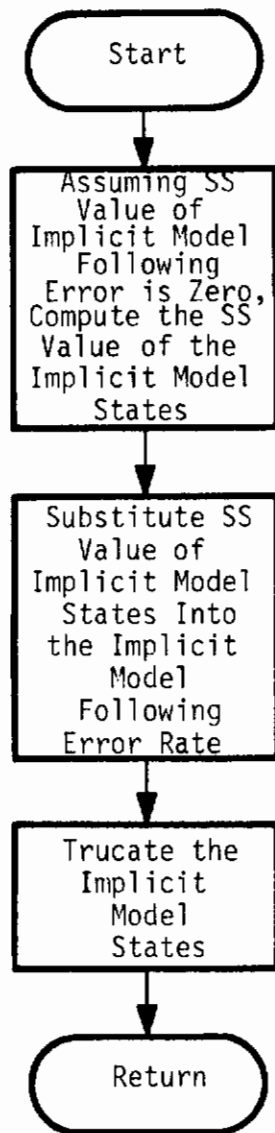


Figure 43. Subroutine IMRATE Flow Chart

# Contrails

```

SUBROUTINE IMRATE(CM,DM,CW,DW,DUMMY1,DUMMY2,NX,NR,NU,
INXM,NRM,NUM,NXA,NRA,NUA,IW,IPRINT,NDM11,NDM12,NDM21,NDM22)
C
C
C      PURPOSE - TO OBTAIN IMPLICIT MODEL ERROR RATES
C      AND TRUNCATE THE IMPLICIT MODEL
C      ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C      DATE WRITTEN - 1975
C
C      SUBPROGRAMS CALLED
C      MPRS
C      TDINVR
C
C      ARGUMENTS LIST
C      NX      INPUT      NUMBER OF STATES
C      NR      INPUT      NUMBER OF OUTPUTS
C      NU      INPUT      NUMBER OF INPUTS
C      NXA     INPUT      NO OF STATES WITHOUT IMPLICIT MODEL
C      NRA     INPUT      NO OF OUTPUTS WITHOUT IMPLICIT MODEL
C      NUA     INPUT      NO OF INPUTS WITHOUT IMPLICIT MODEL
C      IW      INPUT      FILE NO FOR LINE PRINTER
C      IPRINT  INPUT      PRINT CONTROL FLAG
C      OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
C      DIMENSION CM(NRM,NXM),DM(NRM,NUM),CW(NRM,NXM),DW(NRM,NUM)
C      DIMENSION DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22)
C
C      COMPUTE STEADY STATE VALUE OF IMPLICIT MODEL STATES
C      FOR ZERO MODEL ERROR
C
C      NXR=NX-NXA
C      DO 160 I=1,NXR
C      DO 120 J=1,NXR
C      JJ=NXA+J
C 120 DUMMY1(I,J)=CW(I,JJ)
C      DO 140 J=1,NXA
C      JJ=NXR+J
C 140 DUMMY1(I,JJ)=CW(I,J)
C      DO 160 J=1,NU
C      JJ=NX+J
C 160 DUMMY1(I,JJ)=DW(I,J)
C      NDR=NXR
C      NDC=NXR+NXA+NU
C      IF(IPRINT.LT.6)GO TO 170
C      CALL MPRS(CW,NRM,NXM,NXR,NX,0.0,4HCW )
C      CALL MPRS(DW,NRM,NUM,NXR,NU,0.0,4HDW )
C      CALL MPRS(DUMMY1,NDM11,NDM12,NDR,NDC,0.0,4HDMY1)
C 170 CONTINUE
C      CALL TDINVR(ISOL,IDSOL,NDR,-NDC,DUMMY1,NDM11,DUMMY2,DET)
C      IF((ISOL.EQ.1).AND.(IDSOL.EQ.1))GO TO 240
C      WRITE(IW,100)ISOL,IDSOL
C 100 FORMAT(1H1,/,1X,14HTDINVR FAILURE,6H ISOL=,I2,7H IDSOL=,I2)
C      WRITE(IW,200)
C 200 FORMAT(//,1X,43HSTEADY STATE VALUE OF MODEL STATE CANNOT BE,
11X,23HCOMPUTED FOR ZERO ERROR)
C      WRITE(IW,220)
C 220 FORMAT(//,1X,26HMODEL STATES ARE TRUNCATED)
C
C      COMPUTE IMPLICIT MODEL ERROR RATES
C
C 240 CONTINUE
C      DO 260 I=1,NR
C      DO 260 J=1,NX
C 260 DUMMY2(I,J)=CM(I,J)

```

```

IMRATE 2
IMRATE 3
IMRATE 4
IMRATE 5
IMRATE 6
IMRATE 7
IMRATE 8
IMRATE 9
IMRATE10
IMRATE11
IMRATE12
IMRATE13
IMRATE14
IMRATE15
IMRATE16
IMRATE17
IMRATE18
IMRATE19
IMRATE20
IMRATE21
IMRATE22
IMRATE23
IMRATE24
IMRATE25
IMRATE26
IMRATE27
IMRATE28
IMRATE29
IMRATE30
IMRATE31
IMRATE32
IMRATE33
IMRATE34
IMRATE35
IMRATE36
IMRATE37
IMRATE38
IMRATE39
IMRATE40
IMRATE41
IMRATE42
IMRATE43
IMRATE44
IMRATE45
IMRATE46
IMRATE47
IMRATE48
IMRATE49
IMRATE50
IMRATE51
IMRATE52
IMRATE53
IMRATE54
IMRATE55
IMRATE56
IMRATE57
IMRATE58
IMRATE59
IMRATE60
IMRATE61
IMRATE62
IMRATE63
IMRATE64

```

Figure 44. Subroutine IMRATE Program Listing

# Contrails

```
DO 280 I=1, NR
DO 280 J=1, NXA
JJ=NXA+J
DO 280 K=1, NXR
KK=NXA+K
280 CM(I, J)=CM(I, J)-DUMMY2(I, KK)*DUMMY1(K, JJ)
DO 300 I=1, NR
DO 300 J=1, NU
JJ=NX+J
DO 300 K=1, NXR
KK=NXA+K
300 DM(I, J)=DM(I, J)-DUMMY2(I, KK)*DUMMY1(K, JJ)
NX=NXA
IF (IPRINT.LT.6) GO TO 320
CALL MPRS(CM, NRM, NXM, NR, NXA, 0.0, 4, HCM )
CALL MPRS(DM, NRM, NUM, NR, NU, 0.0, 4, HDM )
CALL MPRS(DUMMY1, NDM11, NDM12, NDR, NDC, 0.0, 4, HDMY1)
320 CONTINUE
RETURN
END
```

IMRATE65  
IMRATE66  
IMRATE67  
IMRATE68  
IMRATE69  
IMRATE70  
IMRATE71  
IMRATE72  
IMRATE73  
IMRATE74  
IMRATE75  
IMRATE76  
IMRATE77  
IMRATE78  
IMRATE79  
IMRATE80  
IMRATE81  
IMRATE82  
IMRATE83  
IMRATE84

Figure 44. Subroutine IMRATE Program Listing (Concluded)

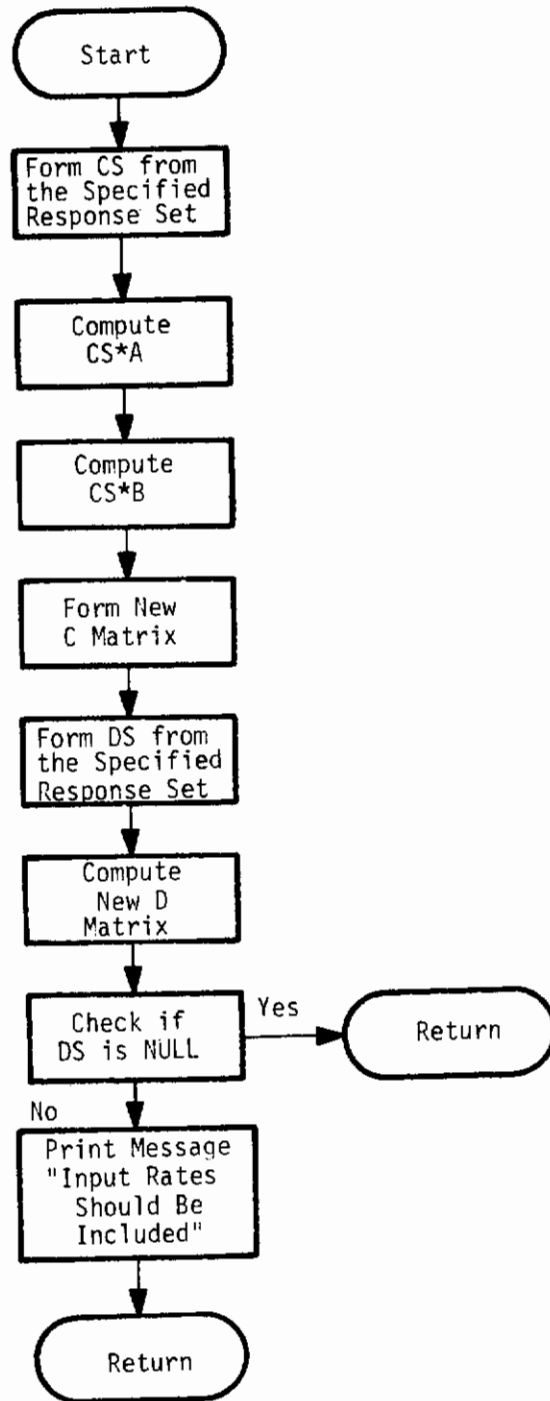


Figure 45. Subroutine DIFFK Flow Chart

# Contrails

```

SUBROUTINE DIFFK(A,B,C,D,DUMMY1,DUMMY2,NX,NR,NU,
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 HONEYWELL 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 LIVE PRINTER
C IPOINT PRINT CONTROL FLAG
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 IRS(NRM),DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22)
C NUS=NU
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(II,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(II,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(II,J)=D(II,J)
C
C COMPUTE NEW D MATRIX

```

```

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



# Contrails

```

      DO 80 I=1,NRS                                DIFFK 65
      DO 80 J=1,NU                                  DIFFK 66
      II=NR+I                                       DIFFK 67
80    D(II,J)=DUMMY2(I,J)                          DIFFK 68
C
C    CHECK IF DS MATRIX IS NULL                    DIFFK 69
C
      DO 90 I=1,NRS                                DIFFK 70
      DO 90 J=1,NRS                                DIFFK 71
      IF(DUMMY1(I,J).NE.0.0)GO TO 100              DIFFK 72
90    CONTINUE                                     DIFFK 73
      NR=NR+NRS                                     DIFFK 74
      RETURN                                        DIFFK 75
C
C    PRINT A MESSAGE THAT THE INPUT RATES ARE NECESSARY FOR DIFFK 76
C    CORRECTLY OBTAINING THE DERIVATIVES OF THE RESPONSES DIFFK 77
C
100   CONTINUE                                     DIFFK 78
      WRITE(IW,120)                                 DIFFK 79
120   FORMAT(1H1,///<.1X,85H*** THE INPUT RATES SHOULD BE INCLUDED IN TAKI DIFFK 80
      ING THE DERIVATIVES OF THE RESPONSES ***,//) DIFFK 81
      RETURN                                        DIFFK 82
140   CONTINUE                                     DIFFK 83
C
C    OBTAIN FIRST AND SECOND DERIVATIVES          DIFFK 84
C
      WRITE(IW,160)                                 DIFFK 85
160   FORMAT(///<.1X,55H*** THE SECOND DERIVATIVE OPTION IS NOT IMPLEMENTED DIFFK 86
      1D ***,//)                                   DIFFK 87
      RETURN                                        DIFFK 88
      END                                          DIFFK 89
                                                    DIFFK 90
                                                    DIFFK 91
                                                    DIFFK 92
                                                    DIFFK 93
                                                    DIFFK 94
                                                    DIFFK 95
```

Figure 46. Subroutine DIFFK Program Listing (Concluded)

# Contrails

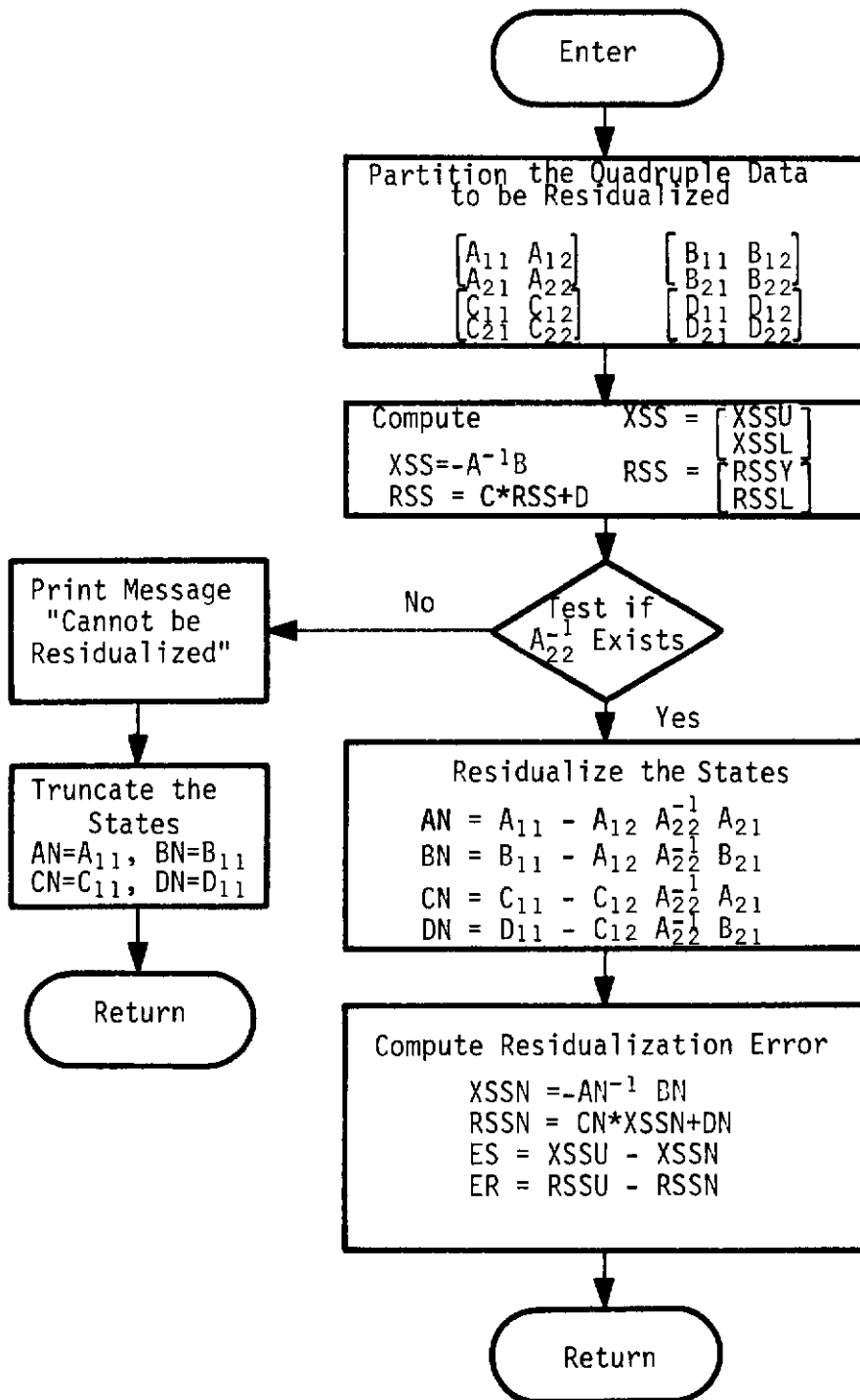


Figure 47. Subroutine REDUCE Flow Chart

# Contrails

```

SUBROUTINE REDUCE(A,B,C,D,DUMMY1,DUMMY2,ES,ER,
1NX,NR,NU,NXR,NPR,NRT,NXM,NRM,NUM,T,IW,IPRINT,
2NDM11,NDM12,NDM21,NDM22)
C
C PURPOSE - TO REDUCE THE ORDER OF STATE SPACE DATA
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C DATE WRITTEN - 1975
C
C SUBPROGRAMS CALLED
C   TDINVR
C   MPOS
C
C ARGUMENT LIST
C   NX      INPUT      NUMBER OF STATES
C   NR      INPUT      NUMBER OF OUTPUTS
C   NU      INPUT      NUMBER OF INPUTS
C   NXR     INPUT      NO OF STATES TO BE RESIDUALIZED
C   NR2     INPUT      NO OF OUTPUTS TO BE RESIDUALIZED
C   NRT     INPUT      NO OF OUTPUTS TO BE TRUNCATED
C   T       INPUT      SAMPLE TIME
C   IW      INPUT      FILE NO FOR LINE PRINTER
C   IPRINT  INPUT      PRINT CONTROL FLAG
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 DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22)
C DIMENSION ES(NXM,NUM),FR(NRM,NUM)
C NXN=NX-NXR
C NRN=NRT-NRR
C NUM=NU
C
C COMPUTE SS VALUE OF STATE AND OUTPUT FOR ORIGINAL SYSTEM
C
C   IF(IPRINT.LT.5)GO TO 260
C   DO 140 I=1,NX
C   DO 140 J=1,NX
140  DUMMY1(I,J)=A(I,J)
C   DO 160 I=1,NX
C   DO 160 J=1,NU
C   JJ=NX+J
160  DUMMY1(I,JJ)=R(I,J)
C   NDR=NX
C   NDC=NX+NU
C   IF(IPRINT.GE.6)CALL MPRS(DUMMY1,NDM11,NDM12,NDR,NDC,T,4NDM1)
C   CALL TDINVR(ISOL,IDSOL,NDR,-NDC,DUMMY1,NDM11,DUMMY2,DET)
C   IF((ISOL.EQ.1).AND.(IDSOL.EQ.1))GO TO 200
C   WRITE(IW,180)ISOL,IDSOL
180  FORMAT(/,1X,14HTDINVR FAILURE,6H ISOL=,I2,7H IDSOL=,I2)
C   WRITE(IW,190)
190  FORMAT(/,1X,40HRESIDUALIZATION ERROR CANNOT BE COMPUTED,
1/1X,30HSINCE SS VALUES DOES NOT EXIST)
C   JPRINT=2
C   GO TO 260
200  CONTINUE
C   JPRINT=IPRINT
C   DO 220 I=1,NR
C   DO 220 J=1,NU
C   JJ=NX+J
220  ES(I,J)=-DUMMY1(I,JJ)
C   DO 240 I=1,NR
C   DO 240 J=1,NU
C   ER(I,J)=D(I,J)
C   DO 240 K=1,NX

```

Figure 48. Subroutine REDUCE Program Listing

# Contrails

	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,4HFS )	REDUCE66
	IF(IPRINT.GE.6)CALL MPRS(ER,NPM,NUM,NR,NU,T,4HER )	REDUCE67
	260 CONTINUE	REDUCE68
C	QUADRUPLE DATA IS REDUCED BY RESIDUALIZATION	REDUCE69
C		REDUCE70
C	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, J)	REDUCE75
	DO 300 I=1,NXR	REDUCE76
	II=NXN+I	REDUCE77
	DO 300 J=1,NXN	REDUCE78
	JJ=NXR+J	REDUCE79
	300 DUMMY1(I, JJ)=A(II, J)	REDUCE80
	DO 320 I=1,NXR	REDUCE81
	II=NXN+I	REDUCE82
	DO 320 J=1,NXN	REDUCE83
	JJ=NXR+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
		REDUCE91
C	RESIDUALIZATION IS NOT POSSIBLE AND SO	REDUCE92
C	QUADRUPLE DATA IS REDUCED BY TRUNCATION	REDUCE93
C		REDUCE94
C	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=NUN	REDUC103
	RETURN	REDUC104
C		REDUC105
C	COMPUTE RESIDUALIZED QUADRUPLES	REDUC106
C		REDUC107
C	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,NXN	REDUC119
	JJ=NXR+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,NRR	REDUC127
	DO 460 J=1,NXN	REDUC128
	JJ=NXR+J	REDUC129
		REDUC130

Figure 48. Subroutine REDUCE Program Listing (Continued)

# Contrails

```
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
C 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(IPRINT.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)+C(I,K)*DUMMY1(K,JJ) REDUC170
WRITE(IW,580) REDUC171
580 FORMAT(/,1X,40HRESIDUALIZATION ERROR MATRICES ES AND ER,/) REDUC172
CALL MPRS(ES,NXN,NUM,NXN+NUN,T,4HES ) REDUC173
CALL MPRS(ER,NRN,NUM,NRN+NUN,T,4HER ) REDUC174
600 CONTINUE REDUC175
NX=NXN                                         REDUC176
NR=NRN                                         REDUC177
NU=NUM                                         REDUC178
RETURN                                         REDUC179
620 CONTINUE REDUC180
STOP 0030 REDUC181
END                                             REDUC182
```

Figure 48. Subroutine REDUCE Program Listing (Concluded)

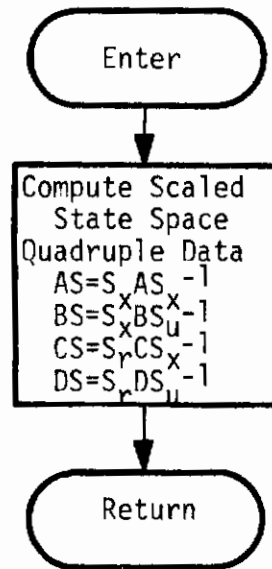


Figure 49. Subroutine SCAL Flow Chart

# Contracts

```

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

Figure 50. Subroutine SCAL Program Listing

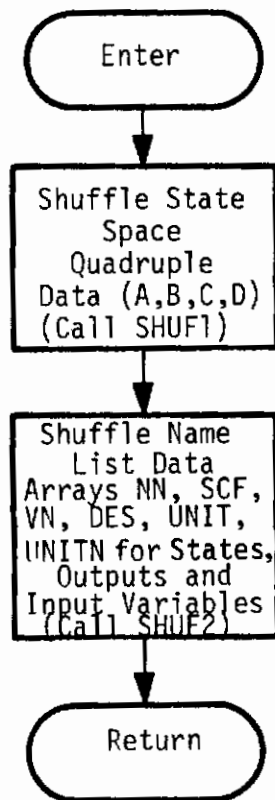


Figure 51. Subroutine SHUFF Flow Chart



# Contrails

```

SUBROUTINE SHUFF(A,R,C,D,NNS,VNS,DESS,UNITS,
1 INNO,VNO,DESO,UNITO,NNI,VNI,DESI,UNITI,
2 NSHUF0,NSHUF1,NSHUF2,DUMMY1,DUMMY2,NX,NR,NU,NXM,NRM,NUM,
3 NDM11,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 OUTPUTS
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 NSHUF0(NXM),NSHUF1(NRM),NSHUF2(NUM)
C DIMENSION DUMMY1(NDM11,NDM12),DUMMY2(NDM21,NDM22)
C
C SHUFFLE A R C D MATRICES
C
C CALL SHUF1(A,NSHUF0,NSHUF1,DUMMY1,NXM,NXM,NX,NX,NDM11,NDM12)
C CALL SHUF1(R,NSHUF0,NSHUF1,DUMMY1,NXM,NUM,NX,NI,NDM11,NDM12)
C CALL SHUF1(C,NSHUF0,NSHUF1,DUMMY1,NRM,NXM,NR,NX,NDM11,NDM12)
C CALL SHUF1(D,NSHUF0,NSHUF1,DUMMY1,NRM,NUM,NR,NI,NDM11,NDM12)
C
C SHUFFLE SCALING, UNIT AND DESCRIPTION ARRAYS
C
C CALL SHUF2(NNS,VNS,DESS,UNITS,NSHUF0,DUMMY1,DUMMY2,
1 NX,NX,NDM11,NDM12,NDM21,NDM22)
C CALL SHUF2(NNO,VNO,DESO,UNITO,NSHUF0,DUMMY1,DUMMY2,
1 NRM,NR,NDM11,NDM12,NDM21,NDM22)
C CALL SHUF2(NNI,VNI,DESI,UNITI,NSHUF1,DUMMY1,DUMMY2,
1 NUM,NI,NDM11,NDM12,NDM21,NDM22)
C RETURN
C 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

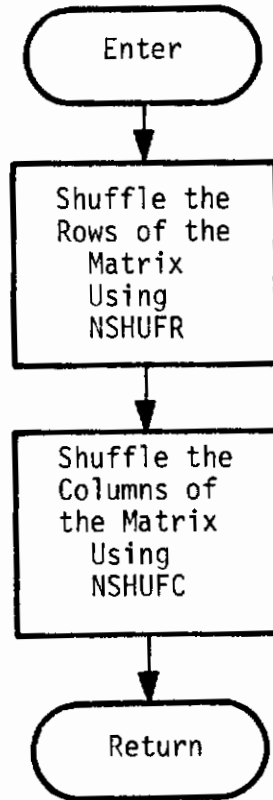


Figure 53. Subroutine SHUFF1 Flow Chart

```

SUBROUTINE SHUF1(ARCD,NSHUF1,NSHUF2,DUMMY1,NRM,NCM,NR,NC,
INDM11,INDM12)
C
C PURPOSE - TO SHUFFLE THE MATRIX ARCD
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C DATE WRITTEN - 1975
C
C ARGUMENT LIST
C   ABCD      IN/OUT  MATRIX TO BE SHUFFLED
C   NSHUF1    INPUT   ROW SHUFFLING ARRAY
C   NSHUF2    INPUT   COLUMN SHUFFLING ARRAY
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 OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
C DIMENSION ARCD(NRM,NCM),DUMMY1(INDM11,INDM12),NSHUF1(NRM)
C DIMENSION NSHUF2(NCM)
C DO 120 I=1,NR
C   II=NSHUF1(I)
C DO 120 J=1,NC
120 DUMMY1(I,J)=ARCD(II,J)
C DO 140 J=1,NC
C   JJ=NSHUF2(J)
C DO 140 I=1,NR
140 ARCD(I,J)=DUMMY1(I,JJ)
C RETURN
C END
```

Figure 54. Subroutine SHUF1 Program Listing

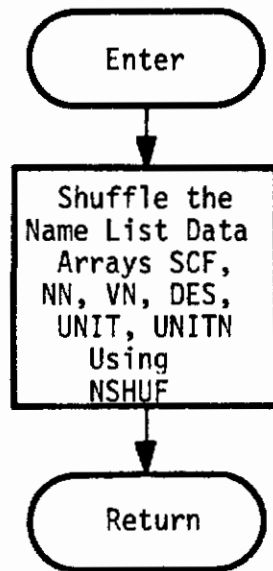


Figure 55. Subroutine SHUF2 Flow Chart

```

SUBROUTINE SHUF2(NN,VN,DES,UNIT,NSHUF,DUMMY3,DUMMY1,
INM,N,NDM1,NDM2,NDM21,NDM22)
C
C PURPOSE - TO SHUFFLE NAME LIST ARRAYS
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C DATE WRITTEN - 1975
C
C ARGUMENT LIST
C NN IN/OUT NUMBER ARRAY
C VN IN/OUT VARIABLE NAME ARRAY
C DES IN/OUT DESCRIPTION ARRAY
C UNIT IN/OUT UNIT ARRAY
C NSHUF INPUT SHUFFLING ARRAY
C NM INPUT MAXIMUM NUMBER OF SYSTEM VARIABLES
C N INPUT NUMBER OF SYSTEM VARIABLES
C OTHER PARAMETERS ARE DEFINED IN CALLING PROGRAM
C
DIMENSION NN(NM),VN(NM,2),DES(NM,16),UNIT(NM,4)
DIMENSION DUMMY3(NDM1,NDM2),DUMMY1(NDM21,NDM22)
DIMENSION NSHUF(NM)
INTEGER DUMMY1
DO 16 I=1,N
II=NSHUF(I)
DUMMY1(I,I)=NN(II)
DO 12 J=1,2
JJ=1+J
120 DUMMY3(JJ,I)=VN(II,J)
DO 14 J=1,10
JJ=3+J
140 DUMMY3(JJ,I)=DES(II,J)
DO 16 J=1,4
JJ=13+J
160 DUMMY3(JJ,I)=UNIT(II,J)
DO 24 I=1,N
NN(I)=DUMMY1(I,I)
DO 20 J=1,2
JJ=1+J
200 VN(I,J)=DUMMY3(JJ,I)
DO 22 J=1,10
JJ=3+J
220 DES(I,J)=DUMMY3(JJ,I)
DO 24 J=1,4
JJ=13+J
240 UNIT(I,J)=DUMMY3(JJ,I)
RETURN
END
SHUF2 2
SHUF2 3
SHUF2 4
SHUF2 5
SHUF2 6
SHUF2 7
SHUF2 8
SHUF2 9
SHUF2 10
SHUF2 11
SHUF2 12
SHUF2 13
SHUF2 14
SHUF2 15
SHUF2 16
SHUF2 17
SHUF2 18
SHUF2 19
SHUF2 20
SHUF2 21
SHUF2 22
SHUF2 23
SHUF2 24
SHUF2 25
SHUF2 26
SHUF2 27
SHUF2 28
SHUF2 29
SHUF2 30
SHUF2 31
SHUF2 32
SHUF2 33
SHUF2 34
SHUF2 35
SHUF2 36
SHUF2 37
SHUF2 38
SHUF2 39
SHUF2 40
SHUF2 41
SHUF2 42
SHUF2 43
SHUF2 44
SHUF2 45
SHUF2 46
SHUF2 47

```

Figure 56. Subroutine SHUF2 Program Listing

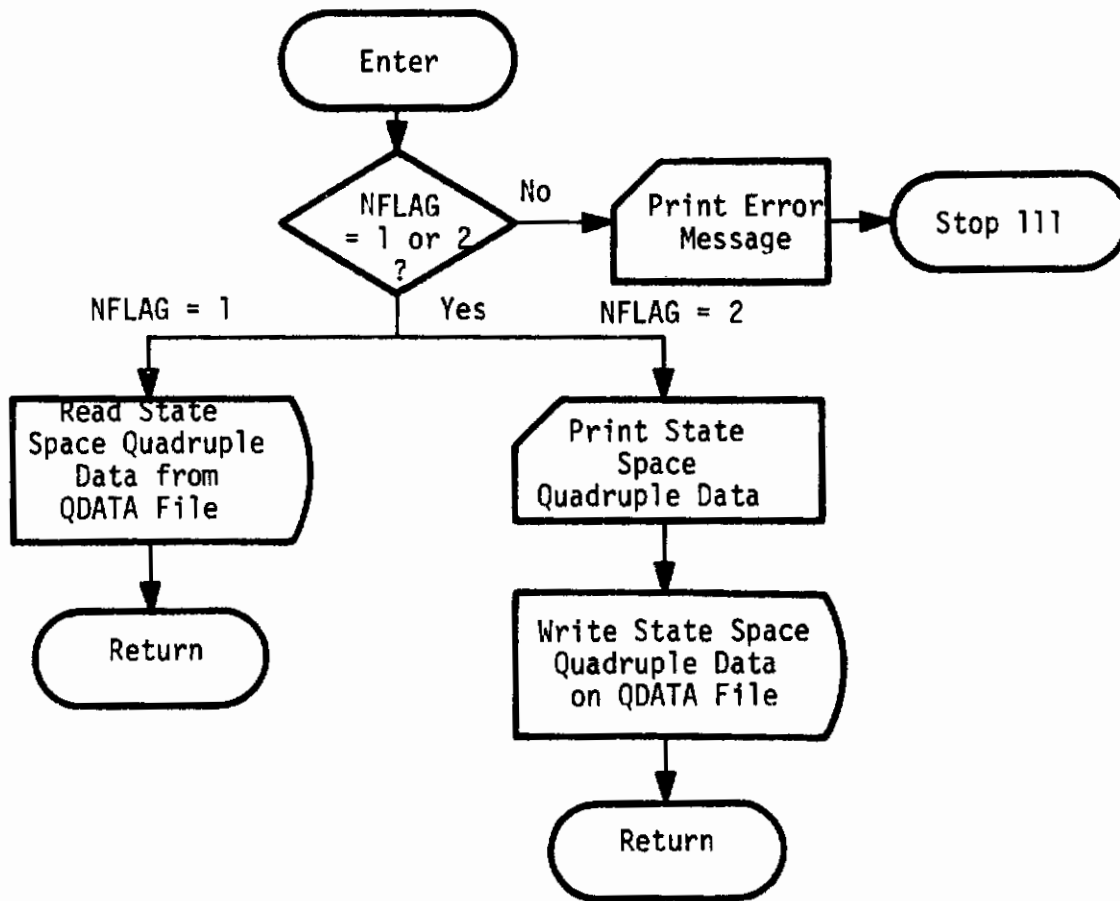


Figure 57. Subroutine QDIO Flow Chart

# Contrails

```

SUBROUTINE QDIO(A,C,D,Q,NX,NR,NU,NXM,NRM,NUM,NXA,NRA,NUA,
INRI,NP2,NR3,NU1,NU2,NU3,T,IQ,IPRINT,IW,JQ,LABEL,MARK,
2LOCATE,NULL,INSERT,NFLAG)
C
C PURPOSE - TO READ AND WRITE QUADRUPLE DATA
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C DATE WRITTEN - 1975
C
C SUBPROGRAMS CALLED
C   ERPM
C   MPDS
C   FILE
C
C ARGUMENTS LIST
C   A      IN/OUT  STATE TRANSITION MATRIX
C   B      IN/OUT  CONTROL INPUT MATRIX
C   C      IN/OUT  STATE OUTPUT MATRIX
C   D      IN/OUT  CONTROL OUTPUT MATRIX
C   Q      IN/OUT  QUADRATIC WEIGHTS MATRIX
C   NX     IN/OUT  NUMBER OF STATES
C   NR     IN/OUT  NUMBER OF OUTPUTS
C   NU     IN/OUT  NUMBER OF INPUTS
C   NXM    INPUT   MAXIMUM NUMBER OF STATES
C   NRM    INPUT   MAXIMUM NUMBER OF OUTPUTS
C   NUM    INPUT   MAXIMUM NUMBER OF INPUTS
C   NXA    IN/OUT  NO OF STATES WITHOUT IMPLICIT MODEL
C   NRA    IN/OUT  NO OF OUTPUTS WITHOUT IMPLICIT MODEL
C   NUA    IN/OUT  NO OF INPUTS WITHOUT IMPLICIT MODEL
C   NR1    IN/OUT  NO OF DESIGN OUTPUTS
C   NR2    IN/OUT  NO OF PERFORMANCE OUTPUTS
C   NR3    IN/OUT  NO OF SENSOR OUTPUTS
C   NU1    IN/OUT  NO OF CONTROL INPUTS
C   NU2    IN/OUT  NO OF GUST INPUTS
C   NU3    IN/OUT  NO OF COMMAND INPUTS
C   T      IN/OUT  SAMPLE TIME
C   IQ     INPUT   FLAG INDICATING USAGE OF Q
C   IPRINT INPUT   PRINT CONTROL FLAG
C   IW     INPUT   FILE NO FOR LINE PRINTER
C   JQ     INPUT   FILE NO FOR QUADRUPLE DATA FILE
C   LABEL  INPUT   LABEL NAME FOR QUADRUPLE DATA
C   MARK   INPUT   HOLLERITH SS...S
C   LOCATE INPUT   HOLLERITH LOCA
C   NULL   INPUT   HOLLERITH NULL
C   INSERT INPUT   HOLLERITH INSE
C   NFLAG  INPUT   CONTROLS ENTRY POINT IN THE SUBROUTINE
C
C DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
C DIMENSION Q(NRM,NRM),LABEL(20),MARK(20)
C IF(NFLAG,NE.2)GO TO 220
C IF(IPRINT,LT.2)GO TO 200
C
C PRINT QUADRUPLE DATA
C
C WRITE(IW,120)
120 FORMAT(//,20X,22H** QUADRUPLE DATA **//)
C IF(T,IE.0.0)GO TO 140
C CALL MPRS(A,NXM,NXM,NX,NX,T,4HA )
C CALL MPRS(B,NXM,NUM,NX,NU,T,4HB )
C CALL MPRS(C,NRM,NXM,NO,NX,T,4HC )
C CALL MPRS(D,NRM,NUM,NR,NU,T,4HD )
C GO TO 160
140 CONTINUE
C CALL MPRS(A,NXM,NXM,NX,NX,T,4HF )

```

Figure 58. Subroutine QDIO Program Listing

# Contrails

```

CALL MPRS(B,NXM,NUM,NX,NU,T,4HG )           QDIO 65
CALL MPRS(C,NRM,NXM,NR,NX,T,4HH )           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,NP,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(NFLAG,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)



# Contrails

RETURN  
END

0010 131  
0010 132

Figure 58. Subroutine QDIO Program Listing (Concluded)

# Contrails

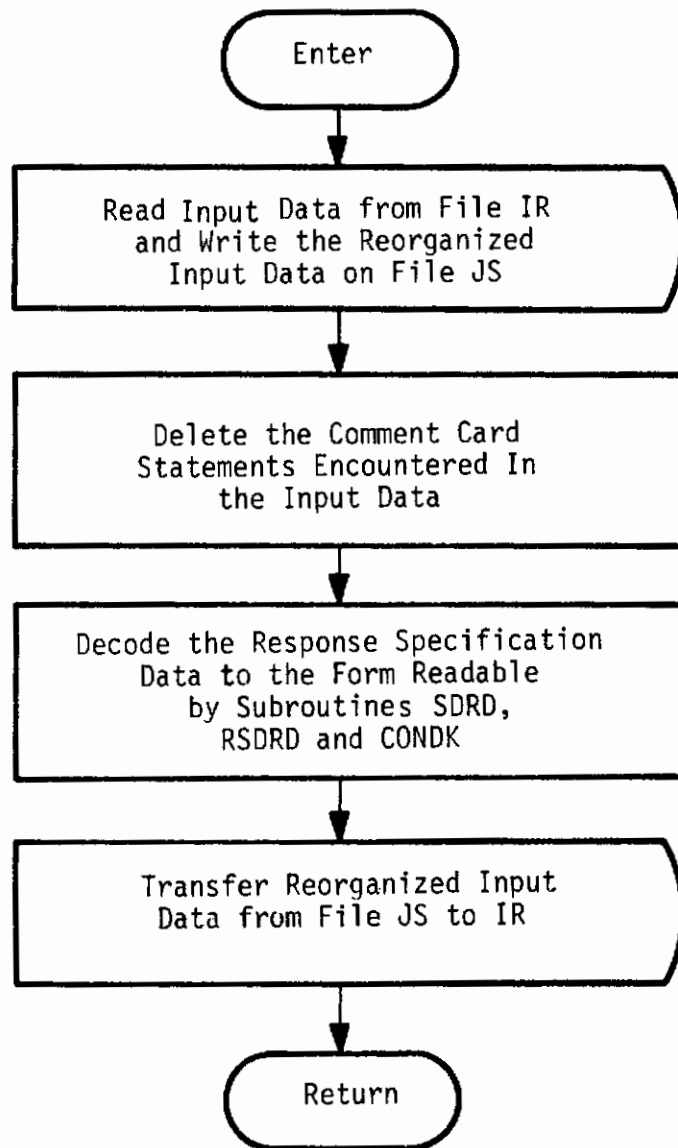


Figure 59. Subroutine IDRO Flow Chart

# Contracts

```

C      SUBROUTINE IDRO(IR,IW,JS)                                IDRO  2
C      PURPOSE - TO REORGANIZE INPUT DATA                    IDRO  3
C      ANALYSIS - A F KONAR / J K VAHESH - THE HONEYWELL INC  IDRO  4
C      DATE WRITTEN - 1975                                     IDRO  5
C      ARGUMENTS LIST                                         IDRO  6
C      IR          INPUT      FILE NO FOR CARD READER BUFFER  IDRO  7
C      IW          INPUT      FILE NO FOR LINE PRINTER        IDRO  8
C      JS          INPUT      FILE NO FOR SCRATCH FILE         IDRO  9
C
C      DIMENSION CARD1(20),CARD2(80),CARD3(76),CARD4(20),CHEAD(6) IDRO 10
C      DATA CHEAD /4HCONS,4HSELE,4HRTA,4HTRUN,4HRESI,4HSCAL/  IDRO 11
C      DATA HRRR,HEHNR,HR,HHP,HEP /4H      ,4HEND ,1H ,1H(,1H)/ IDRO 12
C      DATA HCOM,HOAS,HDOT /1H,,1H,,1H,/  IDRO 13
C      DATA HX,HR,HU/1HX,1HR,1HU/  IDRO 14
C      DATA HCR,HE/2HC ,1HE/  IDRO 15
C      NID=6  IDRO 16
C      REWIND IR  IDRO 17
C      REWIND JS  IDRO 18
C      CARD3(2)=HHP  IDRO 19
C      CARD3(4)=HEP  IDRO 20
C      DO 10 I=5,76  IDRO 21
100  CARD3(I)=HR  IDRO 22
C      CARD4(1)=HEHNR  IDRO 23
C      DO 11 I=2,20  IDRO 24
110  CARD4(I)=HRRR  IDRO 25
C
C      READ CARD IMAGES FROM FILE IR AND IGNORE THE COMMENT CARDS  IDRO 26
C
C      120 CONTINUE  IDRO 27
C      READ(IR,140)CARD1  IDRO 28
140  FORMAT(20A4)  IDRO 29
C      IF(EOF(IR))540,160  IDRO 30
160  CONTINUE  IDRO 31
C      DECODE(4,170,CARD1(1))CC,DUMMY  IDRO 32
170  FORMAT(A2,A?)  IDRO 33
C      IF(CC.EQ.HCR)GO TO 120  IDRO 34
C      WRITE(JS,140)CARD1  IDRO 35
C      DO 18 J=1,NID  IDRO 36
C      IF(CARD1(J).EQ.CHEAD(J))GO TO 200  IDRO 37
180  CONTINUE  IDRO 38
C      GO TO 120  IDRO 39
C
C      READ RESPONSE SPECIFICATION DATA AND ENCODE  IDRO 40
C      INTO SIMPLER RESPONSE SPECIFICATIONS  IDRO 41
C
C      200 CONTINUE  IDRO 42
C      READ(IR,220)CARD2  IDRO 43
220  FORMAT(80A1)  IDRO 44
C      I=0  IDRO 45
240  CONTINUE  IDRO 46
C      I=I+1  IDRO 47
C      IF(I.GE.81)GO TO 200  IDRO 48
C      IF(CARD2(I).EQ.HE)GO TO 520  IDRO 49
C      IF(CARD2(I).EQ.HR)GO TO 240  IDRO 50
C      IF(CARD2(I).EQ.HCOM)GO TO 240  IDRO 51
C      IF(CARD2(I).EQ.HOAS)GO TO 400  IDRO 52
C      IF(CARD2(I).EQ.HDOT)GO TO 520  IDRO 53
C      IF((CARD2(I).NE.HX).AND.(CARD2(I).NE.HP).AND.(CARD2(I).NE.HU))  IDRO 54
160  TO 200  IDRO 55
C      I=I+1  IDRO 56
C      IF(CARD2(I).EQ.HRP)GO TO 280  IDRO 57

```

Figure 60. Subroutine IDRO Program Listing

# Contrails

I=I+1	IDRO 65
IF (CARD2(I).EQ.HRP) GO TO 26	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),HB,HB	IDRO 73
GO TO 300	IDRO 74
280 CONTINUE	IDRO 75
ENCODE (4,250,CARD3(1))CARD2(I-1),HH,HH,HB	IDRO 76
300 CONTINUE	IDRO 77
I=I+2	IDRO 78
IF (CARD2(I).EQ.HEP) GO TO 32	IDRO 79
I=I+1	IDRO 80
IF (CARD2(I).NE.HEP) 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(1).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.HEP) GO TO 44	IDRO 104
I=I+1	IDRO 105
IF (CARD2(I).NE.HEP) 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))NN	IDRO 112
IF (NN.LE.NP) GO TO 420	IDRO 113
NPP1=NP+1	IDRO 114
DO 500 J=NPP1,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
ENDFILE 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)

# Contrails

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(IN,640)CARD2	IDRO 146
640	FORMAT(1H1,///,1X,32HERROR IN REORGANIZING INPUT DATA,///,1X,80A1)	IDRO 147
	STOP 111	IDRO 148
	END	IDRO 149

Figure 60. Subroutine IDRO Program Listing (Concluded)

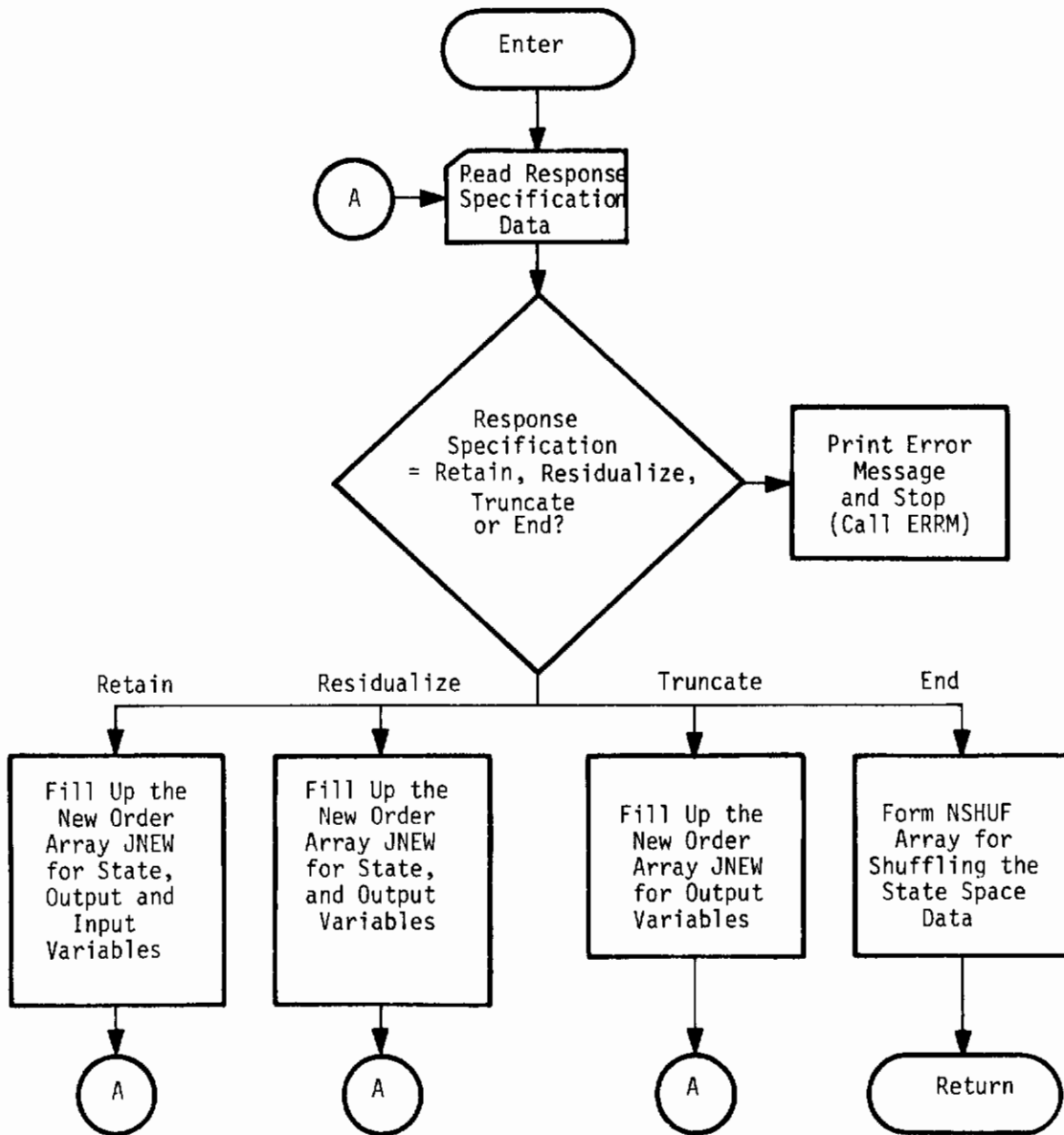


Figure 61. Subroutine RSDRD Flow Chart



# Contrails

	JNEWS(KS)=N	RSDRD 68
	GO TO 160	RSDRD 69
200	CONTINUE	RSDRD 70
	IF(HD,NE,HUBBB)GO TO 210	RSDRD 71
C		RSDRD 72
C	FORM JNEWI ARRAY FOR INPUTS	RSDRD 73
C		RSDRD 74
	KU=KU+1	RSDRD 75
	JNEWI(KU)=N	RSDRD 76
	GO TO 160	RSDRD 77
210	CONTINUE	RSDRD 78
	IF(HD,NE,HRBBB)CALL ERRM(2,4HRS DR,4HD .5,0,IW)	RSDRD 79
C		RSDRD 80
C	FORM JNEWO ARRAY FOR OUTPUTS	RSDRD 81
C		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,HRES)GO TO 360	RSDRD 93
C		RSDRD 94
C	READ SHUFFLE DATA FOR THE RESIDUALIZED SYSTEM VARIABLES	RSDRD 95
C		RSDRD 96
	IF(CARD(4),NE,HSTAT)CALL ERRM(3,4HRS DR,4HD .5,0,IW)	RSDRD 97
260	CONTINUE	RSDRD 98
	READ(IR,100)HD,N	RSDRD 99
	IF(HD,EQ,HENDB)GO TO 350	RSDRD100
	IF(HD,NE,HXBBB)GO TO 280	RSDRD101
C		RSDRD102
C	FORM JNEWS ARRAY FOR STATES	RSDRD103
C		RSDRD104
	KS=KS+1	RSDRD105
	JNEWS(KS)=N	RSDRD106
	GO TO 260	RSDRD107
280	CONTINUE	RSDRD108
	IF(HD,NE,HRBBB)CALL ERRM(4,4HRS DR,4HD .5,0,IW)	RSDRD109
C		RSDRD110
C	FORM JNEWO ARRAY FOR OUTPUTS	RSDRD111
C		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,HTRUN)CALL ERRM(5,4HRS DR,4HD .5,0,IW)	RSDRD122
	IF(CARD(3),NE,HBSTA)CALL ERRM(6,4HRS DR,4HD .5,0,IW)	RSDRD123
C		RSDRD124
C	READ SHUFFLE DATA FOR THE TRUNCATED SYSTEM VARIABLES	RSDRD125
C		RSDRD126
400	CONTINUE	RSDRD127
	READ(IR,100)HD,N	RSDRD128
	IF(HD,EQ,HENDB)GO TO 410	RSDRD129
	IF(HD,NE,HRBBB)CALL ERRM(7,4HRS DR,4HD .5,0,IW)	RSDRD130
C		RSDRD131
C	FORM JNEWO ARRAY FOR OUTPUTS	RSDRD132
C		RSDRD133

Figure 62. Subroutine RSDRD Program Listing (Continued)



# Contrails

```

      KR=KR+1
      JNEWO(KR)=N
      GO TO 400
410  CONTINUE
      NRT=KR-NRR
      GO TO 120
420  CONTINUE
      IF(IRS.EQ.0)RETURN
      IF(IPRINT.EQ.6)CALL DEBUG(2,4HRS DR,4HD ,5.0,1W)
      II=NXP
C
C      FORM SHUFFLE ARRAY FOR STATES
C
      DO 470 I=1,NX
      DO 430 J=1,NXN
      JJ=J
      IF(I.EQ.JNEWS(JJ))GO TO 460
430  CONTINUE
      IF(NXP.EQ.0)GO TO 450
      DO 440 J=1,NXP
      JJ=NXN+J
      IF(I.EQ.JNEWS(JJ))GO TO 460
440  CONTINUE
450  CONTINUE
      II=II+1
      NSHUF(I)=I
      GO TO 470
460  CONTINUE
      NSHUF(JJ)=I
470  CONTINUE
      IF(IPRINT.EQ.6)CALL DEBUG(3,4HRS DR,4HD ,5.0,1W)
C
C      FORM SHUFFLE ARRAY FOR OUTPUTS
C
      II=NRN
      IF(NRN.LE.0)II=NRT+NRR
      IF(II.EQ.0)IN=1
      DO 570 I=1,NR
      IF(NRN.LE.0)GO TO 520
      DO 510 J=1,NRN
      JJ=J
      IF(I.EQ.JNEWO(JJ))GO TO 560
510  CONTINUE
      GO TO 550
520  CONTINUE
      IF(NRR.EQ.0)GO TO 535
      DO 530 J=1,NRR
      JJ=J
      IF(I.EQ.JNEWO(JJ))GO TO 560
530  CONTINUE
535  CONTINUE
      IF(NRT.EQ.0)GO TO 550
      DO 540 J=1,NRT
      JJ=NRP+J
      IF(I.EQ.JNEWO(JJ))GO TO 560
540  CONTINUE
550  CONTINUE
      II=II+1
      NSHUF(I)=I
      GO TO 570
560  CONTINUE
      NSHUF(JJ)=I
570  CONTINUE
      IF(IN.EQ.1)NRN=NR
      IN=0
      IF(IPRINT.EQ.6)CALL DEBUG(4,4HRS DR,4HD ,5.0,1W)
RSDRD134
RSDRD135
RSDRD136
RSDRD137
RSDRD138
RSDRD139
RSDRD140
RSDRD141
RSDRD142
RSDRD143
RSDRD144
RSDRD145
RSDRD146
RSDRD147
RSDRD148
RSDRD149
RSDRD150
RSDRD151
RSDRD152
RSDRD153
RSDRD154
RSDRD155
RSDRD156
RSDRD157
RSDRD158
RSDRD159
RSDRD160
RSDRD161
RSDRD162
RSDRD163
RSDRD164
RSDRD165
RSDRD166
RSDRD167
RSDRD168
RSDRD169
RSDRD170
RSDRD171
RSDRD172
RSDRD173
RSDRD174
RSDRD175
RSDRD176
RSDRD177
RSDRD178
RSDRD179
RSDRD180
RSDRD181
RSDRD182
RSDRD183
RSDRD184
RSDRD185
RSDRD186
RSDRD187
RSDRD188
RSDRD189
RSDRD190
RSDRD191
RSDRD192
RSDRD193
RSDRD194
RSDRD195
RSDRD196
RSDRD197
RSDRD198
RSDRD199
```

Figure 62. Subroutine RSDRD Program Listing (Continued)

```
C
C   FORM SHUFFLE ARRAY FOR INPUTS
C
  II=NUM
  IF (II.EQ.0) IN=1
  DO 670 I=1,NU
  IF (NUM.LE.0) GO TO 640
  DO 630 J=1,NUN
  JJ=J
  IF (I.EQ.JNEWI(JJ)) GO TO 660
630 CONTINUE
640 CONTINUE
  II=II+1
  NSHUF1(II)=I
  GO TO 670
660 CONTINUE
  NSHUF1(JJ)=I
670 CONTINUE
  IF (IN.EQ.1) NUN=NU
  IF (IP>INT.EQ.6) CALL DEBUG(5,4HRSDR,4WD  *5,0,IW)
  IF (IPRINT.LT.6) GO TO 680
  WRITE (IW,675) NX,NXN,NR,NRN,NU,NUN
675 FORMAT (IX,20(I2,IX))
  WRITE (IW,675) JNEWS,JNFWO,JNEWI
  WRITE (IW,675) NSHUF1,NSHUFO,NSHUF1
680 CONTINUE
  RETURN
  END
```

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

Figure 62. Subroutine RSDRD Program Listing (Concluded)

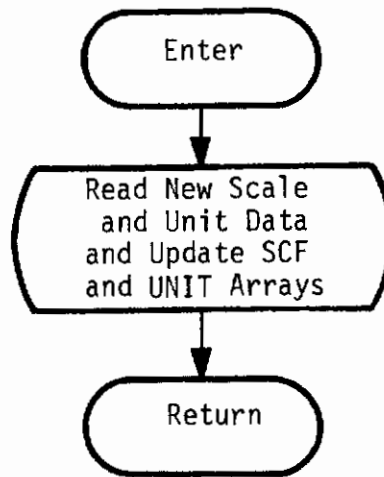


Figure 63. Subroutine SDRD Flow Chart

# Contrails

```

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  ANALISIS - A F KONAR / J K MAHESH - THE HONEYWELL INC  SDRD  6
C  DATE WRITTEN - 1975                    SDRD  7
C                                          SDRD  8
C  SUBPROGRAMS CALLED                     SDRD  9
C    DEBUG                                SDRD 10
C    ERRM                                  SDRD 11
C                                          SDRD 12
C  ARGUMENTS LIST                         SDRD 13
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),UNITS(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 HENDB,HXBRR,HRBBR,HUBRR/4HEND ,4HX  ,4HR  ,4HU  / SDRD 30
C  DATA HSCAL/4HSCAL/                                 SDRD 31
C  ISC=0                                                SDRD 32
C  IF(IPRINT.EQ.6)CALL DEBUG(1,4HSDRD,4H  ,5,0,1W)    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,280)HD,N,SC,(UN(J),J=1,4),(UNN(J),J=1,4)  SDRD 47
C 280 FORMAT(A4,1X,I2,3X,E14.6,6X,4A4,4X,4A4)        SDRD 48
C  IF(IPRINT.EQ.6)CALL DEBUG(2,4HSDRD,4H  ,5,0,1W)    SDRD 49
C  IF(HD.EQ.HENDB)RETURN                             SDRD 50
C  ISC=1                                              SDRD 51
C  IF(HD.NE.HXBRR)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
C  UNITNS(N,J)=UNN(J)                                  SDRD 58
300 UNITS(N,J)=UNN(J)                                  SDRD 59
C  GO TO 260                                          SDRD 60
C 320 CONTINUE                                       SDRD 61
C  IF(HD.NE.HRBBR)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

# Contrails

	UNITN0(N,J)=UNN(J)	SDRD 68
340	UNITO(N,J)=UNN(J)	SDRD 69
	GO TO 260	SDRD 70
350	CONTINUE	SDRD 71
	IF(HD.NE.HURRR)CALL EPRM(2,4HSDRD,4H .6*0,1W)	SDRD 72
C		SDRD 73
C	FOR INPUTS	SDRD 74
C		SDRD 75
	SCFI(0)=SC	SDRD 76
	DO 38 J=1,4	SDRD 77
	UNITN1(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)

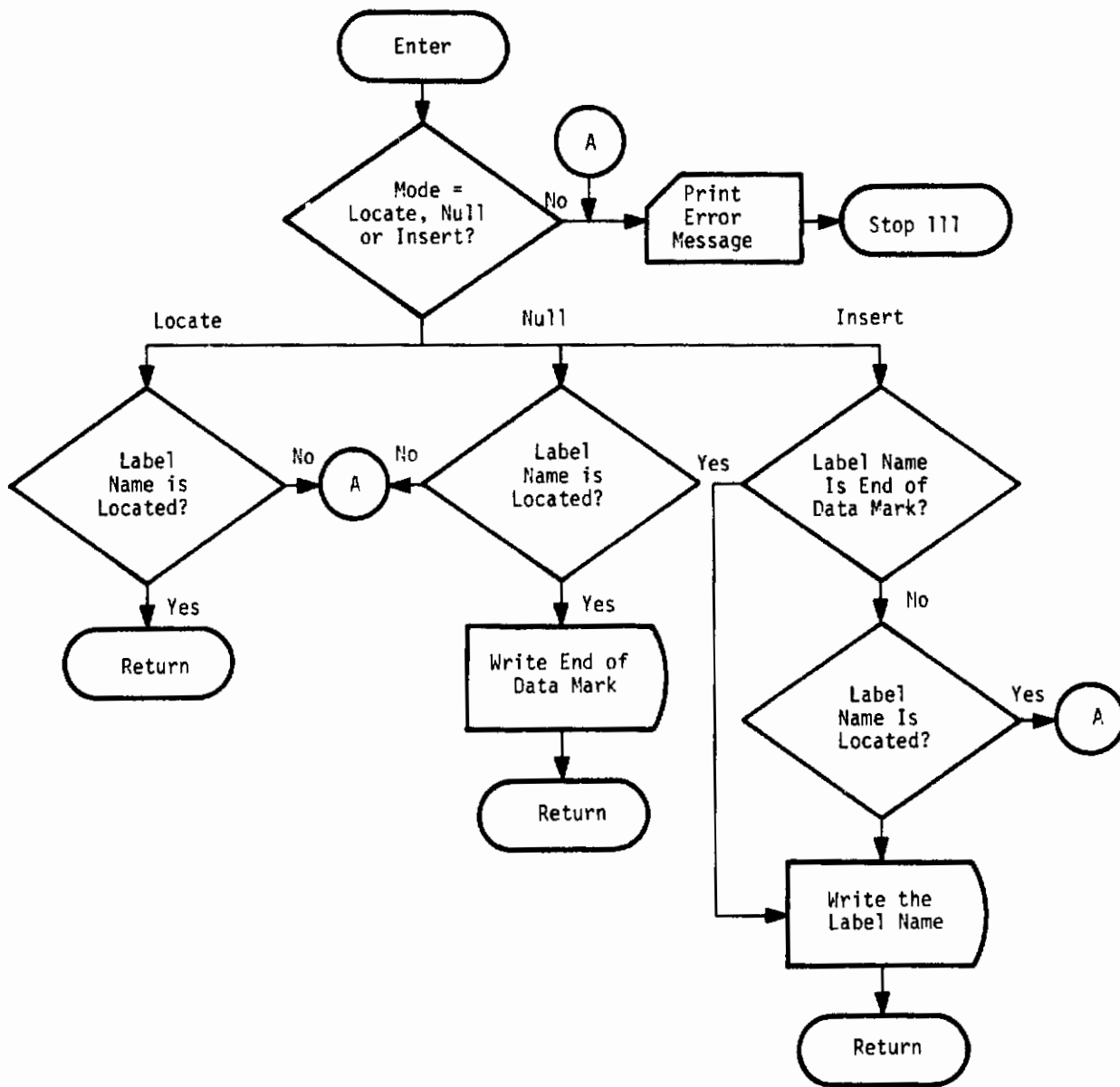


Figure 65. Subroutine FILE Flow Chart

# Contrails

```

SUBROUTINE FILE(NFILE,MODE,NAME)
C
C PURPOSE - TO POSITION THE DATA FILE
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C DATE WRITTEN - 1975
C
C ARGUMENT LIST
C NFILE FILE NUMBER OF THE DISK FILE
C MODE MODE PARAMETER (LOCATE,INSERT,NULL)
C NAME LABEL NAME
C
DIMENSION NAME(20),LABEL(20),LAST(20)
INTEGER HDOLR
DATA LOCATE,INSERT,NULL/4HLOCATE,4HINSE,4HNULL/
DATA HDOLR/4H5555/
IW=9
DO 10 I=1,20
100 LAST(I)=HDOLR
IF(MODE.EQ.LOCATE)GO TO 140
IF(MODE.EQ.INSERT)GO TO 120
IF(MODE.EQ.NULL)GO TO 140
C
C PRINT ERROR MESSAGE
C
WRITE(IW,110)
110 FORMAT(1H1,/,/,1X,45HMODE OF OPERATION FOR DATA FILE NOT SPECIFIED)
STOP 111
C
C CHECK IF END OF DATA MARK IS BEING INSERTED
C
120 CONTINUE
DO 13 I=1,20
IF(NAME(I).NE.LAST(I))GO TO 140
130 CONTINUE
C
C WRITE END OF DATA MARK AND ALSO AN END OF FILE
C
WRITE(NFILE)(NAME(I),I=1,20)
ENDFILE NFILE
REWIND NFILE
RETURN
C
C CHECK IF LABEL ON FILE MATCHES WITH NAME
C
140 CONTINUE
REWIND NFILE
150 CONTINUE
READ(NFILE)(LABEL(I),I=1,20)
DO 16 I=1,20
IF(LABEL(I).NE.NAME(I))GO TO 170
160 CONTINUE
GO TO 220
C
C CHECK IF LABEL IS THE END OF DATA MARK
C
170 CONTINUE
DO 18 I=1,20
IF(LABEL(I).NE.LAST(I))GO TO 200
180 CONTINUE
IF(MODE.EQ.INSERT)GO TO 210
C
C PRINT ERROR MESSAGE
C

```

Figure 66. Subroutine FILE Program Listing

# Contrails

```
WRITE(IW,190)NAME,NFILE FILE 65
190 FORMAT(1H),//,1X,20A4,//,1X,20HCANNOT BE FOUND ON DATA FILE ,I2) FILE 66
STOP 111 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(1H),//,1X,20A4,//,1X,2)ALREADY ON DATA FILE ,I2) FILE 94
STOP 111 FILE 95
END FILE 96
```

Figure 66. Subroutine FILE Program Listing (Concluded)



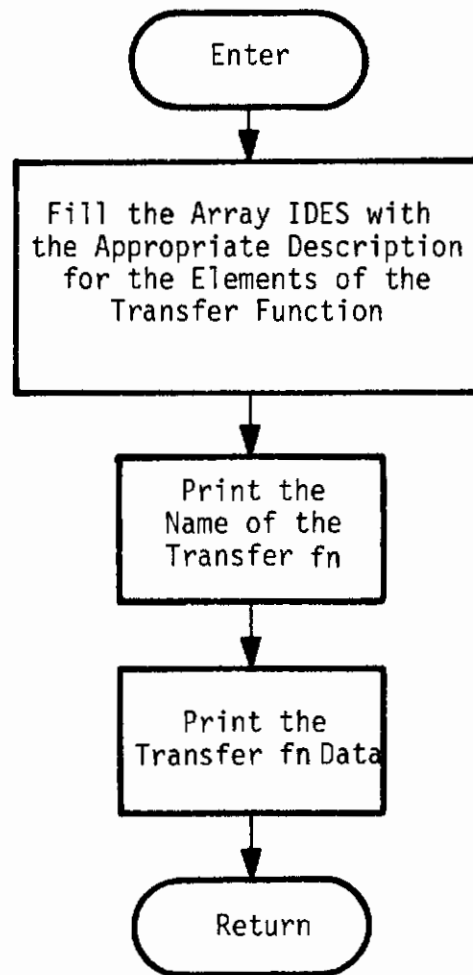


Figure 67. Subroutine TPR Flow Chart

# Contrails

```

SUBROUTINE TPR(H,NE,NEM,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 HONEYWELL INC          TPR  5
C DATE WRITTEN - 1975                                           TPR  6
C                                                                    TPR  7
C ARGUMENTS LIST                                                TPR  8
C   H      INPUT      TRANSFER FUNCTION                          TPR  9
C   NE     INPUT      NO OF ELEMENTS OF THE TRANSFER FN         TPR 10
C   NEM    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 H(2,NEM),NAME(3)                                    TPR 22
C COMMON /SC1/ IDES(6,3),MI,I,K                                  TPR 23
C                                                                    TPR 24
C CHECK FOR DIMENSION ERROR                                     TPR 25
C                                                                    TPR 26
C IF((NEM,NE,6).OR.(NE.GT,NEM))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,FO,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,(H))                            TPR 54
C                                                                    TPR 55
C PRINT THE TRANSFER FN                                         TPR 56
C                                                                    TPR 57
C MI=6-4E+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)(H(1,I),I=1,NE)                                  TPR 61
220 FORMAT(/,1X,9HNUMERATOR,6X,5G14.6)                          TPR 62
C WRITE(IW,240)(H(2,I),I=1,NE)                                  TPR 63
240 FORMAT(/,1X,11HDENOMINATOR,4X,5G14.6)                       TPR 64

```

Figure 68. Subroutine TPR Program Listing

# Contrails

	RETURN	TPR	65
C		TPR	66
C	PRINT ERROR MESSAGE	TPR	67
C		TPR	68
260	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)

# Contracts

```
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      ARGUMENTS LIST                                         HPR    6
C      CARD          INPUT      SYSTEM LABEL NAME           HPR    7
C      IW           INPUT      FILE NO FOR LINE PRINTER     HPR    8
C      DIMENSION CARD(20)                                     HPR    9
C      INTEGER CARD                                          HPR   10
C      WRITE(IW,120)                                          HPR   11
120  FORMAT(1H1,/,/,2CX,88(1H*))                             HPR   12
C      WRITE(IW,140)                                          HPR   13
140  FORMAT(20X,1H*,86X,1H*)                                  HPR   14
C      WRITE(IW,160)CARD                                       HPR   15
160  FORMAT(20X,1H*,2X,2CA4,4X,14*)                          HPR   16
C      WRITE(IW,140)                                          HPR   17
C      WRITE(IW,180)                                          HPR   18
180  FORMAT(20X,88(1H*))                                       HPR   19
C      RETURN                                                 HPR   20
C      RETURN                                                 HPR   21
C      RETURN                                                 HPR   22
C      RETURN                                                 HPR   23
```

Figure 69. Subroutine HPR Program Listing

# Contracts

```

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

Figure 70. Subroutine IDPR Program Listing

# Contrails

```

SUBROUTINE MPRS(A,MAX,MYC,NP,NC,T,ITITLE)          MPRS  2
C                                                    MPRS  3
C PURPOSE - TO PRINT MATRIX DATA                MPRS  4
C ANALYSTS - A F KONAR / J K WAHESH - THE HONEYWELL INC MPRS  5
C DATE WRITTEN - 1975                             MPRS  6
C                                                    MPRS  7
C ARGUMENTS LIST                                  MPRS  8
C   A      INPUT  MATRIX DATA                    MPRS  9
C   MAX    INPUT  MAXIMUM NO OF ROWS              MPRS 10
C   MYC    INPUT  MAXIMUM NO OF COLUMNS          MPRS 11
C   NR     INPUT  NUMBER OF ROWS                  MPRS 12
C   NC     INPUT  NUMBER OF COLUMNS              MPRS 13
C   T      INPUT  SAMPLE TIME                     MPRS 14
C   ITITLE INPUT  TITLE OR NAME OF THE MATRIX    MPRS 15
C                                                    MPRS 16
C   DIMENSION A(MAX,MAX)                          MPRS 17
C                                                    MPRS 18
C   BEGINNING OF COLUMN SIZE LOOP                  MPRS 19
C                                                    MPRS 20
C   JC =                                           MPRS 21
160 CONTINUE                                       MPRS 22
C   IC = IC + 1                                    MPRS 23
C   JC = IC + 1                                    MPRS 24
C   IF (JC .GT. NC) JC = NC                       MPRS 25
C                                                    MPRS 26
C   BEGINNING OF ROW SIZE LOOP                     MPRS 27
C                                                    MPRS 28
C   JR=0                                           MPRS 29
150 CONTINUE                                       MPRS 30
C                                                    MPRS 31
C   PRINT MATRIX NAME AND SIZE                     MPRS 32
C                                                    MPRS 33
C   IF (T.EQ.0.0) WRITE(9,80) ITITLE,NR,NC        MPRS 34
80  FORMAT (//8H MATRIX ,A4.16X,7HSIZE = ,I2.3H X ,I2) MPRS 35
C   IF (T.EQ.0.0) WRITE(9,90) ITITLE,T,NR,NC      MPRS 36
90  FORMAT (//8H MATRIX ,A4.3H(T=,E10.4.1H),2X,7HSIZE = ,I2.3H X ,I2) MPRS 37
C                                                    MPRS 38
C   PRINT COLUMN INDEX                             MPRS 39
C                                                    MPRS 40
C   WRITE (9, 160) (K, K = IC, JC)                 MPRS 41
160 FORMAT (//8X,1H(2X,13.7H-COLUMN))             MPRS 42
C   WRITE(9,170)                                    MPRS 43
170 FORMAT (/)                                     MPRS 44
C   IR=JR+1                                         MPRS 45
C   JR=JR+1300                                      MPRS 46
C   IF (JR .GT. NR) JR=NR                           MPRS 47
C   DO 18. I=IR, JR                                 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,JC)          MPRS 52
190 FORMAT (1X,13.4H-ROW,1X,13(12.4))            MPRS 53
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 160                      MPRS 62
C                                                    MPRS 63
C   RETURN TO CALLING PROGRAM                      MPRS 64

```

Figure 71. Subroutine MPRS Program Listing

# Contrails

C  
RETURN  
END

MPRS 65  
MPRS 66  
MPRS 67

Figure 71. Subroutine MPRS Program Listing (Concluded)

# Contracts

```

SUBROUTINE MPRS1(A,NRM,NCM,NR,NC,MHEAD)
C
C PURPOSE - TO PRINT LSA MATRIX DATA
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C DATE WRITTEN - 1975
C
C ARGUMENTS LIST
C   A      INPUT  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   MHEAD  INPUT  MATRIX TITLE OR NAME
C
C DIMENSION A(NRM,NCM)
COMMON /INOUT/ IP,IW,IPRINT,JN,JG,INSERT,LOCATE,NULL,MARK(20)
IF((IPRINT.NE.3).AND.(IPRINT.LT.5))RETURN
C
C WRITE NAME AND SIZE OF THE MATRIX
C
C   WRITE(IW,80)MHEAD,NR,NC
80  FORMAT(//,10X,A10,16HMATRIX , SIZE = ,I2,3H X ,I2)
C   JC=0
100 IC=JC+1
C   JC=JC+7
C   IF(JC.GT.NC)JC=NC
C   KC=JC-IC+1
C
C WRITE COLUMN HEADINGS
C
C   150 WRITE(IW,160)(K,K=IC+JC)
160 FORMAT(//,8X,7(2X,I3,7H-COLUMN,3X))
C   WRITE(IW,170)
170 FORMAT(//)
C   DO 18 I=1,NR
C
C WRITE ROW HEADINGS
C
C   180 WRITE(IW,190)I,(A(I,J),J=IC+JC)
190 FORMAT(1X,I3,4H-ROW,1X,7(E15,7))
320 IF(JC.LT.NC)GO TO 130
C   WRITE(IW,170)
C   RETURN
C   END
MPRS1  2
MPRS1  3
MPRS1  4
MPRS1  5
MPRS1  6
MPRS1  7
MPRS1  8
MPRS1  9
MPRS1 10
MPRS1 11
MPRS1 12
MPRS1 13
MPRS1 14
MPRS1 15
MPRS1 16
MPRS1 17
MPRS1 18
MPRS1 19
MPRS1 20
MPRS1 21
MPRS1 22
MPRS1 23
MPRS1 24
MPRS1 25
MPRS1 26
MPRS1 27
MPRS1 28
MPRS1 29
MPRS1 30
MPRS1 31
MPRS1 32
MPRS1 33
MPRS1 34
MPRS1 35
MPRS1 36
MPRS1 37
MPRS1 38
MPRS1 39
MPRS1 40
MPRS1 41
MPRS1 42
MPRS1 43
MPRS1 44
MPRS1 45
```

Figure 72. Subroutine MPRS1 Program Listing



# Contrails

```

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 MAHESH - 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
C DIMENSION A(NRM,NCM)                                    ZERO 13
C DO 120 I=1,NRM                                          ZERO 14
C DO 120 J=1,NCM                                          ZERO 15
120 A(I,J)=0.0                                           ZERO 16
C RETURN                                                  ZERO 17
C END                                                      ZERO 18
```

Figure 73. Subroutine ZERO Program Listing

# Contrails

```

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

# Contracts

```

C      SUBROUTINE INPT1(A,NRM,NCM,NR,NC,IR)                                INPT1  2
C      PURPOSE - TO READ ISA MATRIX DATA                                INPT1  3
C      ANALISTS - A F KONAR / J K MAHESH - THE HONEYWELL INC           INPT1  4
C      DATE WRITTEN - 1975                                             INPT1  5
C      DATE WRITTEN - 1975                                             INPT1  6
C      ARGUMENTS LIST                                                  INPT1  7
C      A      OUTPUT      MATRIX DATA                                INPT1  8
C      NRM    INPUT      MAXIMUM NUMBER OF ROWS                     INPT1  9
C      NCM    INPUT      MAXIMUM NUMBER OF COLUMNS                 INPT1 10
C      NR     INPUT      NUMBER OF ROWS                              INPT1 11
C      NC     INPUT      NUMBER OF COLUMNS                          INPT1 12
C      IR     INPUT      FILE NO FOR INPUT DATA BUFFER             INPT1 13
C      IR     INPUT      FILE NO FOR INPUT DATA BUFFER             INPT1 14
C      IR     INPUT      FILE NO FOR INPUT DATA BUFFER             INPT1 15
C      DIMENSION A(NRM,NCM)                                           INPT1 16
C      READ(1R,120)((A(I,J),J=1,NC),I=1,NR)                          INPT1 17
120  FORMAT(6G10.3)                                                  INPT1 18
C      RETURN                                                           INPT1 19
C      END                                                               INPT1 20

```

Figure 75. Subroutine INPT1 Program Listing

# Contrails

```

SUBROUTINE DEBUG(N,A1,A2,N1,N2,IW)          DEBUG  2
C                                           DEBUG  3
C PURPOSE - TO PRINT DEBUGGING MESSAGE    DEBUG  4
C ANALYSIS - A F KONAR / J K MAHESH - 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
C   WRITE(IW,120)N,A1,A2,N1,N2             DEBUG 16
120 FORMAT(//,1X,27HEXECUTION ENTERED POSITION ,I2,1X,  DEBUG 17
111HSUBROUTINE ,2A4,1X,12HIN OVERLAY (,I1,1H,,I1,1H))  DEBUG 18
RETURN                                     DEBUG 19
END                                         DEBUG 20
```

Figure 76. Subroutine DEBUG Program Listing

# Contracts

```

SUBROUTINE ERRM(N,A1,A2,N1,N2,IW)                                ERRM  2
C                                                                    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
C   WRITE(IW,120)N,A1,A2,N1,N2                                   ERRM 16
120 FORMAT(1H1,/,/,1X,27HERROR DETECTED AT POSITION ,I2,1X,     ERRM 17
111HSUBROUTINE ,2A4,1X,12HIN OVERLAY (,I1,1H,,I1,1H))          ERRM 18
C   STOP 111                                                    ERRM 19
C   END                                                            ERRM 20
```

Figure 77. Subroutine ERRM Program Listing

# Contrails

```

SUBROUTINE DERRM(M1,M2,M3,M4,MS1,MS2,MS3,MS4,N1,N2,A1,A2,IW)      DERRM  2
C                                                                    DERRM  3
C PURPOSE - TO PRINT ERROR MESSAGE WHEN DIMENSIONS FOR          DERRM  4
C SCRATCH ARRAYS IS NOT SUFFICIENT                               DERRM  5
C ANALYSIS - A F KONAR / J K MAHESH - THE MONEYWELL INC         DERRM  6
C DATE WRITTEN - 1975                                           DERRM  7
C                                                                    DERRM  8
C ARGUMENTS LIST                                                DERRM  9
C   M1      INPUT      ACTUAL DIMENSION FOR SCRATCH ARRAY S1    DERRM 10
C   M2      INPUT      ACTUAL DIMENSION FOR SCRATCH ARRAY S2    DERRM 11
C   M3      INPUT      ACTUAL DIMENSION FOR SCRATCH ARRAY S3    DERRM 12
C   M4      INPUT      ACTUAL DIMENSION FOR SCRATCH ARRAY S4    DERRM 13
C   MS1     INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S1   DERRM 14
C   MS2     INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S2   DERRM 15
C   MS3     INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S3   DERRM 16
C   MS4     INPUT      MAXIMUM DIMENSION FOR SCRATCH ARRAY S4   DERRM 17
C   N1      INPUT      PRIMARY OVERLAY NO                         DERRM 18
C   N2      INPUT      SECONDARY OVERLAY NO                       DERRM 19
C   A1      INPUT      NAME OF THE SUBROUTINE                     DERRM 20
C   A2      INPUT      NAME OF THE SUBROUTINE (CONTINUED)        DERRM 21
C   IW      INPUT      FILE NO FOR LINE PRINTER                  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,/,/,)X,28HDIMENSION ERROR IN OVERLAY (,11,1H,,11,1H), DERRM 28
      113HIN SUBROUTINE,2X,2A4)                                  DERRM 29
C   DO 260 I=1,4                                                  DERRM 30
C   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

# Contrails

```

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 MONEYWELL 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(1H1,/,/,1X,2PHDIMENSION ERROR IN OVERLAY (.I1,1H,.,I1,1H), DERRMS29
      113MIN SUBROUTINE,2X,244) DERRMS30
C   DO 26, I=1,4 DERRMS31
C   WRITE(IW,250)A(I),MS(I),M(I) DERRMS32
250 FORMAT(//,1X,10HDIMENSION ,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

# Contrails

```

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 PRINTER                  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(1,4,SHIF,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(2,4,SHIF,4HT ,5,0,IW)              SHIFT 37
C   RETURN                                                         SHIFT 38
C   END                                                            SHIFT 39

```

Figure 80. Subroutine SHIFT Program Listing



# Contrails

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	ARGUMENTS LIST	TDINVR 5
C	ISOL	TDINVR 6
C		TDINVR 7
C		TDINVR 8
C		TDINVR 9
C	IDSOL	TDINVR10
C		TDINVR11
C		TDINVR12
C		TDINVR13
C	NR	TDINVR14
C	NC	TDINVR15
C	A	TDINVR16
C		TDINVR17
C	MRA	TDINVR18
C	KWA	TDINVR19
C	DET	TDINVR20
C		TDINVR21
	DIMENSION A(1),KWA(1)	TDINVR22
	IR=NR	TDINVR23
	ISOL =1	TDINVR24
	IDSOL=1	TDINVR25
	IF (NR) 61,61,11	TDINVR26
11	IF (IR-MRA) 12,12,61	TDINVR27
12	IC=IAHS(NC)	TDINVR28
	IF (IC-IR) 13,14,14	TDINVR29
13	IC=IR	TDINVR30
14	IRMP=1	TDINVR31
	JRMP=MRA	TDINVR32
	KRMP=JRMP+IRMP	TDINVR33
	NES=IP*JBMP	TDINVR34
	NET=IC*JBMP	TDINVR35
	IF (NC) 15,61,16	TDINVR36
15	MDIV=IRMP+1	TDINVR37
	IRIC=IR-IC	TDINVR38
	GO TO 17	TDINVR39
16	MDIV=1	TDINVR40
17	MAD =MDIV	TDINVR41
	MSER=1	TDINVR42
	KSER=IR	TDINVR43
	MZ =1	TDINVR44
	DET=1.0	TDINVR45
18	PIV=0.0	TDINVR46
	I=MSER	TDINVR47
19	IF (I<KSER) 20,20,23	TDINVR48
20	IF (ABS(A(I))-PIV) 22,22,21	TDINVR49
21	PIV=ABS(A(I))	TDINVR50
	IP=I	TDINVR51
22	I=I+IRMP	TDINVR52
	GO TO 19	TDINVR53
23	IF (PIV) 24,62,24	TDINVR54
24	IF (NC) 26,25,25	TDINVR55
25	I=IP-((IP-1)/JRMP)*JRMP	TDINVR56
	J=MSER-((MSER-1)/JRMP)*JBMP	TDINVR57
	JJ=MSER/KRMP+1	TDINVR58
	II=JJ+(IP-MSER)	TDINVR59
	KWA(J)=II	TDINVR60
	GO TO 27	TDINVR61
26	I=IP	TDINVR62
	J=MSER	TDINVR63
27	IF (IP-MSER) 61,31,28	TDINVR64

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+JBMP	TDINVR69
J=J+JBMP	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+JBMP	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+JBMP	TDINVR91
I=I+JBMP	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)/JBMP)*JBMP+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)/JBMP)*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-IRMP	TDINV123
L=L-IRMP	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)

# Contracts

```
ISOL=2  
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)

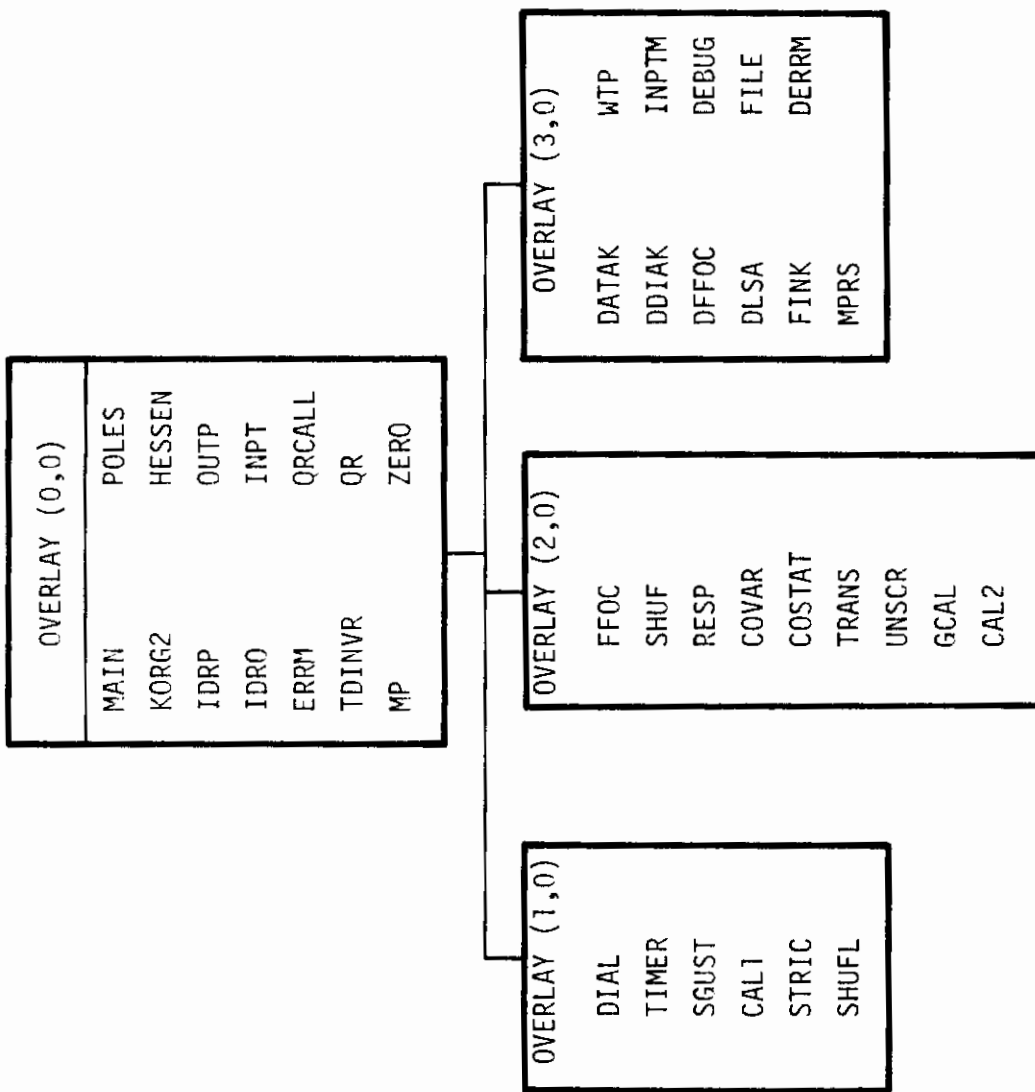


Figure 82. Overlay Structure and Subroutines in KONPACT-2

# Contracts

```
OVERLAY (KONZ=0.0) MAIN 2
PROGRAM MAIN (MINPUT, INPUT, TAPE7=MINPUT, TAPE4=INPUT, MAIN 3
IQDATA, 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 SUBPROGRAMS CALLED MAIN 10
C KORG2 MAIN 11
C LABELLED COMMON LIST MAIN 12
C NXM MAXIMUM NUMBER OF STATES MAIN 13
C NRM MAXIMUM NUMBER OF OUTPUTS MAIN 14
C NUM MAXIMUM NUMBER OF INPUTS MAIN 15
C CODE PROGRAM CODE WORD (DIAK,FFOC,LSA) MAIN 16
C MS1 MAXIMUM DIMENSION FOR SCRATCH ARRAY S1 MAIN 17
C MS2 MAXIMUM DIMENSION FOR SCRATCH ARRAY S2 MAIN 18
C MS3 MAXIMUM DIMENSION FOR SCRATCH ARRAY S3 MAIN 19
C MS4 MAXIMUM DIMENSION FOR SCRATCH ARRAY S4 MAIN 20
C COMMON /INF/ NXM,NRM,NUM, CODE,MS1,MS2,MS3,MS4 MAIN 21
C MAIN 22
C MAIN 23
C MAXIMUM SYSTEM DIMENSIONS MAIN 24
C NXM=50 $ NRM=70 $ NUM=20 MAIN 25
C MAIN 26
C MAXIMUM SCRATCH ARRAY DIMENSIONS MAIN 27
C MS1=00500 $ MS2=17000 $ MS3=00001 $ MS4=00001 MAIN 28
C MAIN 29
C *** NOTE *** SCRATCH ARRAY DIMENSIONS IN PROGRAM DATAK MAIN 30
C SHOULD BE CHANGED MAIN 31
C MAIN 32
C CALL KONPACT ORGANIZING SUBROUTINE MAIN 33
C MAIN 34
C CALL KORG2 MAIN 35
C STOP MAIN 36
C END MAIN 37
C MAIN 38
C MAIN 39
C MAIN 40
C MAIN 41
```

Figure 83. Program MAIN Program Listing

# Contrails

```

OVERLAY(KONZ,1,0)                                DIAK  2
PROGRAM DIAK                                     DIAK  3
C DOURLY-ITERATIVE ALGORITHM FOR SOLVING ALGEBRAIC RICCATI EQUATION DIAK  4
C THIS PROGRAM COMPUTES QUADRATIC CONTROLLERS AND/OR COMPUTES COVARIANDIAK  5
C TIME RESPONSES FOR SYSTEMS MODELED AS          DIAK  6
C                                                  DIAK  7
C          XDOT = F*X + G1*U + G2*ETA            DIAK  8
C AND                                             DIAK  9
C          R = H*X + D*U                          DIAK 10
C WITH                                           DIAK 11
C          J = E(R#*O#R)                          DIAK 12
C                                                  DIAK 13
C DIMENSION F(40,40),G1(40,6),G2(40,2),A(40,40),AN(40,40),E(40,40) DIAK 14
C DIMENSION Q(40,40), WR(40,40),EP(40,40),P(40,40),H(40,40),D(40,6) DIAK 15
C DIMENSION AK(6,40),PI(40,40),DQD(6,6),KWA(40),W(6,40),WI(6,40) DIAK 16
C DIMENSION QO(40,40),RR(90),AM(40,40),BK(6,40),X(40),DX(40),DX1(40) DIAK 17
C DIMENSION XI(40,2),XLDXL(40,2),GM(40,2),GS(40,2),R(8000),IPLR(80) DIAK 18
C DIMENSION ITITL(80),YMAX(80),YMIN(80),CL(2,1),SCAL(80),NEWY(80) DIAK 19
C DIMENSION NORD(40),OR(40,40),IUNIT(80)         DIAK 20
C COMMON A,E,Q,AN,WR,OR,EP,P,PI                 DIAK 21
C EQUIVALENCE (F(1),P(1)),(H(1),PI(1)),(AM(1),EP(1)) DIAK 22
C EQUIVALENCE (R(1),E(1))                       DIAK 23
C DIMENSIONS OF THE ABOVE ARRAYS ARE DEFINED BELOW. CHANGE BOTH SIMULT DIAK 24
C SEE DOCUMENTATION FOR DEFINITIONS OF ARRAY DIMENSIONS DIAK 25
C MX>NX                                          DIAK 26
C MR>NR                                          DIAK 27
C MU>NU                                          DIAK 28
C MN>NV                                          DIAK 29
C MXR>NOR                                       DIAK 30
C MPOIN>(NOP + 1)*(T/ST)                        DIAK 31
C MR=40                                          DIAK 32
C MXR=80                                         DIAK 33
C MX=40                                          DIAK 34
C MU=6                                           DIAK 35
C MN=2                                           DIAK 36
C MPOIN=8000                                     DIAK 37
C CONVERGENCE TEST FACTOR                       DIAK 38
C EE=0.001                                       DIAK 39
C ITERATION COUNTER                             DIAK 40
C RUN COUNTER                                   DIAK 41
C IRUN=1                                         DIAK 42
C READ AND PRINT ID                             DIAK 43
C READ(5,1274) IDATE,NAME1,NAME2                DIAK 44
C 1274 FORMAT(3A10)                              DIAK 45
C WRITE(9,1275) IDATE,NAME1,NAME2               DIAK 46
C 1275 FORMAT(1H1,7X,13HTODAY#S DATE ,A10,5X,16HIDENTIFICATION ,2A10//) DIAK 47
C READ NUMBER OF VARIABLES BEING PLOTTED        DIAK 48
C READ(5,28) NOP                                 DIAK 49
C 28 FORMAT(40I2)                               DIAK 50
C IF NOP = 0, SKIP TO STATEMENT 70              DIAK 51
C IF(NOP.EQ.0) GO TO 70                         DIAK 52
C READ PLOTTING PARAMETERS - THEY ARE FIXED FOR ALL RUNS DIAK 53
C IPLR = ARRAY OF PLOTTING VARIABLE NOS. - READ IN ORDER DIAK 54
C ITITL = CORRESPONDING ARRAY OF LABELS        DIAK 55
C IUNIT = CORRESPONDING ARRAY OF UNIT LABELS   DIAK 56
C YMAX,YMIN = CORRESPONDING ARRAYS OF DESIGNATED MAX AND MIN VALUES DIAK 57
C SCAL = CORRESPONDING ARRAY OF SCALE FACTORS  DIAK 58
C READ(5,1272) ((IPLR(I),ITITL(I),IUNIT(I),YMIN(I),YMAX(I),SCAL (I)) DIAK 59
C 1,I=1,NOP)                                     DIAK 60
C 1272 FORMAT(12,2X,A10,2X,A10,2X,G11.3,2X,G11.3,2X,G11.3) DIAK 61
C DEFINE PLOTTING SCALES - FIXED FOR ALL RUNS  DIAK 62
C IF YMIN AND YMAX ARE 0, USE COMPUTED MAX AND MIN (NEWY=1) DIAK 63
C IF SCAL = 0, USE SCALE FACTOR OF 1           DIAK 64

```

Figure 84. Program DIAK Program Listing

# Contrails

```

DO 1 I=1,NOP                                DIAK 65
NEWY(I)=0                                    DIAK 66
IF(YMIN(I).EQ.0..AND.YMAX(I).EQ.0.) NEWY(I)=1 DIAK 67
IF(SCAL(I).EQ.0.) SCAL(I)=1.                DIAK 68
1 CONTINUE                                  DIAK 69
C READ AND PRINT PLOTTING TIME PARAMETERS - FIXED FOR ALL RUNS DIAK 70
C T = TOTAL PLOTTING TIME                    DIAK 71
C DT = SAMPLING INTERVAL                     DIAK 72
C ST = PLOTTING SAMPLING INTERVAL           DIAK 73
C T1 = FIRST DELAY IN GUST PROFILE          DIAK 74
C T2 = SECOND DELAY IN GUST PROFILE         DIAK 75
READ(5,1278) T,DT,ST,T1,T2                 DIAK 76
1278 FORMAT(5G12.4)                          DIAK 77
C PRINT PLOTTING PARAMETERS                  DIAK 78
WRITE(9,1279) T,DT,ST,T1,T2                DIAK 79
1279 FORMAT(1H0/7X,31H TIME RESPONSES PLOTTING TIME =,G12.4/22X,18H SAMDIK 80
1PLE INTERVAL =,G12.4/22X,27H PLOTTING SAMPLE INTERVAL =,G12.4/ DIAK 81
22X,19H FIRST DELAY TIME =,G12.4/22X,20H SECONDD DELAY TIME =,G12.4DIAK 82
3//)                                         DIAK 83
WRITE(9,1285) ((IPLR(I),ITITL(I),JUNIT(I),YMIN(I),YMAX(I),SCAL(I)DIAK 84
I),I=1,NOP)                                DIAK 85
1285 FORMAT(7X,18HPLOTTING VARIABLES//2X,86HRESPONSE NUMBER RESPONSE VDIAK 86
IARIABLE RESPONSE UNITS MIN SCALE MAX SCALE SCALE FACTOR//12X, 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 DIAK 89
70 READ(5,28) IMAX,ITER,ITERQ              DIAK 90
WRITE(9,4002)IMAX,ITER,ITERQ               DIAK 91
4002 FORMAT(///7X,37H MAX NUMBER OF INNER-LOOP ITERATIONS I3,37H MAX NDIAK 92
UMBER 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             CIAK 95
C INPD=1 COMPLETELY NEW DATA                DIAK 96
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 GADIK 98
C INPD=4 CHANGE SELECTED DATA              DIAK 99
C INPD=5 CHANGE SELECTED DATA IN MEASUREMENT MATRIX, QUADRATIC WEIGHTSDIAK 100
C OPTION FOR NEW GAINS                       DIAK 101
C INPK=1 NEW INPUT GAINS                     DIAK 102
C INPK=2 NEW STARTING ROUTINE GAINS          DIAK 103
C INPK=3 USE GAINS IN STORAGE                DIAK 104
C INPK=4 USE INPUT GAINS IN STORAGE          DIAK 105
C NCONT=0 DON#T COMPUTE OPTIMAL GAINS - USE INPUT GAINS AND DATA IN CODIAK 106
C AND TIME RESPONSE ANALYSIS ONLY           DIAK 107
C NCONT=1 COMPUTE OPTIMAL GAINS              DIAK 108
C NCONT=2 DO AUTOMATIC SELECTION OF Q ON CONTROL RATES DIAK 109
C SEE SUBROUTINE TIMER FOR PLOTTING OPTIONS USING NPLOT, NPRIN, NSTEP, DIAK 110
C NOCOV=1 NO COVARIANCE ANALYSIS            DIAK 111
C NOCOV=2 COVARIANCE ANALYSIS               DIAK 112
C NOCOV=3 SKIP CORRELATION ANALYSIS         DIAK 113
C READ AND PRINT PROGRAM OPTIONS            DIAK 114
READ(5,28) NOCOV,NSTEP,NRAND,NPRIN ,NPLOT DIAK 115
READ(5,28) INPK                             DIAK 116
INPD=1                                       DIAK 117
READ(5,28) NCONT                             DIAK 118
WRITE(9,37) INPD,INPK,NCONT,NCOV,NSTEP,NRAND,NPRIN ,NPLOT DIAK 119
37 FORMAT(1H1/7X,23HNEW PROBLEM WITH INPD =,I3,2X,6HINPK =,I3,2X, DIAK 120
17HNCONT =,I3/7X,7HNCOV =,I3/7X,7HNSTEP =,I3,2X,7HNRAND =,I3/7X, DIAK 121
27HNPRIN =,I3,2X,7HNPLOT =,I3//)           DIAK 122
1210 CONTINUE                                DIAK 123
C READ FLIGHT CONDITION ID                  DIAK 124
READ(5,1270)IFLT                            DIAK 125
1270 FORMAT(A10)                             DIAK 126
C PRINT FLIGHT CONDITION ID AND RUN NO.     DIAK 127
WRITE(9,1271) IFLT, IRUN                     DIAK 128
1271 FORMAT(1H1/7X,18H FLIGHT CONDITION A10,5X,3HRUN,I3) DIAK 129
C READ AND PRINT SYSTEM PARAMFTERS         DIAK 130

```

Figure 84. Program DIAK Program Listing (Continued)

# Contrails

```

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 NUMBER 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//) DIAK 148
C      NC IS THE NUMBER OF UPPER TRIANGULAR ELEMENTS IN P DIAK 149
      NC=(NX*(NX+1))/2 DIAK 150
C      DIAK 151
C ZERO ARRAYS DIAK 152
C      DIAK 153
C      RIGHT HAND PARAMETERS DEFINED BELOW DIAK 154
      DO 8020 I=1,MX DIAK 155
      DO 8013 J=1,MX DIAK 156
      F (I,J)=0. DIAK 157
      A (I,J)=0. DIAK 158
      AN (I,J)=0. DIAK 159
      E (I,J)=0. DIAK 160
      EP (I,J)=0. DIAK 161
      PJ (I,J)=0. DIAK 162
      AM(I,1)=0. DIAK 163
8013  CONTINUE DIAK 164
      DO 8014 J=1,NU DIAK 165
8014  G1(I,J)=0. DIAK 166
      DO 8015 J=1,NN DIAK 167
      XI(I,J)=0. DIAK 168
      CL(J,1)=0. DIAK 169
8015  G2(I,J)=0. DIAK 170
      DO 8016 J=1,2 DIAK 171
8016  XLDXL(I,J)=0. DIAK 172
8020  CONTINUE DIAK 173
      DO 3060 I=1,NR DIAK 174
      DO 3061 J=1,NR DIAK 175
3061  QO(I,J)=0. DIAK 176
      DO 3062 J=1,MX DIAK 177
3062  H(I,J)=0. DIAK 178
      DO 3060 J=1,NU DIAK 179
3060  D(I,J)=0. DIAK 180
C      READ DATA FOR THIS RUN DIAK 181
1240  IF (INPD.GT.1) GO TO 53 DIAK 182
C      IF INPD = 1 (NEW DATA), READ ORDERING OF STATES DIAK 183
C      NORD = ARRAY OF THE ORDER OF STATES DIAK 184
      READ(5,28) (NORD(I),I=1,NX) DIAK 185
      WRITE(9,67) (NORD(I),I=1,NX) DIAK 186
67    FORMAT(//7X,22H STATES ARE ORDERED AS//(7X,20I4)//) DIAK 187
53    CONTINUE DIAK 188
C      READ CHANGES IN F,G1,G2 (ROW, COLUMN, ELEMENT VALUE) DIAK 189
C      F = STABILITY MATRIX (OPEN LOOP) DIAK 190
C      G1 = CONTROL INPUT MATRIX DIAK 191
C      G2 = DISTURBANCE INPUT MATRIX DIAK 192
C      WHERE DIAK 193
      XDOT = F*X + G1*U + G2*ETA DIAK 194
C      IF INPD>1 (CHANGES TO EXISTING DATA), ROW AND COLUMNS CORRESPOND DIAK 195
C      RE-ORDERED STATES DIAK 196

```

Figure 84. Program DIAK Program Listing (Continued)



# Contrails

```

        CALL INPT (F,MX,MX)
        CALL INPT(G1,MX,MU)
        CALL INPT (G2,MX,MN)
        IF(INPD.GT.1) GO TO 54
C   IF DATA IS NEW, RE-ORDER THE STATES (CALL SHUFL)
        CALL SHUFL(F,MX,MX,NX,NX,1,1,NORD,0,MX)
        CALL SHUFL(G1,MX,MU,NX,NU,1,0,NORD,0,MX)
        CALL SHUFL(G2,MX,MN,NX,NN,1,0,NORD,0,MX)
54  CONTINUE
C   PRINT F,G1,G2
        WRITE(9,20)
        CALL MP(MX,MX,NX,NX,F)
        WRITE(9,21)
        CALL MP(MX,MU,NX,NU,G1)
        WRITE(9,22)
22  FORMAT(1H1/7X,10H G2 MATRIX//)
        CALL MP(MX,MN,NX,NN,G2)
20  FORMAT(1H1/7X,10H F MATRIX//)
21  FORMAT(1H1/7X,10H G1 MATRIX//)
C   READ CHANGES IN XI AND XLDXL
C   XI = INITIAL STATE VALUES IN SIMULATION
C   XLDXL = STATE AND STATE RATE LIMITS
        CALL INPT(XI,MX,MN)
        CALL INPT(XLDXL,MX,2)
        IF(INPD.GT.1) GO TO 55
C   IF DATA IS NEW, RE-ORDER THE STATES (CALL SHUFL)
        CALL SHUFL(XI,MX,MN,NX,NN,1,0,NORD,0,MX)
        CALL SHUFL(XLDXL,MX,2,NX,2,1,0,NORD,0,MX)
55  CONTINUE
C   READ CL = STEP GUST AND COMMAND INPUT LEVELS
        CALL INPT(CL,MN,1)
C   PRINT XI, XLDXL, CL
        WRITE(9,1276)
1276 FORMAT(1H1/7X,24HINITIAL CONDITION MATRIX//)
        CALL MP(MX,MN,NX,NN,XI)
        WRITE(9,1277)
1277 FORMAT(1H1/7X,71HSTATE LIMIT - RATE LIMIT MATRIX//)
        CALL MP(MX,2,NX,2,XLDXL)
        WRITE(9,1273)
1273 FORMAT(1H1/7X,20HCOMMAND LEVEL MATRIX//)
        CALL MP(MN,1,NN,1,CL)
C   READ IN CHANGES IN H AND D
C   H = STATE-RESPONSE OUTPUT MATRIX
C   D = CONTROL-RESPONSE OUTPUT MATRIX
C
C           R = H*X + D*U
C   WHERE
C
73  CALL INPT (H,MR,MX)
        CALL INPT(D,MR,MU)
        IF(INPD.GT.1) GO TO 1250
C   IF DATA IS NEW, RE-ORDER STATES (CALL SHUFL)
        CALL SHUFL(H,MR,MX,NR,NX,0,1,NORD,0,MX)
C   READ CHANGES IN M (AM)
3   AM = MEASUREMENT MATRIX - USED FOR RESPONSE ANALYSIS ONLY
C   WHERE
C           Y = M*X
C
1250 CALL INPT (AM,MX,MX)
        IF(INPD.GT.1) GO TO 56
C   IF DATA IS NEW, RE-ORDER STATES (CALL SHUFL)
        CALL SHUFL(AM,MX,MX,NX,NX,0,1,NORD,0,MX)
56  CONTINUE
C   PRINT H, D, AM
        WRITE(9,23)
        CALL MP(MR,MX,NR,NX,H)

```

DIAK 197  
 DIAK 198  
 DIAK 199  
 DIAK 200  
 DIAK 201  
 DIAK 202  
 DIAK 203  
 DIAK 204  
 DIAK 205  
 DIAK 206  
 DIAK 207  
 DIAK 208  
 DIAK 209  
 DIAK 210  
 DIAK 211  
 DIAK 212  
 DIAK 213  
 DIAK 214  
 DIAK 215  
 DIAK 216  
 DIAK 217  
 DIAK 218  
 DIAK 219  
 DIAK 220  
 DIAK 221  
 DIAK 222  
 DIAK 223  
 DIAK 224  
 DIAK 225  
 DIAK 226  
 DIAK 227  
 DIAK 228  
 DIAK 229  
 DIAK 230  
 DIAK 231  
 DIAK 232  
 DIAK 233  
 DIAK 234  
 DIAK 235  
 DIAK 236  
 DIAK 237  
 DIAK 238  
 DIAK 239  
 DIAK 240  
 DIAK 241  
 DIAK 242  
 DIAK 243  
 DIAK 244  
 DIAK 245  
 DIAK 246  
 DIAK 247  
 DIAK 248  
 DIAK 249  
 DIAK 250  
 DIAK 251  
 DIAK 252  
 DIAK 253  
 DIAK 254  
 DIAK 255  
 DIAK 256  
 DIAK 257  
 DIAK 258  
 DIAK 259  
 DIAK 260  
 DIAK 261  
 DIAK 262

Figure 84. Program DIAK Program Listing (Continued)

# Contrails

```

WRITE(9,24)
CALL MP(MR,MU,NR,NU,D)
WRITE(9,29)
29 FORMAT(1H1/7X,10H M MATRIX//)
CALL MP(MX,MX,NX,NX,AM)
23 FORMAT(1H1/7X,10H H MATRIX//)
24 FORMAT(1H1/7X,10H D MATRIX//)
C CHECK GAINS INPUT OPTION
1230 GO TO (3001,3002,3010,3011),INPK
C NEW INPUT GAINS
C BK = INPUT GAINS MATRIX
C WHERE
C U = BK*X (WHEN COMPUTING OPTIMAL GAINS)
C OR
C U = BK*Y = BK*AM*X (WHEN COMPUTING RESPONSES ONLY)
C
C ZERO AND READ BK
3001 DO 3003 I=1,NU
DO 3003 J=1,MX
3003 BK(I,J)=0.
CALL INPT (BK,MU,MX)
IF(INCONT.EQ.0) GO TO 57
C IF NCONT>0, RE-ORDER STATES (BECAUSE U = BK*X)
CALL SHUFL(BK,MU,MX,NU,NX,0,1,NORD,Q,MX)
57 CONTINUE
C PRINT BK
WRITE(9,30)
30 FORMAT(1H1/7X,19H INPUT GAINS MATRIX//)
CALL MP(MU,MX,NU,NX,BK)
C SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS
GO TO 1220
C USE STARTING ROUTINE (STRIC) TO COMPUTE STARTING GAINS - AS A LAST
C BK = -G1*(W(T)): (A: MEANS INVERSE OF MATRIX A)
C WHERE
C W(T) = INTEGRAL(0,BT)OF(EXP(F*T)*G1*G1#*EXP(F#*T))DT
C FOR AN ARBITRARY TIME BT
3002 CALL STRIC(F,G1,A,AN,E,Q,MF,NU,MX,MU)
CALL TDINVR(ISOL,IDSOL,NF,NF,AN,MX,KWA,DET)
IF((ISOL+IDSOL)-2) 3004;3004,3005
C (W(T)): IS NO GOOD - GO TO NEXT RUN - BUT FIRST, READ REMAINING DATA
C THIS RUN AND CHECK TO SEE IF THE NEXT RUN IS SOLVABLE - THE START
C GAINS MAY NOT BE GOOD - IF SO, STOP
3005 WRITE(9,3006)
3006 FORMAT(1H1/7X,31H INVERSE OF W(T) DOES NOT EXIST/7X,18H CHECK NEXT
1PROBLEM//)
CALL INPT (QQ,MR,MR)
READ(5,1215) IDUM
IF(IDUM.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
WRITE(9,3008)
3008 FORMAT(/7X,51HNEW PROBLEM NOT SOLVABLE WITHOUT NEW STARTING GAINS/
1/)
STOP 11
C DEFINE RK
C AN = (WCT)):
C
3004 DO 3009 I=1,NU
DO 3009 J=1,NF
BK(I,J)=0.
DO 3009 K=1,NF
3009 BK(I,J)=BK(I,J)-G1(K,I)*AN(K,J)
C PRINT BK
WRITE(9,31)
31 FORMAT(1H1/7X,22H STARTING GAINS MATRIX//)

```

```

DIAK 263
DIAK 264
DIAK 265
DIAK 266
DIAK 267
DIAK 268
DIAK 269
DIAK 270
DIAK 271
DIAK 272
DIAK 273
DIAK 274
DIAK 275
DIAK 276
DIAK 277
DIAK 278
DIAK 279
DIAK 280
DIAK 281
DIAK 282
DIAK 283
DIAK 284
DIAK 285
DIAK 286
DIAK 287
DIAK 288
DIAK 289
DIAK 290
DIAK 291
DIAK 292
DIAK 293
DIAK 294
DIAK 295
DIAK 296
DIAK 297
DIAK 298
DIAK 299
DIAK 300
DIAK 301
DIAK 302
DIAK 303
DIAK 304
DIAK 305
DIAK 306
DIAK 307
DIAK 308
DIAK 309
DIAK 310
DIAK 311
DIAK 312
DIAK 313
DIAK 314
DIAK 315
DIAK 316
DIAK 317
DIAK 318
DIAK 319
DIAK 320
DIAK 321
DIAK 322
DIAK 323
DIAK 324
DIAK 325
DIAK 326
DIAK 327
DIAK 328

```

Figure 84. Program DIAK Program Listing (Continued)

# Contrails

<pre> CALL MP(MU, MX, NU, NF, BK) C SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS GO TO 1220 C USE USE LAST COMPUTED GAINS IN STORAGE FOR STARTING GAINS C DEFINE BK = AK 3010 WRITE(9,33) 33 FORMAT(1H1/7X,28H USE GAINS MATRIX IN STORAGE//) C SKIP TO STATEMENT 1220 TO READ QUADRATIC WEIGHTS GO TO 1220 C USE INPUT GAINS IN STORAGE - RK = BK 3011 WRITE(9,34) 34 FORMAT(1H1/7X,34H USE INPUT GAINS MATRIX IN STORAGE//) C READ CHANGES IN QUADRATIC WEIGHTS FOR PERFORMANCE INDEX C C J = E(R**Q*R) C WHERE Q IS THE MATRIX OF QUADRATIC WEIGHTS C QQ = Q 1220 CONTINUE CALL INPT(QQ, MR, MR) NQ=1 81 CONTINUE C PRINT QQ WRITE(9,36) 36 FORMAT(1H1/7X,27H QUADRATIC WEIGHTING MATRIX//) CALL MP(MR, MR, NR, NR, QQ) C IF NCONT = 0 (NO OPTIMAL CONTROL COMPUTATIONS), SKIP TO STATEMENT 890 C RESPONSE COMPUTATIONS IF(NCONT.EQ.0) GO TO 893 C CALCULATE A+E*Q FOR PICCATI EQUATION 0 = PA + A*P + Q - PEP C W = D**Q DO 4 I=1,NU DO 4 J=1,NR W(I,J)=0. DO 4 K=1,NR 4 W(I,J)=W(I,J)+D(K,I)*QQ(K,J) C DQD = D**Q*D DO 5 I=1,NU DO 5 J=1,NU DQD(I,J)=0. DO 5 K=1,NR 5 DQD(I,J)=DQD(I,J)+W(I,K)*D(K,J) C INVERT DQD - DQD = (D**Q*D): IF(NU-1)302,302,301 302 DQD(1,1)=1./DQD(1,1) GOTO 303 301 CONTINUE CALL TDINVR(ISOL,IDSOL,NU,NU,DQD,MU,KWA,DET) IF((ISOL+IDSOL)-2)6,6,7 C (D**Q*D): DOES NOT EXIST - GO TO NEXT RUN 7 WRITE(9,35) 35 FORMAT(1H1/7X,30H INVERSE OF DQD DOES NOT EXIST//7X,19H CHECK NEXT 1 PROBLEM//) GO TO 1200 6 CONTINUE 303 CONTINUE C W1 = D**Q*H DO 8 I=1,NU DO 8 J=1,NX W1(I,J)=0. DO 8 K=1,NR 8 W1(I,J)=W1(I,J)+W(I,K)*H(K,J) C W = (D**Q*D):*D**Q*H C STORE W FOR OPTIMAL CONTROL COMPUTATION DO 9 I=1,NU DO 9 J=1,NX W(I,J)=0. </pre>	<pre> DIAK 329 DIAK 330 DIAK 331 DIAK 332 DIAK 333 DIAK 334 DIAK 335 DIAK 336 DIAK 337 DIAK 338 DIAK 339 DIAK 340 DIAK 341 DIAK 342 DIAK 343 DIAK 344 DIAK 345 DIAK 346 DIAK 347 DIAK 348 DIAK 349 DIAK 350 DIAK 351 DIAK 352 DIAK 353 DIAK 354 DIAK 355 DIAK 356 DIAK 357 DIAK 358 DIAK 359 DIAK 360 DIAK 361 DIAK 362 DIAK 363 DIAK 364 DIAK 365 DIAK 366 DIAK 367 DIAK 368 DIAK 369 DIAK 370 DIAK 371 DIAK 372 DIAK 373 DIAK 374 DIAK 375 DIAK 376 DIAK 377 DIAK 378 DIAK 379 DIAK 380 DIAK 381 DIAK 382 DIAK 383 DIAK 384 DIAK 385 DIAK 386 DIAK 387 DIAK 388 DIAK 389 DIAK 390 DIAK 391 DIAK 392 DIAK 393 DIAK 394 </pre>
--	--

Figure 84. Program DIAK Program Listing (Continued)

# Contrails

```

DO 9 K=1,NU
9 W(I,J)=W(I,J)+DQD(I,K)*W(K,J)
C AN = F - G1*(D**Q*D):*D**Q*H
C AN = A OF EQUATION 0 = A#P + PA + Q - PEP
DO 10 I=1,NX
DO 10 J=1,NX
AN(I,J)=F(I,J)
DO 10 K=1,NU
10 AN(I,J)=AN(I,J)-G1(I,K)*W(K,J)
C Q = -H**Q*D*(D**Q*D):*D**Q*H
DO 12 I=1,NX
DO 12 J=1,NX
Q(I,J)=0.
DO 12 K=1,NU
12 Q(I,J)=Q(I,J)-W1(K,I)*W(K,J)
C E = Q*H
DO 13 I=1,NR
DO 13 J=1,NX
E(I,J)=0.
DO 13 K=1,NR
13 E(I,J)=E(I,J)+QD(I,K)*H(K,J)
C Q = H**Q*H - H**Q*D*(D**Q*D):*D**Q*H
C Q = Q OF EQUATION 0 = A#P + PA + Q - PEP
DO 14 I=1,NX
DO 14 J=1,NX
DO 15 K=1,NR
15 Q(I,J)=Q(I,J)+H(K,I)*F(K,J)
14 Q(J,I)=Q(I,J)
C W1 = (D**Q*D):*G1#
DO 16 I=1,NU
DO 16 J=1,NX
W1(I,J)=0
DO 16 K=1,NU
16 W1(I,J)=W1(I,J)+DQD(I,K)*G1(J,K)
C E = G1*(D**Q*D):*G1#
C E = E OF EQUATION 0 = A#P + PA + Q - PEP
DO 17 I=1,NX
DO 17 J=1,NX
E(I,J)=0.
DO 18 K=1,NU
18 E(I,J)=E(I,J)+G1(I,K)*W1(K,J)
17 E(J,I)=E(I,J)
C PRINT AN,E,Q
WRITE(9,32)
32 FORMAT(1H1/7X,36HSTARTING MATRICES FOR PA+A#P+Q-PEP=0//)
WRITE(9,25)
CALL *P(MX,MX,NX,NX,AN)
WRITE(9,26)
CALL *P(MX,MX,NX,NX,E)
WRITE(9,27)
CALL *P(MX,MX,NX,NX,Q)
25 FORMAT(//7X,10H A MATRIX//)
26 FORMAT(1H1/7X,10H E MATRIX//)
27 FORMAT(1H1/7X,10H Q MATRIX//)
C DUMP F,H, AND AM ON DISC TO CONSERVE STORAGE
C P, P1, AND EP USE STORAGE EQUIVALENT TO THESE MATRICES
REWIND 2
WRITE(2) F
WRITE(2) H,AM
ITERC=0
C CHECK GAINS INPUT OPTION
GO TO (3000,3000,?(50,3000),INPK)
C FOR ALL OPTIONS EXCEPT INPK = 3 (USE AK IN STORAGE), AK = BK
3000 DO 7010 I=1,NU
DO 7010 J=1,NX
7010 AK(I,J)=BK(I,J)

```

DIAK 395  
 DIAK 396  
 DIAK 397  
 DIAK 398  
 DIAK 399  
 DIAK 400  
 DIAK 401  
 DIAK 402  
 DIAK 403  
 DIAK 404  
 DIAK 405  
 DIAK 406  
 DIAK 407  
 DIAK 408  
 DIAK 409  
 DIAK 410  
 DIAK 411  
 DIAK 412  
 DIAK 413  
 DIAK 414  
 DIAK 415  
 DIAK 416  
 DIAK 417  
 DIAK 418  
 DIAK 419  
 DIAK 420  
 DIAK 421  
 DIAK 422  
 DIAK 423  
 DIAK 424  
 DIAK 425  
 DIAK 426  
 DIAK 427  
 DIAK 428  
 DIAK 429  
 DIAK 430  
 DIAK 431  
 DIAK 432  
 DIAK 433  
 DIAK 434  
 DIAK 435  
 DIAK 436  
 DIAK 437  
 DIAK 438  
 DIAK 439  
 DIAK 440  
 DIAK 441  
 DIAK 442  
 DIAK 443  
 DIAK 444  
 DIAK 445  
 DIAK 446  
 DIAK 447  
 DIAK 448  
 DIAK 449  
 DIAK 450  
 DIAK 451  
 DIAK 452  
 DIAK 453  
 DIAK 454  
 DIAK 455  
 DIAK 456  
 DIAK 457  
 DIAK 458  
 DIAK 459  
 DIAK 460

Figure 84. Program DIAK Program Listing (Continued)

# Contrails

```

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)
      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)+Q(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
      J   P IS WORKING MATRIX HERE - RICCATI MATRIX RETURNS IN EP
      CALL CAL(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 PRODI
      1ORLEM//)
      READ(5,1215) IDUM
      IF(IDUM,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 ITERA
C   THUS INITIALIZE DIFFERENCES AND CONVERGENCE CRITERIA
C   P IS DIFFERENCE AND PI IS THE TOTAL RICCATI MATRIX
C   INITIALLY PI IS ZERO
      DO 100 I=1,NX
      DO 100 J=1,NX
      EP(I,I)=0.

```

Figure 84. Program DIAK Program Listing (Continued)

# Contrails

```

      DO 101 K=1,NX
101  EP(I, I)=EP(I, J)+E(I, K)*P(K, J)
100  A(I, J)=AN(I, J)-EP(I, J)
      DO 102 I=1,NX
      DO 102 J=1,NX
      DO 102 K=1,NX
103  Q(I, J)=Q(I, J)+P(I, K)*EP(K, J)
102  Q(J, I)=Q(I, J)
      FEE=EF
1000 CONTINUE
      DO 2010 I=1,NX
      DO 2010 J=1,NX
2010  PI(I, I)=P(I, J)+PI(I, J)
      CALL SECOND(TT)
      WRITE(9,3055) TT
3055  FORMAT(//7X,6HTIME =,F10.5//)
C    CALL CAL -P IS AGAIN WORKING MATRIX - RICCATI MATRIX RETURNS IN Q
      CALL CALL(A,Q,P,KWA,NX,MX,IMAX,1,IERR,FEE)
      FEE=EF*10.
      CALL SECOND(TT)
      WRITE(9,3055) TT
      IF(IERR.EQ.0) GO TO 874
C    ERROR ENCOUNTERED IN CAL - GO TO NEXT RUN
      WRITE(9,39)
      39 FORMAT(1H1//7X,30H RICCATI SOLUTION IS DIVERGING//7X,19H CHECK NEXT
1    PROBLEM//)
      READ(5,1215) IDUM
      IF(IDUM.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 = Q
      874 DO 877 I=1,NX
      DO 877 J=1,NX
      877  P(I, J)=Q(I, J)
      IF(ITERC.GT.0) GO TO 3057
C    ON SECOND ITERATION - SOLVE FOR DIFFERENCE P = P - PI
      DO 3058 I=1,NX
      DO 3058 J=1,NX
3058  P(I, J)=P(I, J)-PI(I, J)
3057  CONTINUE
      ITERC=ITERC+1
C    UPDATE A AND Q FOR NEXT ITERATION *WHERE
C    A = AN - E*(P+PI) - (P+PI) IS TOTAL RICCATI MATRIX
C    Q = -P*E*P
C    TO SOLVE FOR THE DIFFERENCE P FROM
C
C    
$$Q = A * P + P * A + Q$$

C
      DO 3050 I=1,NX
      DO 3050 J=1,NX
      EP(I, I)=J.
      A(I, J)=AN(I, J)
      DO 3050 K=1,NX
      EP(I, I)=EP(I, J)+E(I, K)*P(K, J)
3050  A(I, J)=A(I, J)-E(I, K)*(PI(K, J)+P(K, J))
      DO 3052 I=1,NX
      DO 3052 J=1,NX
      Q(I, J)=J.
      DO 3051 K=1,NX
3051  Q(I, J)=Q(I, J)-P(I, K)*EP(K, J)
3052  Q(J, I)=Q(I, J)
C    BEFORE GOING TO THE NEXT ITERATION, CHECK FOR CONVERGENCE

```

DIAK 527  
 DIAK 528  
 DIAK 529  
 DIAK 530  
 DIAK 531  
 DIAK 532  
 DIAK 533  
 DIAK 534  
 DIAK 535  
 DIAK 536  
 DIAK 537  
 DIAK 538  
 DIAK 539  
 DIAK 540  
 DIAK 541  
 DIAK 542  
 DIAK 543  
 DIAK 544  
 DIAK 545  
 DIAK 546  
 DIAK 547  
 DIAK 548  
 DIAK 549  
 DIAK 550  
 DIAK 551  
 DIAK 552  
 DIAK 553  
 DIAK 554  
 DIAK 555  
 DIAK 556  
 DIAK 557  
 DIAK 558  
 DIAK 559  
 DIAK 560  
 DIAK 561  
 DIAK 562  
 DIAK 563  
 DIAK 564  
 DIAK 565  
 DIAK 566  
 DIAK 567  
 DIAK 568  
 DIAK 569  
 DIAK 570  
 DIAK 571  
 DIAK 572  
 DIAK 573  
 DIAK 574  
 DIAK 575  
 DIAK 576  
 DIAK 577  
 DIAK 578  
 DIAK 579  
 DIAK 580  
 DIAK 581  
 DIAK 582  
 DIAK 583  
 DIAK 584  
 DIAK 585  
 DIAK 586  
 DIAK 587  
 DIAK 588  
 DIAK 589  
 DIAK 590  
 DIAK 591  
 DIAK 592

Figure 84. Program DIAK Program Listing (Continued)

# Contrails

```

C      CONVERGENCE IS WHEN THE ABSOLUTE CHANGE IN THE ELEMENTS OF THE RIDIAK 593
C      MATRIX BETWEEN ITERATIONS IS LESS THAN THE ABSOLUTE VALUE OF THE DIAK 594
C      TIMES EE                                                              DIAK 595
C      ONLY CHECK THE UPPER TRIANGULAR ELEMENTS                             DIAK 596
      ICT=0                                                                    DIAK 597
      DO 105 I=1,NX                                                            DIAK 598
      DO 105 J=I,NX                                                            DIAK 599
      API=ABS(P(I,J))                                                         DIAK 600
      IF(API.LT.1.E-20) GO TO 135                                             DIAK 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                                             DIAK 603
C      IF THE ELEMENTS ARE LARGE, CONSIDER THEM AS DIVERGING AND GO TO NEXTDIAK 604
      WRITE(9,39)                                                             DIAK 605
      READ(5,1215) IDUM                                                       DIAK 606
      IF(IDUM.GT.0) STOP 77                                                  DIAK 607
      READ(5,28) INPD,INPK                                                    DIAK 608
      IF(INPK.EQ.1) GO TO 1216                                               DIAK 609
      IF(INPK.EQ.2.AND.(INPD.EQ.1.OR.INPD.EQ.4)) GO TO 1216                 DIAK 610
C      NEXT RUN IS NOT SOLVABLE WITH PRESENT STARTING GAINS - SO STOP      DIAK 611
      WRITE(9,3008)                                                           DIAK 612
      STOP 11                                                                  DIAK 613
888  API=ABS(P(I,J))                                                         DIAK 614
      IF(API.LT.1.E-20) GO TO 105                                           DIAK 615
      106 RAT=P(I,J)/PI(I,J)                                                 DIAK 616
      RAT=ABS(RAT)                                                            DIAK 617
      IF(RAT.EE)105,105,107                                                  DIAK 618
C      COUNT CONVERGED ELEMENTS                                             DIAK 619
      105 ICT=ICT+1                                                           DIAK 620
      107 CONTINUE                                                            DIAK 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                    DIAK 623
      108 IF(NC-ICT)109,122,109                                             DIAK 624
      109 IF(ITERC-ITER)1000,1001,1001                                       DIAK 625
C      IF ITERC EQUALS ITER, NO CONVERGENCE - PRINT LAST TWO RICCATI MATDIAK 626
C      AND GO TO NEXT RUN                                                  DIAK 627
      1001 WRITE(9,121)ITER,ICT                                              DIAK 628
      120 FORMAT(1H1/7X,18H NOT CONVERGED IN 13,34H ITERATIONS-FIRST TERM TOU DIAK 629
      IFAIL WAS 14/)                                                         DIAK 630
      ITERM=ITER-1                                                            DIAK 631
      DO 3054 I=1,NX                                                         DIAK 632
      DO 3054 J=1,NX                                                         DIAK 633
      3054 P(I,J)=P(I,J)+PI(I,J)                                             DIAK 634
      WRITE(9,121)ITER                                                       DIAK 635
      121 FORMAT(///23H P MATRIX AT ITERATION 13//)                         DIAK 636
      CALL MP(MX,MX,NX,NX,P)                                                 DIAK 637
      WRITE(9,121)ITERM                                                       DIAK 638
      CALL MP(MX,MX,NX,NX,P)                                                 DIAK 639
      WRITE(9,39)                                                            DIAK 640
C      *** MODIFICATIONS                                                    DIAK 641
CR   READ(5,1215) IDUM                                                       DIAK 642
CR   IF(IDUM.GT.0) STOP 77                                                  DIAK 643
CR   READ(5,28) INPD,INPK                                                  DIAK 644
CR   IF(INPK.EQ.1) GO TO 1216                                             DIAK 645
CR   IF(INPK.EQ.2.AND.(INPD.EQ.1.OR.INPD.EQ.4)) GO TO 1216             DIAK 646
C   NEXT RUN IS NOT SOLVABLE WITH PRESENT STARTING GAINS - SO STOP      DIAK 647
CR   WRITE(9,3008)                                                         DIAK 648
CR   STOP 11                                                                DIAK 649
C      *** MODIFICATIONS                                                    DIAK 650
      122 CONTINUE                                                            DIAK 651
C      COMPUTE OPTIMAL GAINS                                                DIAK 652
C      K = -(D**Q*D):*D**Q*M - (D**Q*D):*G1**0                            DIAK 653
      DO 3056 I=1,NX                                                         DIAK 654
      DO 3056 J=1,NX                                                         DIAK 655
      3056 P(I,J)=P(I,J)+PI(I,J)                                             DIAK 656
      DO 125 I=1,NU                                                           DIAK 657
      DO 125 J=1,NX                                                         DIAK 658

```

Figure 84. Program DIAK Program Listing (Continued)

# Contrails

```

      AK(I,J)=-W(I,J)
      DO 125 K=1,NX
125  AK(I,J)=AK(I,J)-W1(I,K)*P(K,J)
C    SET COMMAND FEEDFORWARD GAINS TO ZERO
      NXMNC=NX-NCS+1
      DO 86 I=1,NU
      JJ=I+JF-NU
      DO 86 J=NXMNC,NX
C *** MODIFICATIONS
C    A(I,I)=0.
C *** MODIFICATIONS
      86 AK(I,J)=0.
C *** MODIFICATIONS
C    RECOMPUTE A - CLOSED LOOP STABILITY MATRIX
      REWIND 2
      READ(2) A
      DO 88 I=1,NX
      DO 88 J=1,NX
      DO 88 K=1,NU
      88 A(I,J)=A(I,J)+G1(I,K)*AK(K,J)
C *** MODIFICATIONS
C    PRINT GAINS MATRIX AND RICCATI MATRIX
4004  FORMAT(1H1/7X,13H GAINS MATRIX//)
4010  WRITE(9,4005)
4005  FORMAT(1H1/7X,13H RICCATI MATRIX//)
      CALL MP(MX,MX,NX,NX,P)
      WRITE(9,4004)
      CALL MP(MU,MX,NU,NX,AK)
C    RE-READ H AND M MATRICES FROM DISC
      REWIND 2
      READ(2)
      READ(2) H,AM
      IF (NCONT.LT,2) GO TO 82
C    RECOMPUTE QUADRATIC WEIGHTS ON CONTROL RATES
      NSCSS=NF-NU+1
      DO 80 I=NSCSS,NF
      II=I+NSCRR-NSCSS
      JJ=I-NSCSS+1
      DO 80 J=NSCSS,NF
      JJ=J+NSCRR-NSCSS
      80 QQ(1,JJ)=-P(1,J)*G1(I,J)/(H(JJ,J)*D(II,I))
      REWIND 2
      READ(2) F
      NUU=NI*NU
      NCU=0
      DO 84 I=1,NU
      DO 84 J=NSCSS,NF
      IF (ABS(AK(I,J)).GT,.05) GO TO 84
      NCU=NCU+1
      84 CONTINUE
      IF (NCU.EQ,NUU) GO TO 85
      IF (NQ.GT,ITERQ) GO TO 85
      NQ=NQ+1
      INPK=3
      INPD=2
      GO TO 81
      85 CONTINUE
      WRITE(6,83) IRUN
      83 FORMAT(17H0 MATRIX FOR CASE,I3)
      CALL OUTP(MR,MR,NR,NR,QQ,6)
      82 CONTINUE
      WRITE(6,7776)
7776  FORMAT(20(4H ))
C    PUNCH IDENTIFICATION
      WRITE(6,9010) IRUN
9010  FORMAT(21HGAINS MATRIX FOR CASE,I3)

```

DIAK 659  
 DIAK 660  
 DIAK 661  
 DIAK 662  
 DIAK 663  
 DIAK 664  
 DIAK 665  
 DIAK 666  
 DIAK 667  
 DIAK 668  
 DIAK 669  
 DIAK 670  
 DIAK 671  
 DIAK 672  
 DIAK 673  
 DIAK 674  
 DIAK 675  
 DIAK 676  
 DIAK 677  
 DIAK 678  
 DIAK 679  
 DIAK 680  
 DIAK 681  
 DIAK 682  
 DIAK 683  
 DIAK 684  
 DIAK 685  
 DIAK 686  
 DIAK 687  
 DIAK 688  
 DIAK 689  
 DIAK 690  
 DIAK 691  
 DIAK 692  
 DIAK 693  
 DIAK 694  
 DIAK 695  
 DIAK 696  
 DIAK 697  
 DIAK 698  
 DIAK 699  
 DIAK 700  
 DIAK 701  
 DIAK 702  
 DIAK 703  
 DIAK 704  
 DIAK 705  
 DIAK 706  
 DIAK 707  
 DIAK 708  
 DIAK 709  
 DIAK 710  
 DIAK 711  
 DIAK 712  
 DIAK 713  
 DIAK 714  
 DIAK 715  
 DIAK 716  
 DIAK 717  
 DIAK 718  
 DIAK 719  
 DIAK 720  
 DIAK 721  
 DIAK 722  
 DIAK 723  
 DIAK 724

Figure 84. Program DIAK Program Listing (Continued)



# Contrails

```

C   PUNCH OPTIMAL GAINS                                DIAK 725
      CALL DUTP(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(ISOL,IDSOL,NX,NX,P,MX,KWA,DET)         DIAK 732
      IF((ISOL+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 I DIAK 737
      IT//)                                               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,J)=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 W)                                    DIAK 746
      DO 58 I=1,NU                                         DIAK 747
      DO 58 J=1,NX                                       DIAK 748
      58 W(I,J)=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 MP(MX,MX,NU,NX,AN)                             DIAK 753
      WRITE(6,9011) IRUN                                    DIAK 754
      9011 FORMAT(,21HKSTAR MATRIX FOR CASE,13)         DIAK 755
      CALL DUTP(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
      W(I,J)=HK(I,J)                                       DIAK 762
      AK(I,J)=0.                                           DIAK 763
      DO 894 K=1,NX                                       DIAK 764
      894 AK(I,J)=AK(I,J)+BK(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 QR(I,J)=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 6075 I=1,NX                                       DIAK 790

```

Figure 84. Program DIAK Program Listing (Continued)

# Contrails

```

DO 6075 J=1,NX
6075 P(I,J)=A(I,J)
CALL CAL(P,E,O,KWA,NX,MX,IMAX*2,IERR,EE)
IF(IERR.EQ.0) GO TO 896
WRITE(9,43)
43 FORMAT(1H1/7X,28H COVARIANCE MATRIX UNDEFINED//7X,27H IGNORE COVAR
IANCE ANALYSIS//)
GO TO 850
896 WRITE(9,4051)
4051 FORMAT(/7X,18H COVARIANCE MATRIX//)
CALL MP(MX,MX,NX,NX,E)
DO 4053 I=1,NR
DO 4053 J=1,NX
AN(I,J)=0.
DO 4053 K=1,NX
4053 AN(I,J)=AN(I,J)+H(I,K)*E(K,J)
DO 4054 I=1,NR
DO 4054 J=1,NR
WR(I,J)=0.
DO 4054 K=1,NX
4054 WR(I,J)=WR(I,J)+AN(I,K)*H(J,K)
WRITE(9,4055)
4055 FORMAT(1H1/7X,27H RESPONSE COVARIANCE MATRIX//)
CALL MP(MR,MR,NR,NR,WR)
DO 6077 I=1,NR
DO 6077 J=1,NR
OR(I,J)=OR(I,J)+WR(I,J)
6077 AJ=AJ+WR(I,J)*OR(I,J)
DO 7015 I=1,NX
DO 7015 J=1,NX
P(I,J)=0.
DO 7015 K=1,NX
7015 P(I,J)=P(I,J)+E(I,K)*AM(J,K)
DO 7016 I=1,NX
DO 7016 J=1,NX
Q(I,J)=0.
DO 7016 K=1,NX
7016 Q(I,J)=Q(I,J)+AM(I,K)*P(K,J)
WRITE(9,44)
44 FORMAT(1H1/7X,28H MEASUREMENT COVARIANCE MATRIX//)
CALL MP(MX,MX,NX,NX,Q)
DO 1112 I=1,NU
DO 1112 L=1,NX
W(I,L)=0.
DO 1112 K=1,NX
1112 W(I,L)=W(I,L)+AK(I,K)*E(K,L)
DO 6085 I=1,NU
DO 6085 J=1,NU
DQD(I,J)=0.
DO 6085 K=1,NX
6085 DQD(I,J)=DQD(I,J)+W(I,K)*AK(J,K)
WRITE(9,45)
45 FORMAT(1H1/7X,26H CONTROL COVARIANCE MATRIX//)
CALL MP(MU,MU,NU,NU,DQD)
IF(NOCOV.GT.2) GO TO 2
DO 1111 I=1,NX
DO 1111 J=1,NX
P(I,J)=0.
IF(E(I,I).LT.1.E-20) GO TO 1111
IF(E(I,J).LT.1.E-20) GO TO 1111
P(I,J)=E(I,J)/SQRT(E(I,I)*E(J,J))
1111 CONTINUE
WRITE(9,46)
46 FORMAT(1H1/7X,31H STATE CROSS-CORRELATION MATRIX//)
CALL MP(MX,MX,NX,NX,P)
DO 1113 I=1,NU

```

DIAK 791  
 DIAK 792  
 DIAK 793  
 DIAK 794  
 DIAK 795  
 DIAK 796  
 DIAK 797  
 DIAK 798  
 DIAK 799  
 DIAK 800  
 DIAK 801  
 DIAK 802  
 DIAK 803  
 DIAK 804  
 DIAK 805  
 DIAK 806  
 DIAK 807  
 DIAK 808  
 DIAK 809  
 DIAK 810  
 DIAK 811  
 DIAK 812  
 DIAK 813  
 DIAK 814  
 DIAK 815  
 DIAK 816  
 DIAK 817  
 DIAK 818  
 DIAK 819  
 DIAK 820  
 DIAK 821  
 DIAK 822  
 DIAK 823  
 DIAK 824  
 DIAK 825  
 DIAK 826  
 DIAK 827  
 DIAK 828  
 DIAK 829  
 DIAK 830  
 DIAK 831  
 DIAK 832  
 DIAK 833  
 DIAK 834  
 DIAK 835  
 DIAK 836  
 DIAK 837  
 DIAK 838  
 DIAK 839  
 DIAK 840  
 DIAK 841  
 DIAK 842  
 DIAK 843  
 DIAK 844  
 DIAK 845  
 DIAK 846  
 DIAK 847  
 DIAK 848  
 DIAK 849  
 DIAK 850  
 DIAK 851  
 DIAK 852  
 DIAK 853  
 DIAK 854  
 DIAK 855  
 DIAK 856

Figure 84. Program DIAK Program Listing (Continued)

# Contrails

```

DO 1113 J=1,NX
P(I,J)=0.
P(I,J)=W(I,J)*AK(I,J)
1113 CONTINUE
WRITE(9,47)
47 FORMAT(1H1/7X,41H CONTROL-STATE ROW-SUM CORRELATION MATRIX//)
CALL MP(MX,MX,NU,NX,P)
DO 1114 I=1,NR
DO 1114 J=1,NX
P(I,J)=0.
P(I,J)=AN(I,J)*H(I,J)
1114 CONTINUE
WRITE(9,48)
48 FORMAT(1H1/7X,42H RESPONSE-STATE ROW-SUM CORRELATION MATRIX//)
CALL MP(MX,MX,NR,NX,P)
DO 1115 I=1,NX
DO 1115 J=1,NX
P(I,J)=0.
IF(Q(I,I).LT.1.E-20) GO TO 1115
IF(Q(J,J).LT.1.E-20) GO TO 1115
P(I,J)=Q(I,J)/SQRT(Q(I,I)*Q(J,J))
1115 CONTINUE
WRITE(9,49)
49 FORMAT(1H1/7X,37H MEASUREMENT CROSS-CORRELATION MATRIX//)
CALL MP(MX,MX,NX,NX,P)
DO 1300 I=1,NX
DO 1300 J=1,NX
P(I,J)=0.
DO 1301 K=1,NX
1301 P(I,J)=P(I,J)+AM(I,K)*F(K,J)
P(I,J)=P(I,J)*AM(I,J)
1300 CONTINUE
WRITE(9,1302)
1302 FORMAT(1H1/7X,45H MEASUREMENT-STATE ROW-SUM CORRELATION MATRIX//)
CALL MP(MX,MX,NX,NX,P)
DO 1116 I=1,NU
DO 1116 J=1,NX
P(I,J)=0.
DO 1117 K=1,NX
1117 P(I,J)=P(I,J)+W(I,K)*AM(J,K)
P(I,J)=P(I,J)*W1(I,J)
1116 CONTINUE
WRITE(9,50)
50 FORMAT(1H1/7X,47H CONTROL-MEASUREMENT ROW-SUM CORRELATION MATRIX//)
1)
CALL MP(MX,MX,NU,NX,P)
2)
CONTINUE
DO 63 I=1,NX
63 Q(I,I)=SQRT(Q(I,I))
DO 64 I=1,NU
64 Q(QD(I,I))=SQRT(Q(QD(I,I)))
WRITE(9,65) ((I,QD(I,I)),I=1,NU)
65 FORMAT(1H1//20X,16H R.M.S. CONTROLS/, (5X,13,13X.E15,8))
WRITE(9,66) ((I,Q(I,I)),I=1,NX)
66 FORMAT(1H1//20X,20H R.M.S. MEASUREMENTS/, (5X,13,13X.E15,8))
DO 4056 I=1,NR
4056 WR(I,I)=SQRT(WR(I,I))
WRITE(9,4057) ((I,WR(I,I)),I=1,NR)
4057 FORMAT(1H1//20X,17H R.M.S. RESPONSES/, (5X,13,13X.E15,8))
IF(KCON.LT.NN) GO TO 6076
WRITE(9,60)
60)
60 FORMAT(1H1/7X,37H TOTAL RESPONSE COVARIANCE MATRIX//)
CALL MP(MR,MR,NR,NR,QR)
DO 61 I=1,NR
DO 61 J=1,NR
P(I,J)=0.

```

```

DIAK 857
DIAK 858
DIAK 859
DIAK 860
DIAK 861
DIAK 862
DIAK 863
DIAK 864
DIAK 865
DIAK 866
DIAK 867
DIAK 868
DIAK 869
DIAK 870
DIAK 871
DIAK 872
DIAK 873
DIAK 874
DIAK 875
DIAK 876
DIAK 877
DIAK 878
DIAK 879
DIAK 880
DIAK 881
DIAK 882
DIAK 883
DIAK 884
DIAK 885
DIAK 886
DIAK 887
DIAK 888
DIAK 889
DIAK 890
DIAK 891
DIAK 892
DIAK 893
DIAK 894
DIAK 895
DIAK 896
DIAK 897
DIAK 898
DIAK 899
DIAK 900
DIAK 901
DIAK 902
DIAK 903
DIAK 904
DIAK 905
DIAK 906
DIAK 907
DIAK 908
DIAK 909
DIAK 910
DIAK 911
DIAK 912
DIAK 913
DIAK 914
DIAK 915
DIAK 916
DIAK 917
DIAK 918
DIAK 919
DIAK 920
DIAK 921
DIAK 922

```

Figure 84. Program DIAK Program Listing (Continued)

# Contrails

```
IF (OR(I,I).LT.1.E-20) GO TO 61          DIAK 923
IF (OR(J,J).LT.1.E-20) GO TO 61          DIAK 924
P(I,J)=OR(I,J)/SQRT(OR(I,I)*OR(J,J))     DIAK 925
61 CONTINUE                               DIAK 926
WRITE(9,62)                               DIAK 927
62 FORMAT(1H1/7X,49H TOTAL RESPONSE CROSS-CORRELATION MATRIX//) DIAK 928
CALL MP(MX,MX,NR,NR,P)                   DIAK 929
DO 600 P I=1,NR                           DIAK 930
6002 OR(I,I)=SQRT(OR(I,I))                DIAK 931
WRITE(9,6081) ((I,OR(I,I)),I=1,NR)       DIAK 932
6081 FORMAT(1H1/7X,22HTOTAL R.M.S. RESPONSES/(3X,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 (NOPEQ.EQ.0) GO TO 897                DIAK 937
IF (NSTEP.EQ.0.AND.NRAND.EQ.0) GO TO 897 DIAK 938
CALL TIMER(A,GP,H,X,XT,DX,DX1,XLDXL,GN,GS,R,IPLR,ITITL,IUNIT,CL, DIAK 939
IT,DT,RT,YMAX,YMIN,IFLT,IRUN,ICATE,NSTEP,NRAND,NPLOT,NPRIN,NN,NX,NF, DIAK 940
2,NG,NFS,NR,MXP,MN,MX,MP,IN,NOP,NAME1,NAME2,SCAL,NEWY,T1,T2,NGLG) DIAK 941
897 CONTINUE                              DIAK 942
CALL POLES(NX,A,MX,RR,I)                 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,NCOCOV,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),1220,1230,1240,1250),INPD    DIAK 958
7777 CONTINUE                             DIAK 959
ENDFI: E 6                                DIAK 960
END                                        DIAK 961
```

Figure 84. Program DIAK Program Listing (Concluded)

# Contrails

```

OVERLAY (KON2,P,0)                                FFOC  2
PROGRAM FFOC                                       FFOC  3
DIMENSION F(40,40),G1(40,06),G2(40,02),H(40,40) , FFOC  4
* D(40,08),AM(40,40),Q(40,40),AK(06,40),DK(06,40),DJK(06,40), FFOC  5
* AKG(06,40),XI(09,09) ,BK(06,40),A(40,40),X(40,40),C(40,40), FFOC  6
* P(40,40),HDK(40,40),S(40,40),PR(008),DELK(06,40),AKP(06,40) FFOC  7
DIMENSION U(40,40),V(9,9),E(9,9),F5(40,40),DQ(6,40) , FFOC  8
* DDD(06,06),KWA(5,1),T(50,50),HB(50,40),UJV(50),DJVT(50),IF(50), FFOC  9
* JF(50),AMT(50,40),Y(40,40),Z(40,40),NORD(40) FFOC 10
C                                                    FFOC 11
C ARRAY DIMENSIONS                                  FFOC 12
C                                                    FFOC 13
C           MX=40                                    FFOC 14
C           MR=40                                    FFOC 15
C           MNN=2                                    FFOC 16
C           MU=6                                     FFOC 17
C           MM=40                                    FFOC 18
C           MFF=9                                    FFOC 19
C           MFR=4                                    FFOC 20
C           MF=50                                    FFOC 21
C INPUT INTEGER PARAMETERS                          FFOC 22
C   READ(5,1) IMAX,NITM,NOPR,NOCOV,NBEGIN ,NDIAK FFOC 23
C   READ(5,1) NX,NR,NU,NN,NFF,NF FFOC 24
1   FORMAT(40I2)                                     FFOC 25
C   READ(5,1) (NORD(I),I=1,NX) FFOC 26
C   ITER= FFOC 27
C   NM=NX FFOC 28
C   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
C   WRITE(9,11) NOCOV,NBEGIN,NOPR FFOC 32
11  FORMAT(//7X,8H NOCOV =,I3,5X,9H NBEGIN =,I3,5X,7H NOPR =,I3//) FFOC 33
C   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
C   READ(5,1277) AJSTAR FFOC 44
C   AJT=AJSTAR FFOC 45
C   WRITE(9,5035) FFOC 46
5035 FORMAT( //7X,29H LOWEST COST EXPECTED(AJSTAR)/) FFOC 47
C   WRITE(9,5033) AJSTAR FFOC 48
5033 FORMAT(//7X,(6G10,4)) FFOC 49
C   WRITE(9,2) (NORD(I),I=1,NX) FFOC 50
2   FORMAT(//7X,27H STATES ARE ORDERED AS SUCH//(7X,30I3)) FFOC 51
C   READ(5,1277) DPROC FFOC 52
C   READ(5,1277) ALAM,DELTA,ALAMD FFOC 53
1277 FORMAT(6G10,4) FFOC 54
3   NG=1 FFOC 55
C   NC=1 FFOC 56
C   NUNST=0 FFOC 57
C                                                    FFOC 58
C ROWS AND COLUMNS OF FIXED GAINS -- K1           FFOC 59
C   READ(5,1) (IF(I),JF(I),I=1,NF) FFOC 60
C   WRITE(9,20) FFOC 61
20  FORMAT(//7X,29H FIXED GAINS ROW COLUMN//) FFOC 62
C   WRITE(9,201) (IF(I),JF(I),I=1,NF) FFOC 63
201 FORMAT(21X,216) FFOC 64

```

Figure 85. Program FFOC Program Listing

# Contrails

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,MNM)	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,MNM)	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,NR,MM,NM,MU,NU,MNM,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,MNM,NX,NN,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,IDSOL,NX,NX,S,MX,KWA,DET)	FFOC	127
	IF((ISOL+IDSOL).LE,?) GO TO 1502	FFOC	128
	WRITE(9,1403) ISOL,IDSOL	FFOC	129
1403	FORMAT(1/7X,51H MEASUREMENT MATRIX IS NOT INVERTIBLE. ERROR CODE =	FFOC	130

Figure 85. Program FFOC Program Listing (Continued)

# Contrails

```

1,213//)
STOP 13
1502 DO 14.5 I=1,NU
      DO 14.5 J=1,NX
      AK(I, J)=0.
      DO 14.5 K=1,NX
1405 AK(I, J)=AK(I, J)+AKG(I, K)*S(K, J)
C.
C   DEFINE K2
      CALL INPT(AK, MU, MM)
      DO 14.4 I=1,NU
      DO 14.4 J=1,NX
      RK(I, J)=AK(I, J)*(1.-AKP(I, J))
1404 AK(I, J)=AK(I, J)*AKP(I, J)
      CALL INPT(RK, MU, MM)
      WRITE(9,1018)
1018 FORMAT(1H1/7X,23H INITIAL GAINS -- K1(1)//)
      CALL MP(MU, MX, NU, NM, AK)
      WRITE(9,1019)
1019 FORMAT(//7X,10H K2 MATPIX//)
      CALL MP(MU, MM, NU, NM, RK)
      IF(ALAM.LT..99) GO TO 1406
      DO 14.7 I=1,NU
      DO 14.7 J=1,NM
1407 DELK(I, J)=0.
      GO TO 1408
C
C   INPUT PRESENT FIXED GAINS -- K1(LAMBDA)
1406 CALL ZERO(AK, MU, MM)
      CALL INPT(AK, MU, MM)
      WRITE(9,1020)
1020 FORMAT(//7X,34H PRESENT FIXED GAINS -- K1(LAMBDA)//)
      CALL MP(MU, MM, NU, NM, AK)
C
C   INPUT FIXED PREDICTOR -- DELK1(LAMBDA)
      CALL ZERO(DELK, MU, MM)
      CALL INPT(DELK, MU, MM)
1408 WRITE(9,1021)
1021 FORMAT(//7X,35H PRESENT PREDICTOR -- DELK1(LAMBDA)//)
      CALL MP(MU, MM, NU, NM, DELK)
C
C   TAKE STEP IN LAMBDA
172 ALAM=ALAM-DELT
      WRITE(9,173) ALAM
173 FORMAT(//7X,9HLAMBDA = .F9,3)
C
C   PREDICT GAINS FOR NEW LAMBDA
      DO 310 I=1,NU
      DO 310 J=1,NM
      DK(I, J)=DELK(I, J)
      DELK(I, J)=AK(I, J)
310 AK(I, J)=AK(I, J)+DK(I, J)
C
C   INITIAL CONDITIONS
C
      NL=1
      LAST=?
      NCHK=
      NGD=0
      NIT=0
      EPS=EPS1
      AJI=1.,*AJT
      AJL=AJT
7     WRITE(9,405) NIT
4051 FORMAT( //7X,9HITERATION,I3)
      WRITE(9,54) EPS

```

```

FFOC 131
FFOC 132
FFOC 133
FFOC 134
FFOC 135
FFOC 136
FFOC 137
FFOC 138
FFOC 139
FFOC 140
FFOC 141
FFOC 142
FFOC 143
FFOC 144
FFOC 145
FFOC 146
FFOC 147
FFOC 148
FFOC 149
FFOC 150
FFOC 151
FFOC 152
FFOC 153
FFOC 154
FFOC 155
FFOC 156
FFOC 157
FFOC 158
FFOC 159
FFOC 160
FFOC 161
FFOC 162
FFOC 163
FFOC 164
FFOC 165
FFOC 166
FFOC 167
FFOC 168
FFOC 169
FFOC 170
FFOC 171
FFOC 172
FFOC 173
FFOC 174
FFOC 175
FFOC 176
FFOC 177
FFOC 178
FFOC 179
FFOC 180
FFOC 181
FFOC 182
FFOC 183
FFOC 184
FFOC 185
FFOC 186
FFOC 187
FFOC 188
FFOC 189
FFOC 190
FFOC 191
FFOC 192
FFOC 193
FFOC 194
FFOC 195
FFOC 196

```

Figure 85. Program FFOC Program Listing (Continued)

# Contrails

```

C
C PRINT GAINS
WRITE(9,4004)
4004 FORMAT(//7X,13H GAINS MATRIX//)
CALL MP(MU,MM,NU,NM,AK)
C
C INITIALIZE GRADIENT PROJECTION
IF(NG,NE,.) GO TO 5
DO 16 I=1,NU
DO 16 J=1,NM
16 AKP(I,J)=AK(I,J)
C
C INITIALIZE ARRAYS
C
C COMPUTE F=G1*K*(LAMBDA)*M=A
C
5 DO 12 J=1,NX
DO 12 I=1,NU
C(I,J)=0.
DO 12 K=1,NM
C(I,J)=C(I,J)+(AK(I,K)+RK(I,K)*ALAM)*AM(K,J)
12 CONTINUE
IF(LAST,NE,.) GO TO 66
DO 15 I=1,NU
DO 15 J=1,NM
151 DJDK(I,J)=AK(I,J)+ALAM*RK(I,J)
WRITE(9,1022)
1022 FORMAT(//7X,37H K*(LAMBDA) FOR RESPONSE CALCULATIONS//)
CALL MP(MU,MM,NU,NM,DJDK)
66 CONTINUE
DO 8 I=1,NX
DO 8 J=1,NM
A(I,J)=F(I,J)
DO 8 K=1,NU
8 A(I,J)=A(I,J)+G(I,K)*C(K,J)
IF(NL,NE,.) GO TO 184
C
C CHECK FOR STABILITY OF A
CALL POLES(NX,A,MX,RR,M)
KK=0
II=1
921 IF(RR/II).LT.0.) GO TO 185
C
C IF UNSTABLE -- HALVE DELTA LAMBDA AND PREDICTOR
69 I=NUNST+1
WRITE(9,188) I
188 FORMAT(//7X,24HUNSTABLE -- CHANGE GAINS//7X,13,14HUNSTABILITY//)
IF(NUNST.EQ.3) GO TO 5071
ALAM=.5*ALAM+DELT
IF(NUNST.EQ.1) GO TO 5071
C
C FIRST OR THIRD INSTABILITY -- HALVE PREDICTOR
DO 5040 I=1,NU
DO 5040 J=1,NM
AK(I,J)=AK(I,J)-DK(I,J)
5040 DELK(I,J)=DK(I,J)/2.
AJT=AJSTAR
NUNST=NUNST+1
GO TO 172
C
C SECOND OR FOURTH INSTABILITY -- HALVE DELTA LAMBDA
5071 DELT=.5*DELT
C *** MODIFICATIONS
IF(DELT.LE.1.0E-06)WRITE(9,7740)DELT
7740 FORMAT(1H1,/,1X,5)H*** EXIT ON DETECTING VERY SMALL VALUE FOR DELT
IT ***.610.4)

```

Figure 85. Program FFOC Program Listing (Continued)



# Contrails

```

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 504
C *** MODIFICATIONS
C ALAMD=ALAMD+DELT
C *** MODIFICATIONS
AJT=A1STAR
NUNST=?
GO TO 172
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)+G1(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 120
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
DQ(I,J)=0.
DO 18 K=1,NR
18 DQ(I,J)=DQ(I,J)+D(K,I)*Q(K,J)
DO 434 I=1,NU
DO 434 J=1,NU
DQD(I,J)=0.
DO 434 K=1,NR
434 DQD(I,J)=DQD(I,J)+DQ(I,K)*D(K,J)
C
C COMPUTE COVARIANCE MATRIX
CALL COVAR(XI,A,C,X,G2,S,E,CS,V,U,NX,VFF,NN,MX,MFF,MFR,MNN,IMAX,
1ITER,1,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

```

```

FFOC 263
FFOC 264
FFOC 265
FFOC 266
FFOC 267
FFOC 268
FFOC 269
FFOC 270
FFOC 271
FFOC 272
FFOC 273
FFOC 274
FFOC 275
FFOC 276
FFOC 277
FFOC 278
FFOC 279
FFOC 280
FFOC 281
FFOC 282
FFOC 283
FFOC 284
FFOC 285
FFOC 286
FFOC 287
FFOC 288
FFOC 289
FFOC 290
FFOC 291
FFOC 292
FFOC 293
FFOC 294
FFOC 295
FFOC 296
FFOC 297
FFOC 298
FFOC 299
FFOC 300
FFOC 301
FFOC 302
FFOC 303
FFOC 304
FFOC 305
FFOC 306
FFOC 307
FFOC 308
FFOC 309
FFOC 310
FFOC 311
FFOC 312
FFOC 313
FFOC 314
FFOC 315
FFOC 316
FFOC 317
FFOC 318
FFOC 319
FFOC 320
FFOC 321
FFOC 322
FFOC 323
FFOC 324
FFOC 325
FFOC 326
FFOC 327
FFOC 328

```

Figure 85. Program FFOC Program Listing (Continued)

# Contrails

```

      C(I,J)=0.
      DO 6 K=1,NM
6      C(I,J)=C(I,J)+(AK(I,K)+BK(I,K)*ALAM)*AM(K,J)
      GO TO 69
4      CONTINUE
      AJT=1.E+20
      IF(NG,NE,1) GO TO 43
      GO TO 1756
17  ITR=1
C
C  CALCULATE COST
C
      AJT=0.
      DO 34 I=1,NX
      DO 34 J=1,NX
34  AJT=AJT+R(I,J)*X(I,J)
      WRITE(9,122) AJT
122  FORMAT(//7X,4H COST = ,E15.8)
      IF(NH=GIN.EQ.0) GO TO 10
      IF(AJT.LT.0.) GO TO 1510
      IF(AJT.LT.AJ1) GO TO 10
1510  WRITE(9,1402)
1402  FORMAT(//7X,43H COST EXCEEDS 10 TIMES LOWEST COST EXPECTED//)
      WRITE(9,14)
14  FORMAT(1H1//7X,18H COVARIANCE MATRIX//)
      CALL MP(MX,MX,NX,NX,X)
      STOP 11
10  IF(NG,NE,1) GO TO 450
130  IF(NG,EQ,2) GO TO 67
C
C  COMPUTE XM, MXM INVERSE
      DO 125 I=1,NX
      DO 125 J=1,NM
      C(I,J)=0.
      DO 125 K=1,NX
125  C(I,J)=C(I,J)+X(I,K)*AM(J,K)
      CALL TRANS(AMT,X,T,DQD,NX,MX,NF,NU,MF,RR,IF)
C
C  COSTATE CALCULATION
C
      CALL COSTAT (R,A,S,X,ES,Y,Z,E,U,V,KWA,MX,MFR,MFF,NX,NFF,IMAX,IERR)
      IF(IERR.EQ.0) GO TO 1400
      WRITE(9,1401)
1401  FORMAT(//7X,45H COSTATE MATRIX UNSTABLE WHEN STATE MATRIX IS//)
      STOP 11
C
C  GRADIENT CALCULATION
C
C  COMPUTE DDQ(H+DKM)XM(HXM)-1 -- 2G1*SXM(MXM)-1
1400  DO 19 I=1,NU
      DO 19 J=1,NX
      R(I,J)=0.
      DO 19 K=1,NR
19  R(I,J)=R(I,J)+DQ(I,K)*HDK(K,J)
      DO 30 I=1,NU
      DO 30 J=1,NX
      X(I,J)=0.
      DO 30 K=1,NX
30  X(I,J)=X(I,J)+G1(K,I)*S(K,J)
      DO 126 I=1,NU
      DO 126 J=1,NM
      DJDK(I,J)=0.
      DO 126 K=1,NX
126  DJDK(I,J)=DJDK(I,J)+(R(I,K)+X(I,K))*C(K,J)*2.
      GO TO 450

```

Figure 85. Program FFOC Program Listing (Continued)

# Contrails

```

67   CONTINUE                                     FFOC 395
C                                         FFOC 396
C   PROJECTED GRADIENT                         FFOC 397
DO 104 I=1,NU                                  FFOC 398
DO 105 J=1,NM                                  FFOC 399
106   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,X,Y,Z,S,R,ES,E+U,V,XI,DQ,AKG,DDQ,HDK,KWA, FFOC 407
INX,NFF,NN,NM,NU,NR,MX,MFF,MFR,MNN,MM,MU,MR,ITER,IMAX,IERR,NOCOV) FFOC 408
ITER=                                           FFOC 409
853   CALL POLES(NX,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
NREGIA = 0                                       FFOC 414
IF(NG,NE.1) GO TO 43                           FFOC 415
IF(NGD,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 22                                         FFOC 421
852   WRITE(9,1024)                            FFOC 422
1024  FORMAT(//7X,31H GRADIENT TRANSFORMATION MATRIX//) FFOC 423
CALL DINSR(T,DJDK,DJV,DJVT,IF,JF,NF,NJ,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(NGR.GT.0) GO TO 500                        FFOC 427
C                                         FFOC 428
C   CORRECT FOR INSTABILITY WHILE COMPUTING GRADIENT FFOC 429
NGR=1                                           FFOC 430
IF(AJT.GT.AJM) NGR=0                           FFOC 431
IF(AJT.LT.0.) NGR=0                            FFOC 432
IF(NGR,NE.0) GO TO 500                        FFOC 433
NGRR=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)=EPS*DK(I,J)/FPSS                       FFOC 439
IF(NGD,EQ.2) AK(I,J)=AK(I,J)+2.*DK(I,J)       FFOC 440
IF(NGD,EQ.3) AK(I,J)=AK(I,J)+.5*DK(I,J)       FFOC 441
IF(NGD,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.4) 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.0ROC) LAST=1                         FFOC 451
AJLAT=AJT                                       FFOC 452
IF(INIT.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)
WRITE(9,5031) SUM
5031 FORMAT(//,7X,16HGRADIENT NORM = ,F15.3)
5009 DO 103 I=1,NU
      DO 103 J=1,NM
103  DJDK(I,J)=DJDK(I,J)/SUM
121  WRITE(9,39)
39   FORMAT(//7X,19HNORMALIZED GRADIENT//)
38   CALL IP(MU,MM,NU,NM,DJDK)
5010 NL=0
      IF(LAST.EQ.1) GO TO 5
C
C   COUNT GRADIENT DIRECTIONS
      NGRB=
      NGD=NGD+1
      NCHK=
      IF(NOPR.GT.0) GO TO 102
      IF(NGD.EQ.3) NCHK=1
      IF(NGD.EQ.3) NGD=0
      GO TO 110
102  IF(NGD.EQ.2) NCHK=1
      IF(NGD.EQ.2) NGD=0
110  AJOG=AJT
      AJOGL=AJL
5011 DO 42 I=1,NU
      DO 42 J=1,NM
      AKG(I,J)=AK(I,J)
      42 DK(I,J)=-EPS          *DJDK(I,J)
      NG=0
      NC=1
      NCO=0
      GO TO 44
C
C   STEP SIZE LOGIC
C
43   IF(AJT.LT.0.) GO TO 301
      IF(AJT.LT.AJL) GO TO 41
C
C   UNSTABLE -- HALVE STEP SIZE
301  NCO=NC
      NC=1
      NIT=NIT-1
      IF(NCO.GT.1) NIT=NIT-1
      AJT=AJOG
      AJL=AJOGL
      EPS=EPS/2.
C *** MODIFICATIONS
      IF(EPS.LE.1.0E-06)WRITE(9,7720)EPS
7720 FORMAT(1H1,//,1X,50H*** EXIT ON DETECTING VERY SMALL VALUE FOR EPS
1 ***,610,4)
      IF(EPS.LE.1.0E-06)GO TO 173;
C *** MODIFICATIONS
5012 DO 123 I=1,NU
      DO 123 J=1,NM
      AK(I,J)=AKG(I,J)
123  DK(I,J)=.5*DK(I,J)
      GO TO 44
41   IF(NC.GT.1) GO TO 45
      IF(AJT.GT.AJL) GO TO 47
      IF(NC.EQ.1) GO TO 47
C
C   DOUBLE STEP SIZE
5013 DO 46 I=1,NU
      DO 46 J=1,NM
46   AK(I,J)=AK(I,J)+DK(I,J)
      NIT=NIT+1

```

Figure 85. Program FFOC Program Listing (Continued)

# Contrails

	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=2.*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(AJD,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=AMIN(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)/EPS	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)

# Contrails

```
59 FORMAT( ///7X,12HSTEP SIZE = ,E15,R)          FFOC 593
   NG=1                                           FFOC 594
   AJSTAP=AJT                                     FFOC 595
   NUNST=C                                        FFOC 596
C                                                 FFOC 597
C INITIALIZE NEW PREDICTOR                        FFOC 598
  DO 311 I=1,NU                                   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
5018 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
316 FORMAT(3G10.4,3,3X,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 IUTP(MU,MM,NU,NM,AK,1)                   FFOC 624
   WRITE(1,7705)                                  FFOC 625
7705 FORMAT(23(4H   ))                           FFOC 626
   WRITE(1,7706)                                  FFOC 627
7706 FORMAT(15HPREDICTOR GAINS)                  FFOC 628
   CALL IUTP(MU,MM,NU,NM,DELK,1)                 FFOC 629
   WRITE(1,7705)                                  FFOC 630
   ENDFILE 1                                       FFOC 631
5025 CONTINUE                                    FFOC 632
C *** MODIFICATIONS                              FFOC 633
   END                                             FFOC 634
```

Figure 85. Program FFOC Program Listing (Concluded)

# Contrails

```

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 - 1974 DATAK 7
C                                                  DATAK 8
C          SUBPROGRAMS CALLED DATAK 9
C          DERRM DATAK 10
C          ERRM DATAK 11
C          ODTAK DATAK 12
C          OFFOC DATAK 13
C          DLSA DATAK 14
C          FINK DATAK 15
C                                                  DATAK 16
COMMON/INOUT/IR,IW,IPRINT,INS,RT,LOCATF,NUILL,MARK(20), DATAK 17
1JQ,JS,JSO,JF,JD DATAK 18
COMMON /INF/ NXM,NRM,NUM,CODE,MS1,MS2,MS3,MS4 DATAK 19
DIMENSION S1(00500) DATAK 20
C          DIMENSION A(NXM,NXM),4(NXM,NUM),C(NRM,NXM),D(NRM,NUM) DATAK 21
DIMENSION S2(17000) DATAK 22
C          DIMENSION R1(NXM,NUM),R2(NXM,NUM) DATAK 23
C          DIMENSION C1(NRM,NXM),C3(NRM,NXM) DATAK 24
C          DIMENSION D11(NRM,NUM),RK(INUM,NRM),BKCR(NUM,NXM) DATAK 25
C                                                  DATAK 26
C          DIMENSION CC(NXRM,NXRUM),NAME(NXRUM) DATAK 27
C          DIMENSION S3(00001) DATAK 28
C          DIMENSION S4(00001) DATAK 29
C          DATA HODIA,HOFFO,HOLSA/4HSDTA,4HSEFO,4HSLSA/ DATAK 30
C          DATA HPRIN/4HPRIN/ DATAK 31
C                                                  DATAK 32
C          COMPUTE ARRAY START INDEXES DATAK 33
C                                                  DATAK 34
C          NXRM=NXM+NRM DATAK 35
C          NXRUM=NXRM+NUM DATAK 36
C          M1=1 * M2=M1+NXM*NXM * M3=M2+NXM*NUM * M4=M3+NRM*NXM DATAK 37
C          M5=M4+NRM*NUM DATAK 38
C          N1=1 * N2=N1+NXM*NUM * N3=N2+NXM*NUM * N4=N3+NRM*NXM DATAK 39
C          N5=N4+NRM*NXM * N6=N5+NRM*NUM * N7=N6+NUM*NRM * N8=N7+NUM*NXM DATAK 40
C          K1=1 * K2=K1+NXRM*NXRUM DATAK 41
C          K3=K2+NXRUM DATAK 42
C                                                  DATAK 43
C          CHECK IF SCRATCH ARRAY SIZES ARE SUFFICIENT DATAK 44
C                                                  DATAK 45
C          IF (K3.GT.N8)N8=K3 DATAK 46
C          IF ((M5.GT.MS1).OR.(N8.GT.MS2)) DATAK 47
C          1CALL DERRM(M5,N8,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
C          IF (CODE.EQ.HODIA)GO TO 160 DATAK 52
C          IF (CODE.EQ.HOFFO)GO TO 260 DATAK 53
C          IF (CODE.EQ.HOLSA)GO TO 360 DATAK 54
C          CALL FPRM(1,4HDATA,4HK ,3,0,IW) DATAK 55
160 CONTINUE DATAK 56
C          CALL ODTAK(S1(M1),S1(M2),S1(M3),S1(M4),S2(N1),S2(N2),S2(N3) DATAK 57
C          1,S2(N4),S2(N5),S2(N6),NXM,NRM,NUM) DATAK 58
C          GO TO 460 DATAK 59
260 CONTINUE DATAK 60
C          CALL OFFOC(S1(M1),S1(M2),S1(M3),S1(M4),S2(N1),S2(N2),S2(N3) DATAK 61
C          1,S2(N4),S2(N5),S2(N6),NXM,NRM,NUM) DATAK 62
C          GO TO 460 DATAK 63
360 CONTINUE DATAK 64

```

Figure 86. Program DATAK Program Listing

# Contrails

```
CALL DLSA(S1(M1),S1(M2),S1(M3),S1(M4),S2(N1),S2(N2),S2(N3)      DATAK 65
I,S2(N4),S2(N5),S2(N6),S2(N7),NX,NR,NU,NXM,NRM,NUM)              DATAK 66
CALL FINK(S1(M1),S1(M2),S1(M3),S1(M4),S2(K1),S2(K2),            DATAK 67
INX,NR,NU,NXM,NRM,NUM,IXRM,NXRUM)                                DATAK 68
460 CONTINUE                                                    DATAK 69
C                                                                    DATAK 70
C   RETURN TO MAIN OVERLAY                                       DATAK 71
C                                                                    DATAK 72
C   END                                                            DATAK 73
```

Figure 86. Program DATAK Program Listing (Concluded)



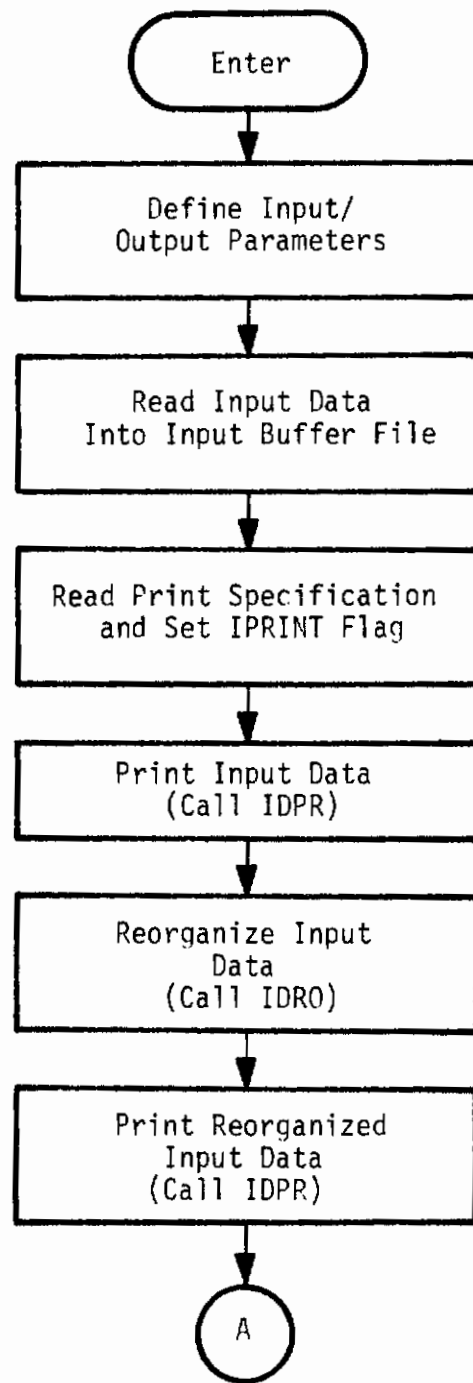


Figure 87. Subroutine KORGE2 Flow Chart

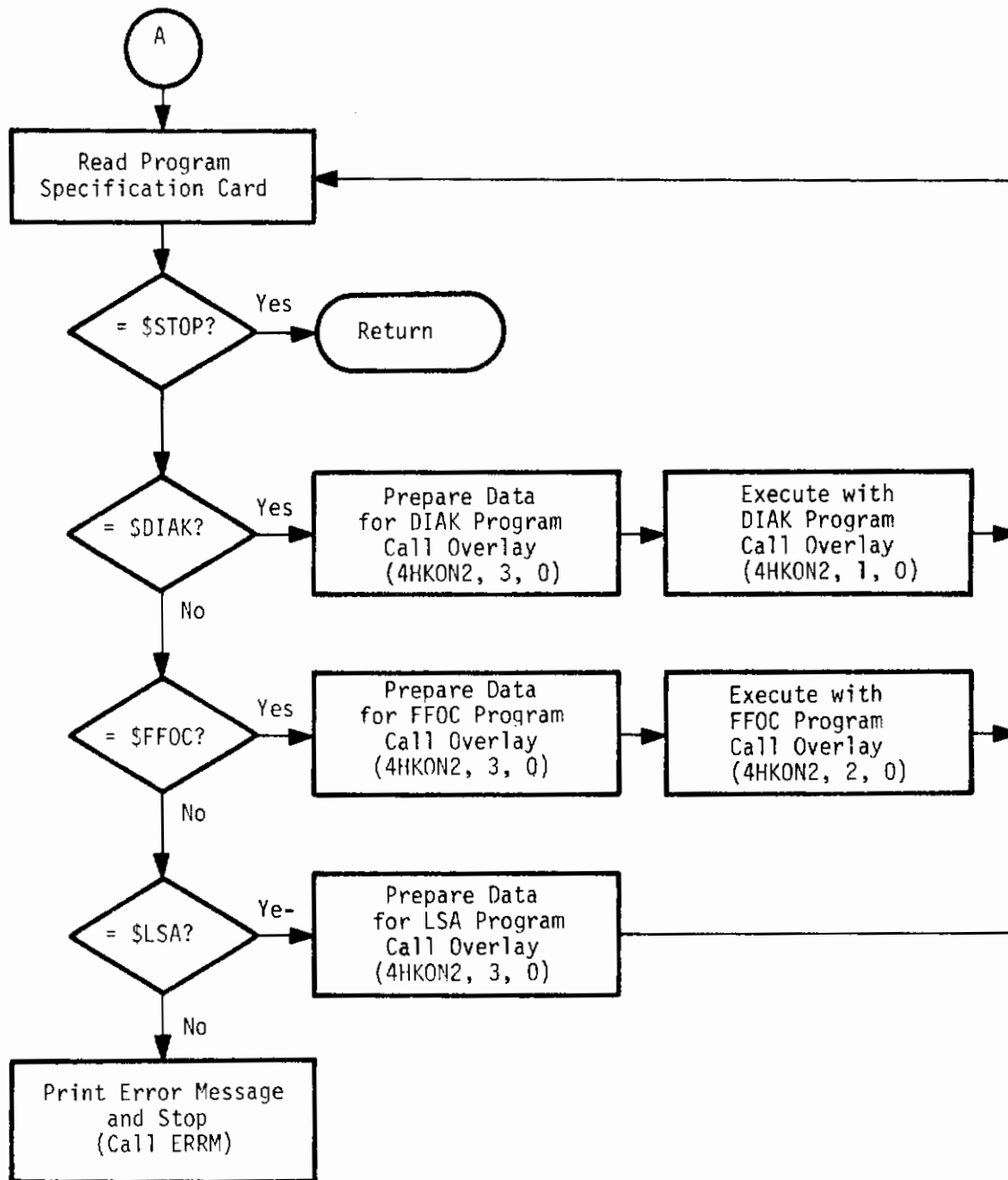


Figure 87. Subroutine KOR2 Flow Chart (Concluded)

# Contracts

```

SUBROUTINE KOR62
C
C ANALYSIS - A F KONAR / J K MAMESH - THE HONEYWELL INC
C PURPOSE - TO ORGANIZE EXECUTION OF KONPACT-2 PROGRAMS
C DATE WRITTEN - JULY 1975
C
C SUBPROGRAMS CALLED
C   IOP0
C   IOPR
C   ERUM
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 *$*
C   JQ      FILE NO FOR QUADRUPEL DATA FILE
C   JS      FILE NO FOR SCRATCH FILE
C   JS1     FILE NO FOR SDSTP FILE
C   JF      FILE NO FOR FDATA FILE
C   JD      FILE NO FOR DDATA FILE
C
C DIMENSION CARD(20)
C COMMON /INPUT/IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20),
1JQ,JS,JSD,JF,JD
C COMMON /INF/ NXM,NRM,AUM,CORE
C INTEGER HINSE,HLOCA,HNULL,HDLR
C DATA HINSE,HLOCA,HNULL,HDLR/4HINSE,4HLOCA,4HNULL,4HDLR/
C DATA HPRIN,HTHIN,HERYT/4HPRIN,4HTHIN,4HERYT/
C DATA HTPUT,HNAL,HPUT/4HTPUT,4HNAL,4HPUT /
C DATA HDLIA,HDFED,HDLISA/4HDLIA,4HDFED,4HDLISA/
C DATA HDSTO/4HSTO/
C DATA -C/1HC/
C
C DEFINE INPUT/OUTPUT PARAMETERS
C
C   IR=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
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
115 CONTINUE
C   WRITE(IR,12)CARD
120 FORMAT(20A4)
130 FORMAT(A1,A3)
C   GO TO 110
140 CONTINUE
C   ENDFILE IR
C   REWIND IR
C
C   READ PRINT SPECIFICATION AND SET IPRINT
C
C
142 CONTINUE
C   READ(IR,12) CARD

```

Figure 88. Subroutine KOR62 Program Listing

# Contrails

```
DECODE(4,139,CARD(1))CC,DUMMY          KORG2 65
IF(CC.EQ.HC)GO TO 142                   KORG2 66
IF(CARD(1).NE.HPRIN)GO TO 152          KORG2 67
IF(CARD(3).EQ.HTHIN)IPRINT=1          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.HTPUT)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 158 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,31H*** 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(IP,120) CARD                          KORG2129
IF(CARD(1).EQ.HPRIN)GO TO 159          KORG2130
```

Figure 88. Subroutine KORG2 Program Listing (Continued)

# Contrails

```
IF(CARD(1).EQ.HDST0) RETURN KORG2131
IF(CARD(1).EQ.HDDIA) GO TO 180 KORG2132
IF(CARD(1).EQ.HDFF0) GO TO 260 KORG2133
IF(CARD(1).EQ.HDLS0) GO TO 360 KORG2134
CALL FPRM(1,4HKORG,4H2 .0.),IW) 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(4HKON2,3,0) KORG2141
CALL OVERLAY(4HKON2,1,0) KORG2142
GO TO 159 KORG2143
260 CONTINUE KORG2144
CODE=CARD(1) KORG2145
CALL OVERLAY(4HKON2,3,0) KORG2146
CALL OVERLAY(4HKON2,2,0) KORG2147
GO TO 159 KORG2148
360 CONTINUE KORG2149
CODE=CARD(1) KORG2150
CALL OVERLAY(4HKON2,3,0) KORG2151
GO TO 159 KORG2152
END KORG2153
```

Figure 88. Subroutine KOR2 Program Listing (Concluded)

# Contrails

```

SUBROUTINE TIMER(A,G2,HDK,X,XT,DX,DX1,XL      ,GN,GS,R,IPLR,ITITL,  TIMER  2
(IUNIT,   CL,T,DT,ST,YMAX,YMIN,IFLT,IRUN,  IDATE,NSTEP,NRAND,NPLOT,TIMER  3
2NPRIN,NN,NX,NF,NG,NC,NP,MXR,MV,MA,MXRSP,NOP,NAME1,NAME2,SCAL,NEWY,TIMER  4
3T1,T2,NGLG)  TIMER  5
  DIMENSION A(MX,MX),G2(MX,MN),HDK(MX,MX),X(MX),DX(MX),DX1(MX)  TIMER  6
  DIMENSION XI(MX,MN),XL(MX,2)      ,GN(MX,MN),GS(MX,MN),R(MXRSP)  TIMER  7
  DIMENSION IPLR(MXR),ITITL(MXR),IUNIT(MXR),IRUF(15),LW(10)  TIMER  8
  DIMENSION YMAX(MXR),YMIN(MXR),CL(MN,1)  TIMER  9
  DIMENSION SCAL (MXR)  TIMER 10
  DIMENSION NEWY(48)  TIMER 11
  INTEGER BLANK  TIMER 12
  RUN=IRUN  TIMER 13
  IF(NOP.GT.0) GO TO 22  TIMER 14
  WRITE(9,203)  TIMER 15
203 FORMAT(1H1/7X,44HNO. OF PLOTS IS ZERO - IGNORE TIME RESPONSES)  TIMER 16
  RETURN  TIMER 17
  22 WRITE(9,102)  TIMER 18
102 FORMAT(1H1/7X,14HTIME RESPONSES//)  TIMER 19
  BLANK=10H  TIMER 20
  LW(1)=10H*  TIMER 21
  LW(2)=10H *  TIMER 22
  LW(3)=10H *  TIMER 23
  LW(4)=10H *  TIMER 24
  LW(5)=10H *  TIMER 25
  LW(6)=10H *  TIMER 26
  LW(7)=10H *  TIMER 27
  LW(8)=10H *  TIMER 28
  LW(9)=10H *  TIMER 29
  LW(10)=10H *  TIMER 30
C  NSTEP=0  NO STEP INPUTS  TIMER 31
C    =1  STEP COMMANDS  TIMER 32
C    =2  STEP GUSTS  TIMER 33
C    =3  ROTH  TIMER 34
C    =4  NO STEP INPUTS - TRANSIENTS ONLY  TIMER 35
C  VRAND=0  NO RANDOM INPUTS  TIMER 36
C  VRAND=1  GUSTS  TIMER 37
C  NPRIN=0  DON'T PRINT RESPONSES  TIMER 38
C  NPRIN=1  PRINT RESPONSES  TIMER 39
C  NPLOT=0  NO PLOTS  TIMER 40
C  NPLOT=1  CALCOMP PLOTS  TIMER 41
C  NPLOT=2  LINE PRINTER PLOTS  TIMER 42
C  NPLOT=3  ROTH  TIMER 43
  IRUF(1)=IDATE  TIMER 44
  IRUF(2)=BLANK  TIMER 45
  IRUF(3)=NAME1  TIMER 46
  IRUF(4)=NAME2  TIMER 47
  IRUF(5)=10H  FLIGHT  TIMER 48
  IRUF(6)=10H  CONDITION  TIMER 49
  IF(NPLOT.EQ.0.OR.NPLOT.EQ.2) GO TO 2  TIMER 50
  CALL FACTOR(2,4)  TIMER 51
  CALL PLOT(0.,-13.,-3)  TIMER 52
  CALL PLOT(0.,0.5,-3)  TIMER 53
  CALL SYMBOL(0.,0.,.14,IRUF(1),90.,40)  TIMER 54
  CALL SYMBOL(.5,0.,.14,IRUF(5),90.,20)  TIMER 55
  CALL SYMBOL(.5,3.,.14,IFLT,90.,10)  TIMER 56
  CALL SYMBOL(1.,0.,.14,3HRUN,9.,.3)  TIMER 57
  CALL NUMBER(1.,.6.,.14, RUN,90.,-1)  TIMER 58
  CALL PLOT(1.5,0.,-3)  TIMER 59
  IF(NPLOT.EQ.1) GO TO 1  TIMER 60
  2 WRITE(9,100) (IRUF(I),I=1,6),IFLT,IRUN  TIMER 61
100 FORMAT(2X,6A10,2X,A10,2X,3HRUN,13//)  TIMER 62
  1 CONTINUE  TIMER 63
  DO 3 I=1,NX  TIMER 64

```

Figure 89. Subroutine TIMER Program Listing

# Contrails

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(NSTEP,EO,4) GO TO 7	TIMER 69
IF(NRAND,EO,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(DT)	TIMER 76
GO TO 1A	TIMER 77
4 NFG=NF	TIMER 78
10 IF(NSTEP,EO,2) GO TO 6	TIMER 79
IF(NSTEP,EO,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(NSTEP,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,EO,0) NTP=1	TIMER100
NTP=NTP/NTP	TIMER101
NTP=NTP+1	TIMER102
NPTOT=(	TIMER103
DO 12 J=1,NN	TIMER104
IF(NSTEP,EO,4) GO TO 51	TIMER105
IF(J,GT,NG) GO TO 41	TIMER106
IF(NRAND,EO,0,AND,NSTEP,EO,1) GO TO 12	TIMER107
GO TO 51	TIMER108
41 IF(NSTEP,EO,0,OR,NSTEP,EO,2) GO TO 12	TIMER109
51 DO 11 I=1,NX	TIMER110
X(I)=X1(I,J)	TIMER111
11 DX1(I)=.	TIMER112
IF(J,LE,NG) GO TO 56	TIMER113
IF(NSTEP,EO,0) GO TO 56	TIMER114
IF(NSTEP,EO,2) GO TO 56	TIMER115
IF(NSTEP,EO,4) GO TO 56	TIMER116
JJ=J+JX-NC-NG	TIMER117
X(JJ)=CL(J,1)	TIMER118
56 CONTINUE	TIMER119
IF(NP*IN,EO,0) GO TO 24	TIMER120
WRITE(9,103) J	TIMER121
103 FORMAT(141/7X, 3PHTIME RESPONSES FOR DISTURBANCE,13//)	TIMER122
24 DO 17 IT=1,NT	TIMER123
IF(J,GT,NG) GO TO 13	TIMER124
IF(NSTEP,LE,1) GO TO 55	TIMER125
IF(NSTEP,GT,3) GO TO 55	TIMER126
CALL SGUST(A,GS,CL,X,DT,T1,T2,J,NFG,NG,IT,NX,MH)	TIMER127
55 CONTINUE	TIMER128
IF(NRAND,EO,0) GO TO 13	TIMER129
ETA=G*AN(1)	TIMER130

Figure 89. Subroutine TIMER Program Listing (Continued)

<pre> GO TO 14 13 ETA=0. 14 DO 15 I=1,NFG    DX(I)=GN(I,J)*ETA+GS(I,J)    DO 52 K=1,NFG 52  DX(I)=DX(I)+A(I,K)*X(K)    IF (XL(I,2).LE.0.) GO TO 15    IF (ABS(DX(I)).GT.XL(I,2)) DX(I)=SIGN(XL(I,2),DX(I)) 15  CONTINUE    DO 16 I=1,NFG    X(I)=X(I)+DT*(3.*DX(I)-DX1(I))/2.    DX1(I)=DX(I)    IF (XL(I,1).LE.0.) GO TO 16    IF (ABS(X(I)).GT.XL(I,1)) X(I)=SIGN(XL(I,1),X(I)) 16  CONTINUE    IF (NTP.EQ.1) GO TO 18    ITM=MOD(IT,NTP)    IF (ITM.NE.1) GO TO 17    ITT=(IT/NTP)+1 18  IF (NTP.EQ.1) ITT=IT    II=ITT-NTTP+2    IP=1    DO 19 I=1,NR    IF (IP.GT.NOP) GO TO 23    IF (IPLR(IP).NE.I) GO TO 19    II=II+NTTP+2    R(II)=I.    DO 20 K=1,NX 20  R(II)=R(II)+HDK(I,K)*X(K)    R(II)=R(II)*SCAL(IP)    IP=IP+1 19  CONTINUE    DO 21 I=1,NFG    IX=I+NR    IF (IP.GT.NOP) GO TO 23    IF (IPLR(IP).NE.IX) GO TO 21    II=II+NTTP+2    R(II)=X(I)*SCAL(IP)    IP=IP+1 21  CONTINUE 23  II=II+NTTP+2    TIME=ITT-1    TIME=TIME*ST    R(II)=TIME    IF (NPRIN.EQ.0) GO TO 17    WRITE(9,202) TIME 202 FORMAT(5X,6HTIME =,F10.3) 200 FORMAT(5X,4(A10.2H=,G8.2,2X))    IPR=ITT    K=-3    DO 26 INP=1,NOP,4    K=K+4    IPR1=IPR+NTTP+2    IPR2=IPR1+NTTP+2    IPR3=IPR2+NTTP+2    KK=K+7-NOP    IF (KK.LE.0) GO TO 54    IF (KK.EQ.3) WRITE(9,200) ITITL(K),R(IPR)    IF (KK.EQ.2) WRITE(9,200) ITITL(K),R(IPR),ITITL(K+1),R(IPR1)    IF (KK.EQ.1) WRITE(9,200) ITITL(K),R(IPR),ITITL(K+1),R(IPR1)    ,ITITL(K+2),R(IPR2)    GO TO 26 54  WRITE(9,200) ITITL(K),R(IPR),ITITL(K+1),R(IPR1),ITITL(K+2),    ,R(IPR2),ITITL(K+3),R(IPR3) 26  IPR=IPR3+NTTP+2    WRITE(9,201) </pre>	<pre> TIMER131 TIMER132 TIMER133 TIMER134 TIMER135 TIMER136 TIMER137 TIMER138 TIMER139 TIMER140 TIMER141 TIMER142 TIMER143 TIMER144 TIMER145 TIMER146 TIMER147 TIMER148 TIMER149 TIMER150 TIMER151 TIMER152 TIMER153 TIMER154 TIMER155 TIMER156 TIMER157 TIMER158 TIMER159 TIMER160 TIMER161 TIMER162 TIMER163 TIMER164 TIMER165 TIMER166 TIMER167 TIMER168 TIMER169 TIMER170 TIMER171 TIMER172 TIMER173 TIMER174 TIMER175 TIMER176 TIMER177 TIMER178 TIMER179 TIMER180 TIMER181 TIMER182 TIMER183 TIMER184 TIMER185 TIMER186 TIMER187 TIMER188 TIMER189 TIMER190 TIMER191 TIMER192 TIMER193 TIMER194 TIMER195 TIMER196 </pre>
--	--

Figure 89. Subroutine TIMER Program Listing (Continued)



# Contrails

```
201 FORMAT(1HC)
17 CONTINUE
  IF (NPLOT.EQ.0) GO TO 12
  IF (NPLOT.EQ.1) GO TO 30
  WRITE(9,103) J
  DO 32 K=1,NOP
  DO 31 I=1,15
31  IRUF(I)=BLANK
  IF (NE-Y(K).EQ.0) GO TO 33
  YMAX(K)=-1.E+20
  YMIN(K)= 1.E+20
  DO 34 L=1,NTTP
  I=(K-1)*(NTTP+1)+L
  YMAX(K)=AMAX1(YMAX(K),R(I))
34  YMIN(K)=AMIN1(YMIN(K),R(I))
33 CONTINUE
  IF (YMIN(K).EQ.YMAX(K)) GO TO 32
  RANGE=YMAX(K)-YMIN(K)
  IRUF(3)=ITITL(K)
  IRUF(5)=IUNIT(K)
  WRITE(9,104) (IRUF(I),I=1,5)
104 FORMAT(1H1/5A10//)
  IRUF(3)=BLANK
  IRUF(5)=BLANK
  X(1)=YMIN(K)
  DO 36 I=1,5
36  X(I+1)=X(I)+RANGE/5.
  WRITE(9,105) (X(I),I=1,6)
105 FORMAT(6F14.3)
  WRITE(9,106)
106 FORMAT(7X,2H I,5(14H-----I))
  DO 37 L=1,NTTP
  I=(K-1)*(NTTP+1)+L
  IF (RANGE.EQ.0.) GO TO 42
  LL=(R(I)-YMIN(K))*70./RANGE
  IF (LL.EQ.0) GO TO 42
  INL=L/10
  IF (INL.LE.0) INL=0
  IF (INL.GT.11) INL=11
  IW=MOD(LL,10)
  IF (IW.LE.0) GO TO 40
  IF (INL.EQ.0) GO TO 53
  DO 38 I=1,INL
38  IRUF(I)=BLANK
  IRUF(INL+1)=LW(IW)
  IB=INL+2
  DO 39 I=18,15
39  IRUF(I)=BLANK
  GO TO 45
40  IW=10
  IW=10
  IE=INL-1
  DO 43 I=1,IE
43  IRUF(I)=BLANK
  IRUF(IE+1)=LW(IW)
  IB=IE+2
  DO 44 I=18,15
44  IRUF(I)=BLANK
  GO TO 45
42  IRUF(1)=LW(1)
  DO 46 I=2,15
46  IRUF(I)=BLANK
45  TIME=L-1
  TIME=TIME*ST
  WRITE(9,107) TIME,(IRUF(I),I=1,12)
107 FORMAT(F6.2,3HS I,12A10)
TIMER197
TIMER198
TIMER199
TIMER200
TIMER201
TIMER202
TIMER203
TIMER204
TIMER205
TIMER206
TIMER207
TIMER208
TIMER209
TIMER210
TIMER211
TIMER212
TIMER213
TIMER214
TIMER215
TIMER216
TIMER217
TIMER218
TIMER219
TIMER220
TIMER221
TIMER222
TIMER223
TIMER224
TIMER225
TIMER226
TIMER227
TIMER228
TIMER229
TIMER230
TIMER231
TIMER232
TIMER233
TIMER234
TIMER235
TIMER236
TIMER237
TIMER238
TIMER239
TIMER240
TIMER241
TIMER242
TIMER243
TIMER244
TIMER245
TIMER246
TIMER247
TIMER248
TIMER249
TIMER250
TIMER251
TIMER252
TIMER253
TIMER254
TIMER255
TIMER256
TIMER257
TIMER258
TIMER259
TIMER260
TIMER261
TIMER262
```

Figure 89. Subroutine TIMER Program Listing (Continued)

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

Figure 89. Subroutine TIMER Program Listing (Concluded)

# Contrails

```

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

Figure 90. Subroutine SGUST Program Listing

	SUBROUTINE CAL1(A,XN,P,KWA,N,NR,IMAX,IT,IERR,EE)	CAL1 2
	DIMENSION A(NR,1),XN(NR,1),P(NR,1),KWA(NR)	CAL1 3
	IERR=0	CAL1 4
	TR=0.	CAL1 5
	DO 300 I=1,N	CAL1 6
300	TR=TR+A(I,1)	CAL1 7
	FN=N	CAL1 8
	TR=AMAX1(TR,-FN)	CAL1 9
	IF(TR)301,2,2	CAL1 10
2	IERR=1	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,1)=P(I,1)-ALF	CAL1 23
60	CONTINUE	CAL1 24
	CALL TDINVR(ISOL,INSOL,N,N,P,NR,KWA,DET)	CAL1 25
	IF((ISOL+IDSOL).LE.2) GO TO 22	CAL1 26
	IERR=1	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,1)*XN(K,J)*2.*ALF	CAL1 33
	DO 5 I=1,N	CAL1 34
	DO 5 J=1,N	CAL1 35
	XN(I,1)=0.	CAL1 36
	DO 5 K=1,N	CAL1 37
5	XN(I,1)=XN(I,1)+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,1)=P(I,1)+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,1)*XN(K,J)	CAL1 49
	ICOT=	CAL1 50
	DO 10 I=1,N	CAL1 51
	DO 10 J=1,N	CAL1 52
	DXIJ=0.	CAL1 53
	DO 11 K=1,N	CAL1 54
11	DXIJ=DXIJ+A(I,K)*P(K,J)	CAL1 55
	XN(I,1)=XN(I,1)+DXIJ	CAL1 56
	XN(J,1)=XN(I,1)	CAL1 57
	AXN=ABS(XN(I,1))	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=2	CAL1 61
	GO TO 601	CAL1 62
201	RAT=ABS(DXIJ/XN(I,1))	CAL1 63
	IF(RAT-EE)14,14,70	CAL1 64

Figure 91. Subroutine CAL1 Program Listing

# Contrails

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)

# Contrails

```
SUBROUTINE STRIC(A,B,PS,W,S,TF,NX,NU,NXM,MU)          STRIC  2
DIMENSION A(NXM,NXM),R(NXM,NU),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=1000                                              STRIC 12
T=0.                                                 STRIC 13
DO 200 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 100 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,J)=0.                                          STRIC 27
DO 6 K=1,NX                                          STRIC 28
6 PS(I,J)=PS(I,J)-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 K=1,NU                                          STRIC 36
TPF(I,K)=0.                                          STRIC 37
DO 8 J=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(J,K)*DT              STRIC 43
200 CONTINUE                                        STRIC 44
WRITE(9,300)                                         STRIC 45
300 FORMAT(1H1/7X,12H W(T) MATRIX/)              STRIC 46
CALL MP(NXM,NXM,NX,NX,W)                            STRIC 47
RETURN                                              STRIC 48
END                                                  STRIC 49
```

Figure 92. Subroutine STRIC Program Listing

# Contrails

```
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
  DO 2 I=1,N                                       SHUFL  7
    2 B(I,J)=A(II,J)                               SHUFL  8
    DO 3 I=1,M                                     SHUFL  9
      DO 3 J=1,N                                   SHUFL 10
        3 A(I,J)=B(I,J)                           SHUFL 11
    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 J=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

# Contrails

```
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
IAKG (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



# Contrails

```

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,NM),HDK(MR,MX),C(MX,MX),RESP 4
1G2(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,MFA),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(1H)/7X,36H COVARIANCE ANALYSIS FOR DISTURBANCE,13//)RESP 22
ITER=iRESP 23
CALL COVAR(XI,A,C,X,G2,S,E,ES,V,U,NX,NFF,NN,MX,MFF,MFB,MN,IMAX,RESP 24
1ITER,IERR,KWA)RESP 25
IF(IERR.EQ.0) GO TO 896RESP 26
WRITE(9,43)RESP 27
43 FORMAT(1H)/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,X)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(1H)/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(1H)/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

# Contrails

DQ(I,J)=0.	RESP	65
DO 1117 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
DQD(I,J)=0.	RESP	70
DO 6085 K=1,NM	RESP	71
6085 DQD(I,J)=DQD(I,J)+DQ(I,K)*AK(J,K)	RESP	72
WRITE(9,45)	RESP	73
45 FORMAT(1H1/7X,26H CONTROL COVARIANCE MATRIX//)	RESP	74
CALL MP(MU,MU,NU,NU,DQD)	RESP	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 MATRIX//)	RESP	84
CALL MP(MX,MX,NX,NX,Z)	RESP	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(DQD(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,NX	RESP	96
1123 Z(I,J)=Z(I,J)+AKG(I,K)*X(K,J)	RESP	97
Z(I,J)=Z(I,J)/SQRT(DQD(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)	RESP	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,NX	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)

# Contrails

```

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(DQ(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,NM,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,47H 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,I3,E16.8))  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,I3,E16.8))  RESP 179
      RETURN                                  RESP 180
      END                                     RESP 181

```

Figure 95. Subroutine RESP Program Listing (Concluded)

# Contracts

```

SUBROUTINE COVAR(XI,A,C,X,G2,S,E,ES,V,U,NX,NFF,NN,MX,MFF,MFB,MN, COVAR 2
1IMAX,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),ES(MX,MX),U(MFB,MFB),V(MFF,MFF) COVAR 5
NFX=NX-NFF COVAR 6
IF(ITER.NE.0) GO TO 150 COVAR 7
C COVAR 8
C COVARIANCE CALCULATION 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+MFB COVAR 15
JJ=J+MFB 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,NFX COVAR 30
DO 1 L=1,NFX COVAR 31
1 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+MFB COVAR 36
152 C(I,J)=C(I,J)+A(I,KK)*XI(K,J) COVAR 37
DO 153 I=1,NFF COVAR 38
DO 153 J=1,NFF COVAR 39
II=I+MFB COVAR 40
JJ=J+MFB COVAR 41
153 S(I,J)=A(JJ,II) COVAR 42
CALL GCAL(S,X,C,NFF,NFX,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,NFX COVAR 47
DO 154 J=1,NFX COVAR 48
XI(I,J)=A(I,J) COVAR 49
S(I,J)=0. COVAR 50
DO 154 K=1,NFF COVAR 51
KK=K+MFB 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,NFX,MX,IMAX,2,IERR) COVAR 54
IF(IERR.GT.0) RETURN COVAR 55
DO 155 I=1,NFX COVAR 56
DO 155 J=1,NFF COVAR 57
JJ=J+MFB COVAR 58
X(I,J)=C(I,J) COVAR 59
155 X(JJ ,I)=X(I,JJ COVAR 60
DO 156 I=1,NFF COVAR 61
DO 156 J=1,NFF COVAR 62
II=I+MFB COVAR 63
JJ=J+MFB COVAR 64

```

Figure 96. Subroutine COVAR Program Listing

# Contrails

```
156 X(I1,J1)=X1(I,J)                                COVAR 65
      DO 157 I=1,NFR                                  COVAR 66
      DO 157 J=1,NFR                                  COVAR 67
157 X(I,J)=S(I,J)                                     COVAR 68
      RETURN                                          COVAR 69
      END                                             COVAR 70
```

Figure 96. Subroutine COVAR Program Listing (Concluded)

# Contrails

```
SUBROUTINE 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
 U(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=S12XA11+A22*S12+S11*A12+R12 COSTAT16
 DO 15R I=1,NFR COSTAT17
 DO 15H J=1,NFF COSTAT18
 JJ=J+NFB COSTAT19
 ES(I,J)=R(I,JJ) COSTAT20
 DO 15R K=1,NFR COSTAT21
 158 ES(I,J)=ES(I,J)+S(I,K)*A(K,JJ) COSTAT22
 DO 15H I=1,NFR COSTAT23
 DO 15H J=1,NFF COSTAT24
 159 X(I,J)=A(J,I) COSTAT25
 DO 16I I=1,NFF COSTAT26
 DO 16I 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 16I I=1,NFR COSTAT33
 DO 16I 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 16I I=1,NFF COSTAT40
 DO 16I 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 16I 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 16I I=1,NFF COSTAT50
 DO 16I 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

# Contrails

```
SUBROUTINE TRANS (AMT,X,T,DDD,NX,MX,NF,MU,MF,B,IF)      TRANS  2
DIMENSION AMT (MF,MX),X (MX,MX),T (MF,MF),DDD (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 (T,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)+DDD (L,M)*R (I,K)*AMT (J,K)      TRANS 15
RETURN      TRANS 16
END      TRANS 17
```

Figure 98. Subroutine TRANS Program Listing

# Contrails

```
SUBROUTINE UNSCR(T,DJDK,DJV,DJVT,IF,JF,NF,NU,NM,ML,MM,MF)      UNSCR  2
DIMENSION I(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.MF.IF(L)) GO TO 1                                        UNSCR  9
IF(J.MF.JF(L)) GO TO 1                                        UNSCR 10
DJV(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.MF.IF(L)) GO TO 5                                        UNSCR 22
IF(J.MF.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



# Contrails

SUBROUTINE GCAL(A,R,X,N,M,NN,MM,IMAX,E,ES,V,U,KWA,MX,IERR)	GCAL	2
C	GCAL	3
C THIS SUBROUTINE SOLVES THE GENERAL MATRIX EQUATION $XA+BX=C$	GCAL	4
C FORM (I-A),(I+B),(I-H),AND (I+R) THEN INVERT (I-A) AND (I-H)	GCAL	5
DIMENSION A(MX,MX),R(MX,MX),X(MX,MX),E(NN,NN)	GCAL	6
DIMENSION ES(MX,MX),V(NN,NN),H(MM,MM),KWA(MX)	GCAL	7
EE=.001	GCAL	8
IERR=0	GCAL	9
DO 1 I=1,N	GCAL	10
DO 2 J=1,N	GCAL	11
2 E(I,J) = -A(I,J)	GCAL	12
1 E(I,I)=E(I,I)+1.	GCAL	13
DO 3 I=1,M	GCAL	14
DO 4 J=1,M	GCAL	15
4 ES(I,J) = -B(I,J)	GCAL	16
3 ES(I,I)=ES(I,I)+1.	GCAL	17
NRR=N	GCAL	18
NCC=N	GCAL	19
NR=NN	GCAL	20
CALL TDINVR(ISOL,IDSOL,NRR,NCC,E,NR,KWA,DET)	GCAL	21
IF((ISOL+IDSOL).LE,2) GO TO 5	GCAL	22
IERR=4	GCAL	23
GO TO 6-1	GCAL	24
5 NR=MX	GCAL	25
CALL TDINVR(ISOL,IDSOL,M,M,ES,NR,KWA,DET)	GCAL	26
IF((ISOL+IDSOL).LE,2) GO TO 6	GCAL	27
IERR=5	GCAL	28
GO TO 6-1	GCAL	29
6 NC=N*1	GCAL	30
C	GCAL	31
C FORM U,V,AND W	GCAL	32
C	GCAL	33
DO 11 I=1,N	GCAL	34
DO 11 J=1,N	GCAL	35
V(I,J) = E(I,J)	GCAL	36
DO 11 K=1,N	GCAL	37
11 V(I,J) = V(I,J) + A(I,K)*E(K,J)	GCAL	38
DO 12 I=1,M	GCAL	39
DO 12 J=1,M	GCAL	40
U(I,J) = ES(I,J)	GCAL	41
DO 12 K=1,M	GCAL	42
12 U(I,J) = U(I,J)+ES(I,K)*B(K,J)	GCAL	43
DO 13 I=1,M	GCAL	44
DO 13 J=1,N	GCAL	45
B(I,J)=0.	GCAL	46
DO 14 K=1,M	GCAL	47
14 B(I,J) = B(I,J) + ES(I,K) * A(K,J)	GCAL	48
13 B(I,J)=2.*B(I,J)	GCAL	49
DO 15 I=1,M	GCAL	50
DO 15 J=1,N	GCAL	51
X(I,J) = 0.	GCAL	52
DO 15 K=1,N	GCAL	53
15 X(I,J) = X(I,J) + H(I,K) * E(K,J)	GCAL	54
ITER=	GCAL	55
100 CONTINUE	GCAL	56
DO 30 I=1,M	GCAL	57
DO 30 J=1,N	GCAL	58
R(I,J)=0.	GCAL	59
DO 30 K=1,M	GCAL	60
30 R(I,J) = B(I,J) + H(I,K)*X(K,J)	GCAL	61
C CONVERGENCE CHECK	GCAL	62
C	GCAL	63
ICOT=	GCAL	64

Figure 100. Subroutine GCAL Program Listing

# Contrails

```

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

Figure 100. Subroutine GCAL Program Listing (Concluded)

# Contrails

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=1	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 60 I=1,N	CAL	16
DO 63 J=1,N	CAL	17
GOTO(41,62),IT	CAL	18
61 P(I,J)=A(I,J)	CAL	19
GOTO 63	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,N,P,NR,KWA,DET)	CAL	25
IF((ISOL+IDSOL).LE.?) GO TO 22	CAL	26
IERR=?	CAL	27
GO TO 601	CAL	28
22 DO 4 I=1,N	CAL	29
DO 4 J=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)*?.*ALF	CAL	33
DO 5 I=1,N	CAL	34
DO 5 J=1,N	CAL	35
XN(I,I)=0.	CAL	36
DO 5 K=1,N	CAL	37
5 XN(I,I)=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)*?.*ALF	CAL	41
7 P(I,I)=P(I,I)+.	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,I)=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 601	CAL	62
201 RAT=ABS(DXIJ/XN(I,J))	CAL	63
IF(RAT-EE)14,14,70	CAL	64

Figure 101. Subroutine CAL Program Listing

# Contrails

14 ICOT=ICOT+1	CAL	65
70 CONTINUE	CAL	66
10 CONTINUE	CAL	67
18 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=I2)	CAL	85
RETURN	CAL	86
END	CAL	87

Figure 101. Subroutine CAL Program Listing (Concluded)

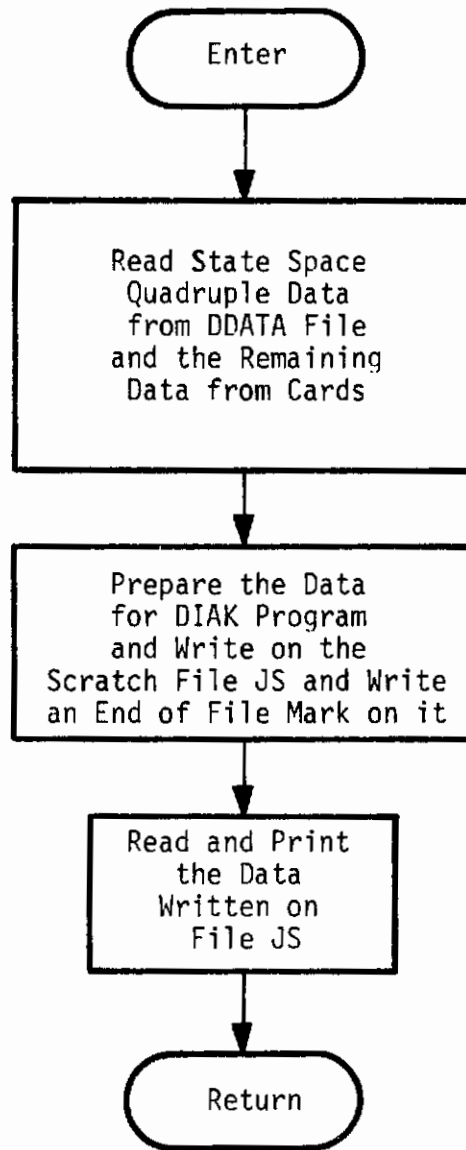


Figure 102. Subroutine DDIAK Flow Chart

# Contrails

```

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 MAHESH - THE MONEYWELL INC      DDIAK  5
C  DATE WRITTEN - 1975                                         DDIAK  6
C                                                                DDIAK  7
C  SUBPROGRAMS CALLED                                          DDIAK  8
C  ZEVO                                                         DDIAK  9
C  FILE                                                         DDIAK 10
C  MPDS                                                         DDIAK 11
C  ERDM                                                         DDIAK 12
C  WTD                                                         DDIAK 13
C  INPTM                                                        DDIAK 14
C                                                                DDIAK 15
C  ARGUMENTS LIST                                              DDIAK 16
C  A                   STATE TRANSITION MATRIX                DDIAK 17
C  B                   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  R2                  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,IW,IPRINT,INSERT,LOCATE, NULL, MARK(20)    DDIAK 36
C  I,JQ,JS,JSD,JF,JD                                           DDIAK 37
C  DATA HRFBH,HRG1R,HRG2R,HRHBR/4H F ,4H G1 ,4H G2 ,4H H /    DDIAK 38
C  DATA HRDBB,HRAMB,HC,HRCAR, HREAD/4H D ,4H AM ,1HC,4H CAR,4HREAD/ DDIAK 39
C  DATA HRTAP,HEND/4H TAP,4HEND /                               DDIAK 40
C  DATA HRBBB,HPEFR/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.HRHRB)GO TO B1                                 DDIAK 48
C  IF(CARD(6).NE.HPEFR)GO TO 162                                DDIAK 49
C  CALL ZERO(A,NXM,NXM)                                         DDIAK 50
C  CALL ZERO(R,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(IO)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,NR1,NR2,NR3,NU1,NU2,NU3 DDIAK 59
C                                                                DDIAK 60
C  PARTITION MATPICES 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

# Contrails

```

      IF (NR1.LE.0) STOP 111
      IF (NR3.LE.0) STOP 111
      DO 28 J=1,NX
      DO 24 J=1,NU1
24  R1(I,J)=R(I,J)
      DO 28 J=1,NU2
      JJ=NU1+J
28  R2(I,J)=R(I,JJ)
      DO 40 J=1,NX
      DO 34 J=1,NR1
34  C1(I,J)=C(I,J)
      DO 40 J=1,NR3
      II=NR1+NR2+1
40  C3(I,J)=C(II,J)
      DO 44 J=1,NR1
      DO 44 J=1,NU1
44  D11(I,J)=D(I,J)
      IF (IPRINT.LT.6) GO TO R4
      CALL MPRS(A,NXM,NXM,NX,NX,T,4HA )
      CALL MPRS(B,NXM,NUM,NX,NU1,T,4HB )
      CALL MPRS(C,NRM,NXM,NR,NX,T,4HC )
      CALL MPRS(D,NRM,NUM,NR,NU1,T,4HD )
      CALL MPRS(R1,NXM,NUM,NX,NU1,T,4HB1 )
      CALL MPRS(R2,NXM,NUM,NX,NU2,T,4HB2 )
      CALL MPRS(C1,NRM,NXM,NR1,NX,T,4HC1 )
      CALL MPRS(C3,NRM,NXM,NR3,NX,T,4HC3 )
      CALL MPRS(D11,NRM,NUM,NR1,NU1,T,4HD11 )
R0  CONTINUE
C
C  ORGANIZE CARD AND TAPE DATA ON TAPE
C
100 READ(IP,120) CARD
120 FORMAT(20A4)
      IF ((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HRTAP)) GO TO 160
      IF ((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HRCAR)) GO TO 100
      IF (CARD(1).EQ.HEND) GO TO 300
      WRITE(JS,120) CARD
      GO TO 100
160 CONTINUE
      IF (CARD(6).EQ.HRFRR) GO TO 180
      IF (CARD(6).EQ.HRG1R) GO TO 200
      IF (CARD(6).EQ.HRG2R) GO TO 220
      IF (CARD(6).EQ.HRHRR) GO TO 240
      IF (CARD(6).EQ.HRDRR) GO TO 260
      IF (CARD(6).EQ.HRAMR) GO TO 280
      IF (CARD(6).EQ.HRBKR) GO TO 235
162 CONTINUE
      WRITE(IW,165)
165 FORMAT(//IX,24HINPUT CONTROL CARD ERROR)
      CALL FRRM(1,4HODIA,4HK .3,0,IW)
C
C  WRITE MATRIX DATA ON SCRATCH FILE FOR DIJK PROGRAM
C
180 CONTINUE
      CALL WTP(A,NX,NX,NXM,NXM,JS)
      GO TO 100
200 CONTINUE
      CALL WTP(B1,NX,NU1,NXM,NUM,JS)
      GO TO 100
220 CONTINUE
      CALL WTP(B2,NX,NU2,NXM,NUM,JS)
      GO TO 100
240 CONTINUE
      CALL WTP(C1,NR,NX,NRM,NXM,JS)
      GO TO 100
260 CONTINUE

```

```

DDIAK 65
DDIAK 66
DDIAK 67
DDIAK 68
DDIAK 69
DDIAK 70
DDIAK 71
DDIAK 72
DDIAK 73
DDIAK 74
DDIAK 75
DDIAK 76
DDIAK 77
DDIAK 78
DDIAK 79
DDIAK 80
DDIAK 81
DDIAK 82
DDIAK 83
DDIAK 84
DDIAK 85
DDIAK 86
DDIAK 87
DDIAK 88
DDIAK 89
DDIAK 90
DDIAK 91
DDIAK 92
DDIAK 93
DDIAK 94
DDIAK 95
DDIAK 96
DDIAK 97
DDIAK 98
DDIAK 99
DDIAK100
DDIAK101
DDIAK102
DDIAK103
DDIAK104
DDIAK105
DDIAK106
DDIAK107
DDIAK108
DDIAK109
DDIAK110
DDIAK111
DDIAK112
DDIAK113
DDIAK114
DDIAK115
DDIAK116
DDIAK117
DDIAK118
DDIAK119
DDIAK120
DDIAK121
DDIAK122
DDIAK123
DDIAK124
DDIAK125
DDIAK126
DDIAK127
DDIAK128
DDIAK129
DDIAK130

```

Figure 103. Subroutine DDIAK Program Listing (Continued)

# Contrails

```
      CALL WTP(DI1,NR1,NU1,NRM,NUM,JS)
      GO TO 100
280  CONTINUE
      CALL WTP(C3,NR3,NX,NRM,NXM,JS)
      GO TO 100
285  CONTINUE
C
C  READ GAIN MATRIX FROM DDATA FILE
C
      READ(1R,20)HEAD
290  CONTINUE
      READ(JD,20)CARD
      DO 295 I=1,20
      IF(CARD(I).NE.HEAD(I))GO TO 290
295  CONTINUE
      CALL INPTM(BK,NUM,NRM,JD)
      REWIND JD
      CALL WTP(BK,NU1,NR3,NUM,NRM,JS)
      GO TO 100
300  CONTINUE
      END FILE JS
      REWIND JS
      IF((IPRINT.LT.5).AND.(IPRINT.NE.3)) GO TO 400
C
C  READ AND PRINT OUT TAPE
C
      WRITE(IW,310)
310  FORMAT(1H1,1X,23H*** DIAK INPUT DATA ***,//)
320  CONTINUE
      READ(JS,120)CARD
      IF (EOF(JS)) 360,340
340  WRITE(IW,350) CARD
350  FORMAT(1X,20A4)
      GO TO 320
360  CONTINUE
      REWIND JS
      WRITE(IW,380)
380  FORMAT(//,1X,30H*** END OF DIAK INPUT DATA ***)
400  CONTINUE
      RETURN
      END
```

DDIAK131  
DDIAK132  
DDIAK133  
DDIAK134  
DDIAK135  
DDIAK136  
DDIAK137  
DDIAK138  
DDIAK139  
DDIAK140  
DDIAK141  
DDIAK142  
DDIAK143  
DDIAK144  
DDIAK145  
DDIAK146  
DDIAK147  
DDIAK148  
DDIAK149  
DDIAK150  
DDIAK151  
DDIAK152  
DDIAK153  
DDIAK154  
DDIAK155  
DDIAK156  
DDIAK157  
DDIAK158  
DDIAK159  
DDIAK160  
DDIAK161  
DDIAK162  
DDIAK163  
DDIAK164  
DDIAK165  
DDIAK166  
DDIAK167  
DDIAK168  
DDIAK169  
DDIAK170  
DDIAK171

Figure 103. Subroutine DDIK Program Listing (Concluded)



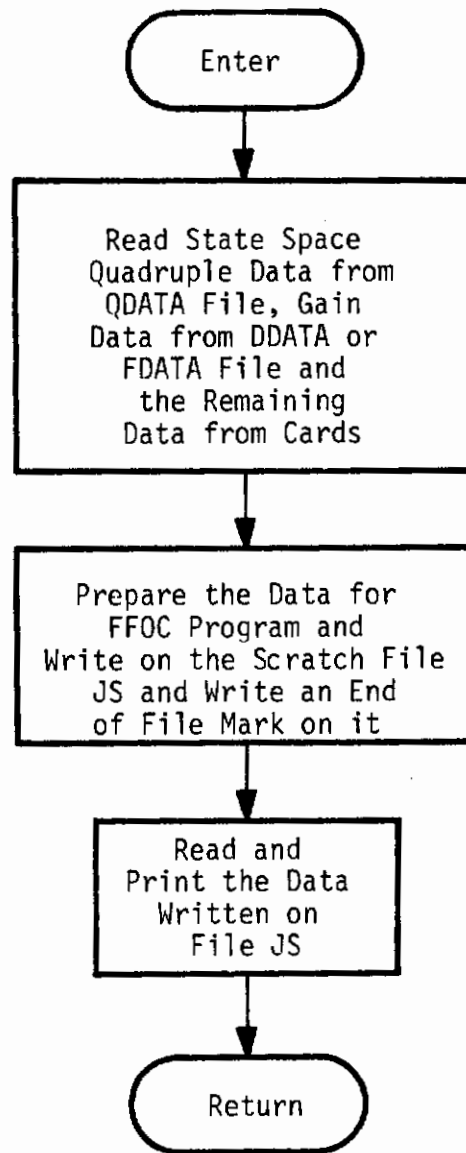


Figure 104. Subroutine DFFOC Flow Chart

# Contrails

```

C      SURROUTINE 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      SUBPROGRAMS CALLED                                             DFFOC  6
C      ZERO                                                            DFFOC  7
C      FILE                                                            DFFOC  8
C      MPDS                                                            DFFOC  9
C      WTD                                                             DFFOC 10
C      INPTH                                                           DFFOC 11
C      ARGUMENTS LIST                                               DFFOC 12
C      A                      STATE TRANSITION MATRIX                DFFOC 13
C      B                      CONTROL INPUT MATRIX                   DFFOC 14
C      C                      STATE OUTPUT MATRIX                    DFFOC 15
C      D                      CONTROL OUTPUT MATRIX                  DFFOC 16
C      B1                     INPUT MATRIX FOR CONTROL INPUTS - G1   DFFOC 17
C      B2                     INPUT MATRIX FOR GUST INPUTS - G2      DFFOC 18
C      C1                     STATE OUTPUT MATRIX FOR DESIGN OUTPUTS - H DFFOC 19
C      C3                     STATE OUTPUT MATRIX FOR MEASUREMENTS - M DFFOC 20
C      D11                    OUTPUT MATRIX FOR DESIGN OUTPUTS - D   DFFOC 21
C      RK                     FEEDBACK GAIN MATRIX                  DFFOC 22
C      NXM                    INPUT  MAXIMUM NO OF STATES           DFFOC 23
C      NRM                    INPUT  MAXIMUM NO OF OUTPUTS          DFFOC 24
C      NUM                    INPUT  MAXIMUM NO OF INPUTS           DFFOC 25
C      DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM)        DFFOC 26
C      DIMENSION B1(NXM,NUM),B2(NXM,NUM)                             DFFOC 27
C      DIMENSION C1(NRM,NXM),C3(NRM,NXM)                             DFFOC 28
C      DIMENSION D11(NRM,NUM),RK(NUM,NRM)                            DFFOC 29
C      DIMENSION HEAD(20),CARD(20)                                   DFFOC 30
C      COMMON/INOUT/IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20)        DFFOC 31
C      I,JQ,J5,JSU,JF,JD                                             DFFOC 32
C      DATA HRFBH,HBGJR,HGZRH,HBHBR/4H F ,4H G1 ,4H G2 ,4H H /      DFFOC 33
C      DATA HRDBB,HRAMR,HC,HRCAR,HREAD/4H D ,4H AM ,JHC,4H CAR,4HREAD/ DFFOC 34
C      DATA HRTAP,HEND/4H TAP,4HEND /                                DFFOC 35
C      DATA HRBBB,HPERR/4H ,4HPE /                                    DFFOC 36
C      DATA HRAKG,HRAKP,HRDEL/4H AKG,4H AK(,4H DEL/                 DFFOC 37
C      READ IF DATA IS ON CARDS ONLY                                 DFFOC 38
C      READ(IR,20)CARD                                               DFFOC 39
C      IF(CARD(6).EQ.HRFBH) GO TO 40                                  DFFOC 40
C      IF(CARD(6).NE.HPEERR) GO TO 162                                DFFOC 41
C      CALL ZERO(A,NXM,NXM)                                           DFFOC 42
C      CALL ZERO(B,NXM,NUM)                                            DFFOC 43
C      CALL ZERO(C,NRM,NXM)                                            DFFOC 44
C      CALL ZERO(D,NRM,NUM)                                            DFFOC 45
C      CALL ZERO(RK,NUM,NRM)                                           DFFOC 46
C      READ(IR,20)HEAD                                                DFFOC 47
C      20 FORMAT(20A4)                                                DFFOC 48
C      CALL FILE(JQ,LOCATE,HEAD)                                       DFFOC 49
C      READ(JQ)T,NX,NR,NU,((A(I,J),I=1,NX),J=1,NX),                 DFFOC 50
C      1((B(I,J),I=1,NX),J=1,NU),((C(I,J),I=1,NR),J=1,NX),          DFFOC 51
C      2((D(I,J),I=1,NR),J=1,NU),NXA,NRA,NUA,NR1,NR2,NR3,NU1,NU2,NU3 DFFOC 52
C      PARTITION MATRICES B,C,D                                       DFFOC 53
C      IF(NU1.LE.0)STOP 111                                           DFFOC 54
C      IF(NU2.LE.0)STOP 111                                           DFFOC 55
C      IF(NR1.LE.0)STOP 111                                           DFFOC 56
C      IF(NR1.LE.0)STOP 111                                           DFFOC 57
C      IF(NR1.LE.0)STOP 111                                           DFFOC 58
C      IF(NR1.LE.0)STOP 111                                           DFFOC 59
C      IF(NR1.LE.0)STOP 111                                           DFFOC 60
C      IF(NR1.LE.0)STOP 111                                           DFFOC 61
C      IF(NR1.LE.0)STOP 111                                           DFFOC 62
C      IF(NR1.LE.0)STOP 111                                           DFFOC 63
C      IF(NR1.LE.0)STOP 111                                           DFFOC 64

```

Figure 105. Subroutine DFFOC Program Listing

# Contrails

```

      IF (NR1.LE.6) STOP 111
      DO 28 J=1,NX
      DO 24 J=1,NU1
24  R1(I,J)=B(I,J)
      DO 28 J=1,NU2
      JJ=NU1+J
28  R2(I,J)=R(I,JJ)
      DO 40 J=1,NX
      DO 34 I=1,NR1
34  C1(I,J)=C(I,J)
      DO 40 I=1,NR3
      II=NR1+NR2+I
40  C3(I,J)=C(II,J)
      DO 44 I=1,NR1
      DO 44 J=1,NU1
44  D11(I,J)=D(I,J)
      IF (IPRINT.LT.6) GO TO 80
      CALL MPRS(A,NXM,NXM,NX,NX,T,4HA )
      CALL MPRS(B,NXM,NUM,NX,NU,T,4HB )
      CALL MPRS(C,NRM,NXM,NR,NX,T,4HC )
      CALL MPRS(D,NRM,NUM,NR,NU,T,4HD )
      CALL MPRS(B1,NXM,NUM,NX,NU1,T,4HB1 )
      CALL MPRS(B2,NXM,NUM,NX,NU2,T,4HB2 )
      CALL MPRS(C1,NRM,NXM,NR1,NX,T,4HC1 )
      CALL MPRS(C3,NRM,NXM,NR3,NX,T,4HC3 )
      CALL MPRS(D11,NRM,NUM,NR1,NU1,T,4HD11 )
80  CONTINUE
C
C  ORGANIZE CARD AND TAPE DATA ON TAPE
C
100 READ(IJ,120) CARD
120 FORMAT(20A4)
      IF ((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HRTAP)) GO TO 160
      IF ((CARD(1).EQ.HREAD).AND.(CARD(2).EQ.HBCAR)) GO TO 100
      IF (CARD(1).EQ.HEND) GO TO 300
      WRITE(JS,120) CARD
      GO TO 100
160 CONTINUE
      IF (CARD(6).EQ.HRFBR) GO TO 180
      IF (CARD(6).EQ.HRG1R) GO TO 200
      IF (CARD(6).EQ.HRG2R) GO TO 220
      IF (CARD(6).EQ.HRHBR) GO TO 240
      IF (CARD(6).EQ.HRDRR) GO TO 260
      IF (CARD(6).EQ.HHAMR) GO TO 280
      IF (CARD(6).EQ.HRAKR) GO TO 285
      IF (CARD(6).EQ.HRAKP) GO TO 285
      IF (CARD(6).EQ.HRDEL) GO TO 285
162 CONTINUE
      WRITE(IW,165)
165 FORMAT(//IX,24HINPUT CONTROL CARD ERROR)
      STOP 111
C
C  WRITE MATRIX DATA ON SCRATCH FILE FOR FFOC PROGRAM
C
180 CONTINUE
      CALL MTP(A,NX,NX,NXM,NXM,JS)
      GO TO 100
200 CONTINUE
      CALL MTP(B1,NX,NU1,NXM,NUM,JS)
      GO TO 100
220 CONTINUE
      CALL MTP(B2,NX,NU2,NXM,NUM,JS)
      GO TO 100
240 CONTINUE
      CALL MTP(C1,NR1,NX,NRM,NXM,JS)
      GO TO 100
      DFFOC 65
      DFFOC 66
      DFFOC 67
      DFFOC 68
      DFFOC 69
      DFFOC 70
      DFFOC 71
      DFFOC 72
      DFFOC 73
      DFFOC 74
      DFFOC 75
      DFFOC 76
      DFFOC 77
      DFFOC 78
      DFFOC 79
      DFFOC 80
      DFFOC 81
      DFFOC 82
      DFFOC 83
      DFFOC 84
      DFFOC 85
      DFFOC 86
      DFFOC 87
      DFFOC 88
      DFFOC 89
      DFFOC 90
      DFFOC 91
      DFFOC 92
      DFFOC 93
      DFFOC 94
      DFFOC 95
      DFFOC 96
      DFFOC 97
      DFFOC 98
      DFFOC 99
      DFFOC100
      DFFOC101
      DFFOC102
      DFFOC103
      DFFOC104
      DFFOC105
      DFFOC106
      DFFOC107
      DFFOC108
      DFFOC109
      DFFOC110
      DFFOC111
      DFFOC112
      DFFOC113
      DFFOC114
      DFFOC115
      DFFOC116
      DFFOC117
      DFFOC118
      DFFOC119
      DFFOC120
      DFFOC121
      DFFOC122
      DFFOC123
      DFFOC124
      DFFOC125
      DFFOC126
      DFFOC127
      DFFOC128
      DFFOC129
      DFFOC130

```

Figure 105. Subroutine DFFOC Program Listing (Continued)

# Contrails

```
260 CONTINUE                                DFFOC131
    CALL WTP(D11,NR1,NU11,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.HRAKG)JDF=JD              DFFOC142
    READ(TR,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,NRM)                   DFFOC149
    CALL INPTM(BK,NUM,NRM,JDF)              DFFOC150
    REWIND JDF                               DFFOC151
    CALL WTP(BK,NU1,NR3,NUM,NRM,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 (EOF(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)

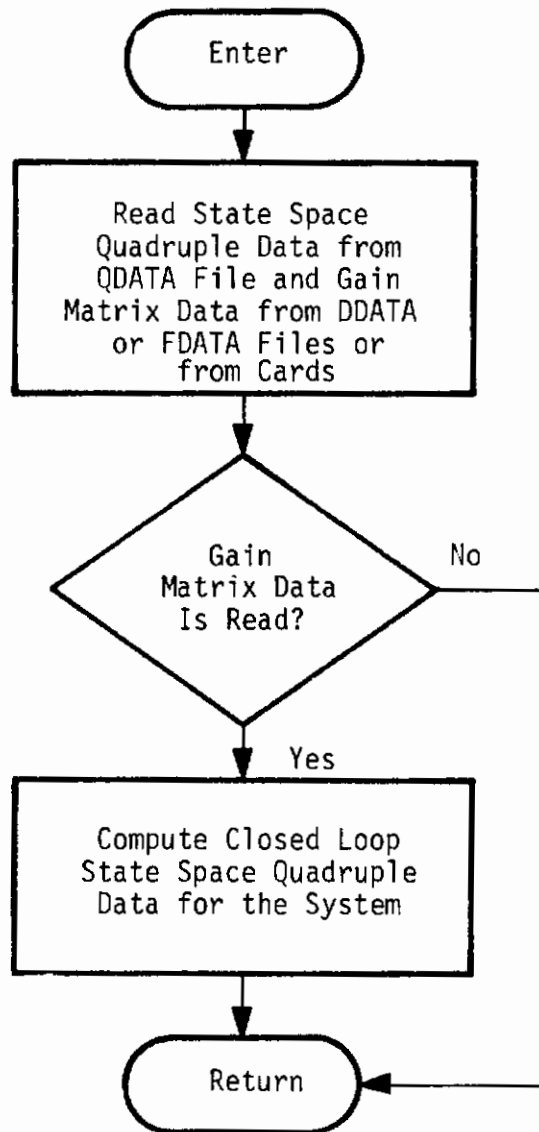


Figure 106. Subroutine DLSA Flow Chart

# Contrails

```

SUBROUTINE DLSA(A,R,C,D,B1,R2,C1,C3,D1),BK,BKC3,NX,NR,NU,
INXM,NRM,NUM)
C
C   PURPOSE - TO PREPARE DATA FOR LSA PROGRAM
C   ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C   DATE WRITTEN - 1975
C
C   SUBPROGRAMS CALLED
C       ZERO
C       FILE
C       MPDS
C       INPTM
C
C   ARGUMENTS LIST
C       A           STATE TRANSITION MATRIX
C       B           CONTROL INPUT MATRIX
C       C           STATE OUTPUT MATRIX
C       D           CONTROL OUTPUT MATRIX
C       H1          INPUT MATRIX FOR CONTROL INPUTS - G1
C       B2          INPUT MATRIX FOR GUST INPUTS - G2
C       C1          STATE OUTPUT MATRIX FOR DESIGN OUTPUTS - H
C       C3          STATE OUTPUT MATRIX FOR MEASUREMENTS - M
C       D1         OUTPUT MATRIX FOR DESIGN OUTPUTS - D
C       BK         FEEDBACK GAIN MATRIX
C       BKC3       BK*C3
C       NX         INPUT  MAXIMUM NO OF STATES
C       NR         INPUT  MAXIMUM NO OF OUTPUTS
C       NU         INPUT  MAXIMUM NO OF INPUTS
C
C   DIMENSION A(NXM,NXM),R(NXM,NUM),C(NRM,NXM),D(NRM,NUM)
C   DIMENSION B1(NXM,NUM),R2(NXM,NUM)
C   DIMENSION C1(NRM,NXM),C3(NRM,NXM)
C   DIMENSION D11(NRM,NUM),RK(NUM,NRM)
C   DIMENSION BKC3(NUM,NXM)
C   DIMENSION HEAD(20),CARD(20)
C   COMMON/INOUT/IR,IW,IPRINT,INSERT,LOCATE,NULL,MARK(20)
C   I,JQ,JS,JSD,JF,JD
C   DATA HBFBB,HBG1R,HG2R,HBBBR/4H F ,4H G1 ,4H G2 ,4H H /
C   DATA HBDBB,HRAKG,HC,HRCAR,HREAD/4H D ,4H AKG,HC,4H CAR,4HREAD/
C   DATA HRTAP,HEND/4H TAP,4HEND /
C   DATA HBBBR,HPEBR/4H ,4HPE /
C   DATA HRAKP/4H AKP/
C   IGAIN=0
C
C   READ QUADRUPLE DATA FROM QDATA FILE
C
C   READ(IR,20) CARD
C   IF(CARD(6).NE.HPEBR) GO TO 420
C   CALL ZERO(A,NXM,NXM)
C   CALL ZERO(B,NXM,NUM)
C   CALL ZERO(C,NRM,NXM)
C   CALL ZERO(D,NRM,NUM)
C   CALL ZERO(BK,NUM,NRM)
C   READ(IR,20) HEAD
C20 FORMAT(20A4)
C   CALL FILE(JQ,LOCATE,HEAD)
C   READ(IQ)T,NX,NR,NU,((A(I,J),I=1,NX),J=1,NX),
C   1((R(I,J),I=1,NX),J=1,NU),((C(I,J),I=1,NR),J=1,NX),
C   2((D(I,J),I=1,NP),J=1,NU),NXA,NRA,NUA,NR1,NR2,NR3,NU1,NU2,NU3
C
C   PARTITION MATRICES B,C,D
C
C   DO 28 I=1,NX
DLSA 2
DLSA 3
DLSA 4
DLSA 5
DLSA 6
DLSA 7
DLSA 8
DLSA 9
DLSA 10
DLSA 11
DLSA 12
DLSA 13
DLSA 14
DLSA 15
DLSA 16
DLSA 17
DLSA 18
DLSA 19
DLSA 20
DLSA 21
DLSA 22
DLSA 23
DLSA 24
DLSA 25
DLSA 26
DLSA 27
DLSA 28
DLSA 29
DLSA 30
DLSA 31
DLSA 32
DLSA 33
DLSA 34
DLSA 35
DLSA 36
DLSA 37
DLSA 38
DLSA 39
DLSA 40
DLSA 41
DLSA 42
DLSA 43
DLSA 44
DLSA 45
DLSA 46
DLSA 47
DLSA 48
DLSA 49
DLSA 50
DLSA 51
DLSA 52
DLSA 53
DLSA 54
DLSA 55
DLSA 56
DLSA 57
DLSA 58
DLSA 59
DLSA 60
DLSA 61
DLSA 62
DLSA 63
DLSA 64

```

Figure 107. Subroutine DLSA Program Listing

# Contrails

	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,NP1	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(B1,NXM,NUM,NX,NU1,T,4HH1 )	DLSA 84
	CALL MPRS(B2,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
C		DLSA 89
C	READ GAIN MATRIX DATA FROM ODATA 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.HBCAR)) GO TO 560	DLSA 97
420	CONTINUE	DLSA 98
	WRITE(IW,440)	DLSA 99
440	FORMAT(//,1X,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) CAPD	DLSA 112
	DO 520 I=1,20	DLSA 113
	IF(CAPD(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 400	DLSA 119
560	CONTINUE	DLSA 120
	CALL ZERO(RK,NUM,NRM)	DLSA 121
	CALL INPTM(RK,NUM,NRM,IR)	DLSA 122
	GO TO 400	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)

# Contrails

```
      RKC3(I,J)=0.0
      DO 70 K=1,NR3
70     RKC3(I,J)=RKC3(I,J)+RK(I,K)*C3(K,J)
      DO 80 I=1,NX
      DO 80 J=1,NX
      DO 80 K=1,NU1
80     A(I,J)=A(I,J)+B1(I,K)*RKC3(K,J)
      DO 90 I=1,NR1
      DO 90 J=1,NX
      DO 90 K=1,NU1
90     C(I,J)=C(I,J)+D11(I,K)*RKC3(K,J)
      DO 100 I=1,NX
      DO 100 J=1,NU2
100    R(I,J)=R2(I,J)
      NU=NU2
      IF(IP4)NT.LT.6) RETURN
      CALL MPRS(A,NXM,NXM,NX,NX,T,4HA )
      CALL MPRS(B,NXM,NUM,NX,NU,T,4HR )
      CALL MPRS(C,NRM,NXM,NR,NX,T,4HC , )
      CALL MPRS(D,NRM,NUM,NR,NU,T,4HD )
      CALL MPRS(BK,NUM,NXM,NU1,NX,T,4HBK )
      RETURN
      END
```

DLSA 131  
DLSA 132  
DLSA 133  
DLSA 134  
DLSA 135  
DLSA 136  
DLSA 137  
DLSA 138  
DLSA 139  
DLSA 140  
DLSA 141  
DLSA 142  
DLSA 143  
DLSA 144  
DLSA 145  
DLSA 146  
DLSA 147  
DLSA 148  
DLSA 149  
DLSA 150  
DLSA 151  
DLSA 152  
DLSA 153

Figure 107. Subroutine DLSA Program Listing (Concluded)



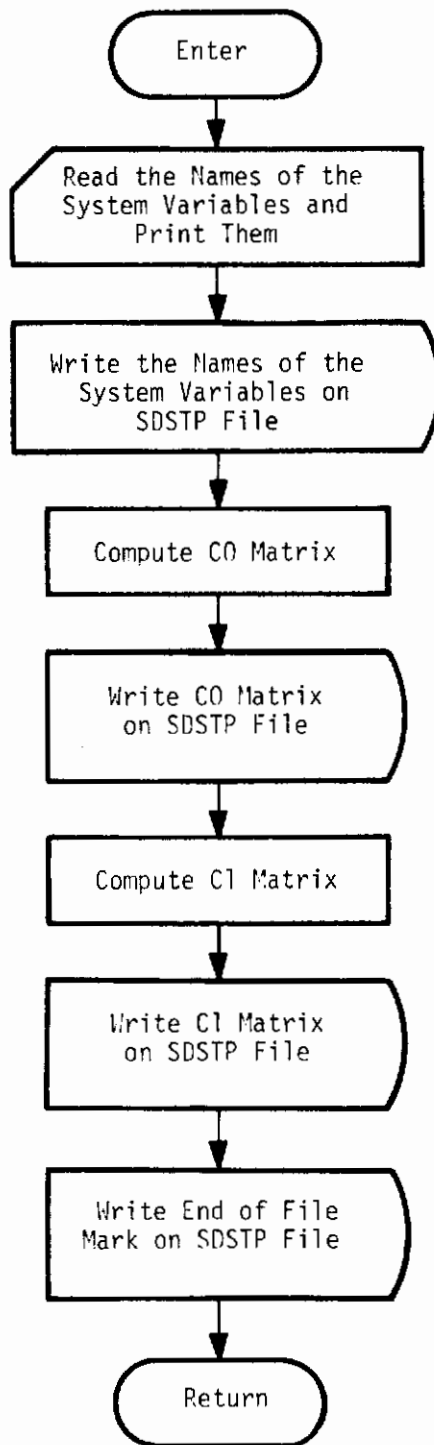


Figure 108. Subroutine FINK Flow Chart



# Contrails

```
DO 30 I=1,NX
DO 28 J=1,NX
280 CC(I,I)=-A(I,J)
DO 30 J=1,NU
JJ=NX+J
300 CC(I,JJ)=H(I,J)
DO 34 I=1,NR
II=NX+I
DO 32 J=1,NX
320 CC(II,J)=-C(I,J)
DO 34 J=1,NU
JJ=NX+J
340 CC(II,JJ)=D(I,J)
DO 36 I=1,NR
II=NX+I
360 CC(II,II)=1.0
CALL MPHS(CC,NXRM,NXRUM,NXR,NXRU,T,4HC0 )
WRITE(JSD)((CC(I,J),J=1,NXR),I=1,NR)
C
C COMPUTE C1 AND WRITE ON SDSTP FILE
C
CALL ZERO(CC,NXRM,NXRUM)
DO 26 I=1,NX
260 CC(I,I)=1.0
CALL MPHS(CC,NXRM,NXRUM,NXR,NXRU,T,4HC1 )
WRITE(JSD)((CC(I,J),J=1,NXR),I=1,NR)
C
C WRITE AN END OF FILE MARK ON SDSTP
C
ENDFILE JSD
GO TO 120
END
```

FINK	65
FINK	66
FINK	67
FINK	68
FINK	69
FINK	70
FINK	71
FINK	72
FINK	73
FINK	74
FINK	75
FINK	76
FINK	77
FINK	78
FINK	79
FINK	80
FINK	81
FINK	82
FINK	83
FINK	84
FINK	85
FINK	86
FINK	87
FINK	88
FINK	89
FINK	90
FINK	91
FINK	92
FINK	93
FINK	94
FINK	95
FINK	96

Figure 109. Subroutine FINK Program Listing (Concluded)

# Contrails

	SUBROUTINE MP(K,L,I,J;A)	MP	2
	DIMENSION A(K,L)	MP	3
	DO 1 II=1,I	MP	4
	WRITE(9,5)II	MP	5
	5 FORMAT(5H ROW 13)	MP	6
	1 WRITE(9,2) (A(II,JJ),JJ=1,J)	MP	7
2	FORMAT(2X,14E12.4)	MP	8
	RETURN	MP	9
	END	MP	10

Figure 110. Subroutine MP Program Listing

# Contrails

```
SUBROUTINE OUTP(I,I1,JJ,Y,I2)          OUTP  2
DIMENSION Y(I,I),YD(5),ID(5),JD(5)    OUTP  3
50 FORMAT(5(2I2,F12.5))                OUTP  4
III=0                                   OUTP  5
DO 100 K=1,I1                           OUTP  6
DO 100 M=1,JJ                             OUTP  7
IF(Y(I,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(I2,50)(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(I2,50)(ID(L),JD(L),YD(L),L=1,III)   OUTP 18
RETURN                                      OUTP 19
END                                         OUTP 20
```

Figure 111. Subroutine OUTP Program Listing

# Contrails

```
SUBROUTINE POLES(NX,A,MX,RP,M)
DIMENSION A(MX,1),RR(1)
CALL HESSEN(NX,A,MX)
CALL ORCALL(MX,A,RP,M,NX)
WRITE(9,6087)
6087 FORMAT(1H1/7X,1)METGENVALUES/12X,4HREAL,9X,9HIMAGINARY,8X,13HDAMPING
      RATIO,5X,9HFREQUENCY//)
MM=M/?
DO 6043 K=1,MM
  J=2*K-1
  OMEGA=SQRT(RR(1)*RR(1)+RR(I+1)*RR(I+1))
  IF(ABS(RR(I+1)).GT.,.00000001) GO TO 1
  WRITE(9,6084) RP(I)
  GO TO 6083
1  DELTA=RR(1)/OMEGA
  WRITE(9,6084) RP(I),RR(I+1),DELTA,OMEGA
6083 CONTINUE
6084 FORMAT(8X,4F15,8)
RETURN
END
```

POLES 2  
POLES 3  
POLES 4  
POLES 5  
POLES 6  
POLES 7  
POLES 8  
POLES 9  
POLES 10  
POLES 11  
POLES 12  
POLES 13  
POLES 14  
POLES 15  
POLES 16  
POLES 17  
POLES 18  
POLES 19  
POLES 20  
POLES 21

Figure 112. Subroutine POLES Program Listing

# Contrails

```
SUBROUTINE HESSEN(I,A,D)
DIMENSION A(1)
INTEGER P,PM,PX,D
ID=D+1
NN=(N-1)*ID+1
KX=NN-ID-ID+1
PM=1
PX=N
DO 70 K=2,KX,1D
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=JK
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<)
IF(B.EQ.0.) GO TO 65
R=B*T
KM=K+D
JM=JK+D
50 AJM=A(JM)-R*A(KM)
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)+R*A(J)
IF(ABS(AP).LE.(.1E-9*ABS(A(P)))) AP=0.
A(P)=AP
60 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 (JK,LE,NK) GO TO 45  
    CONTINUE  
    RETURN  
    END
```

```
HESSEN65  
HESSEN66  
HESSEN67  
HESSEN68
```

Figure 113. Subroutine HESSEN Program Listing (Concluded)



# Contrails

```
SUBROUTINE QRCALL(D,A,R,M,NIN)
INTEGER D
DIMENSION A(D,1),R(1)
N = NIN
ANN = 1.
ACT = .1E-7
ITER = 0
M = 0
IF(N.LE.1) RETURN
IF(N.EQ.2) GO TO 25
15 DELTA=ACT*ABS(A(N,N))
ACC = ABS(A(N,N-1))
IF(ACC.EQ.0.) GO TO 16
IF(ACC.GT.DELTA) GO TO 25
IF(ITER.GT.25) GO TO 16
IF(ANN.GT.ACT) GO TO 25
16 M = M+2
R(M-1) = A(N,N)
R(M) = 0.
17 K = NIN-N+1
ITER = 0
N = N-1
20 IF(N.GT.2) GO TO 15
IF(N.EQ.2) GO TO 25
IF(N.EQ.1) GO TO 16
R(M+1)=ACT
RETURN
25 B = .5*(A(N-1,N-1)+A(N,N))
DAN=ABS(A(N,N)-A(N-1,N-1))
SAN=ABS(A(N,N))+ABS(A(N-1,N-1))
IF(DAN.LE.ACT*SAN) DAN=0.
DAN=DAN*DAN*.25
C=A(N,N-1)*A(N-1,N)
T=DAN+C
IF((C.LT.0.) .AND. (ABS(T).LE.ACT*DAN) ) T=0.
IF(ABS(T).LE.ACT) T=0.
C = SQRT(ABS(T))
IF(N.NE.2) GO TO 50
26 IF(T.GE.0.) GO TO 30
M = M+2
R(M-1) = B
R(M) = C
27 N = N-1
GO TO 17
30 M = M+2
R(M-1) = B+C
R(M) = 0.
K = NIN-N+1
M = M+2
R(M-1) = B-C
R(M) = 0.
GO TO 27
50 IF(T.GE.0.) GO TO 60
R(M+5) = B
R(M+6) = C
R(M+7) = B
R(M+8) = -C
GO TO 70
60 X = B+C
Y = B-C
R(M+6) = 0.
R(M+8) = 0.
R(M+5) = X
```

```
QRCALL 2
QRCALL 3
QRCALL 4
QRCALL 5
QRCALL 6
QRCALL 7
QRCALL 8
QRCALL 9
QRCALL10
QRCALL11
QRCALL12
QRCALL13
QRCALL14
QRCALL15
QRCALL16
QRCALL17
QRCALL18
QRCALL19
QRCALL20
QRCALL21
QRCALL22
QRCALL23
QRCALL24
QRCALL25
QRCALL26
QRCALL27
QRCALL28
QRCALL29
QRCALL30
QRCALL31
QRCALL32
QRCALL33
QRCALL34
QRCALL35
QRCALL36
QRCALL37
QRCALL38
QRCALL39
QRCALL40
QRCALL41
QRCALL42
QRCALL43
QRCALL44
QRCALL45
QRCALL46
QRCALL47
QRCALL48
QRCALL49
QRCALL50
QRCALL51
QRCALL52
QRCALL53
QRCALL54
QRCALL55
QRCALL56
QRCALL57
QRCALL58
QRCALL59
QRCALL60
QRCALL61
QRCALL62
QRCALL63
QRCALL64
```

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 (ITER.LE.0) GO TO 130	QRCALL69
	X = ABS(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 = ABS(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 QR(N,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	QRCALL96
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 EIGENVALUES NOT FOUND)	QRCALL108
	RETURN	QRCALL109
	END	QRCALL110

Figure 114. Subroutine QRCALL Program Listing (Concluded)

# Contrails

	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 = 10*(N-1)+1	QR	8
	N1 = 90-ID	QR	9
	N2 = 81-ID	QR	10
	N3 = 72-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 = N+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(I1)+RHO	QR	28
	G2 = A(I+1)*(A(I)+A(I1+1)-SIGMA)	QR	29
	G3 = A(I+1)*A(I1+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 56	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

# Contrails

K = J+D	QR	65
ETA = A(J)+P1*A(K)	QR	66
L = K+D	QR	67
IF(I.LE.N2) ETA = ETA+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 63	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)-P2*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)

# Contracts

```

C      SUBROUTINE INPTM(A,II,JJ,IR)
C      PURPOSE - TO READ NONZERO ELEMENTS OF A MATRIX FROM FILE IR
C      ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C      DATE WRITTEN - 1975
C
C      ARGUMENTS LIST
C      A      INPUT      MATRIX DATA
C      II     INPUT      MAXIMUM NO OF ROWS
C      JJ     INPUT      MAXIMUM NO OF COLUMNS
C      IR     INPUT      FILE NO FOR READING MATRIX DATA
C
C      DIMENSION A(II,JJ),ID(5),JD(5),YD(5)
C      2 FORMAT (5(2I2,E12.5))
C      1 READ(IR,2)(ID(L),JD(L),YD(L),L=1,5)
C      IF(EOF(IR))10,6
C      6 CONTINUE
C      IF(ID(1))3,10,3
C      3 DO 5 L=1,5
C      IF (ID(L))4,1,4
C      4 I=ID(L)
C      J=JD(L)
C      5 A(I,J)=YD(L)
C      GO TO 1
C      10 CONTINUE
C      RETURN
C      END

```

INPTM	2
INPTM	3
INPTM	4
INPTM	5
INPTM	6
INPTM	7
INPTM	8
INPTM	9
INPTM	10
INPTM	11
INPTM	12
INPTM	13
INPTM	14
INPTM	15
INPTM	16
INPTM	17
INPTM	18
INPTM	19
INPTM	20
INPTM	21
INPTM	22
INPTM	23
INPTM	24
INPTM	25
INPTM	26
INPTM	27
INPTM	28

Figure 116. Subroutine INPTM Program Listing

# Contracts

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

**Figure 117. Subroutine WTP Program Listing**

# Contracts

	FUNCTION GRAN(N)	GRAN	2
	X=N	GRAN	3
	IF (N .EQ. 0) GO TO 1	GRAN	4
	ISEED = 33973679892	GRAN	5
	X=ISEED	GRAN	6
	TEM=RANF(X)	GRAN	7
	X=0.	GRAN	8
1	TFM = 0.0	GRAN	9
	DO 2 I = 1,12	GRAN	10
2	TEM=TFM+RANF(X)	GRAN	11
	TFM = TEM - 6.0	GRAN	12
	GRAN = TEM	GRAN	13
	RETURN	GRAN	14
	END	GRAN	15

Figure 118. Subroutine GRAN Program Listing

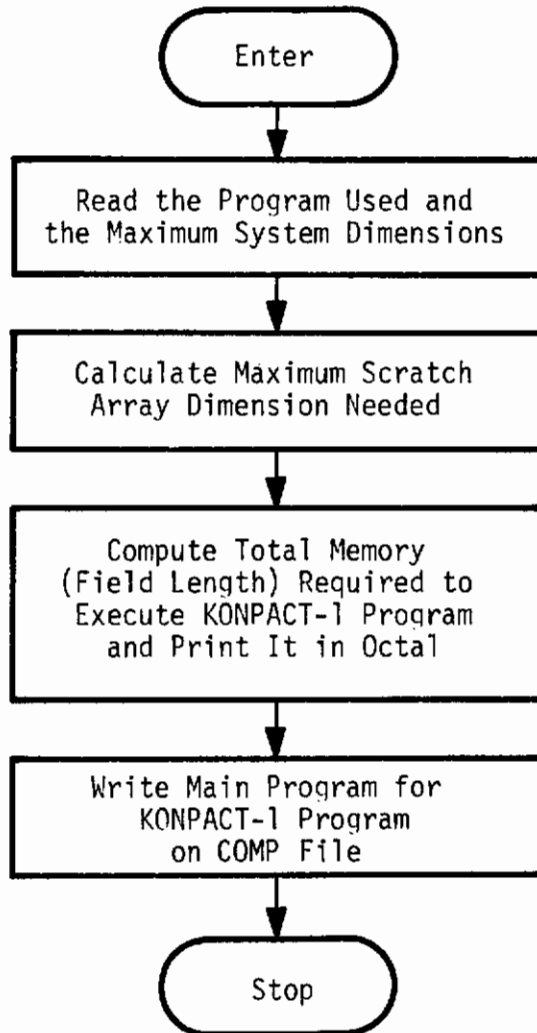


Figure 119. Program PRECOM Flow Chart



## 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.

# Contracts

```

PROGRAM PRECOM(INPUT,OUTPUT,COMP,TAPES=INPUT,TAPE9=OUTPUT
1,TAPE6=COMP)
C
C ANALYSIS - A F KONAR / J K MAHESH - THE HONEYWELL INC
C PURPOSE - TO READ THE PROGRAMS USED AND THE MAXIMUM SYSTEM
C DIMENSIONS AND SET UP THE MAIN PROGRAM FOR KONPACT=1 PROGRAMS
C DATE WRITTEN - DECEMBER 1975
C
C DIMENSION CARD(20)
DATA HNAME,HNRME,HNUME,HNYME/4HNAM=,4HNRM=,4HNUM=,4HNYM=
DATA MMSBE,MMTHE,MCH,MKBBB/4MMSB=,4MMTB=,2HC,4HK /
DATA MK1HB,MK2BB,MK3BB,MK4BB/4HK1,4HK2,4HK3,4HK4 /
MS1F=0 $ MS2F=0 $ MS3F=0 $ MS4F=0 $ MS5F=0
C
C INITIALIZE MAXIMUM SYSTEM DIMENSIONS
C
C NXM=0 $ NHM=0 $ NUM=0 $ NYM=0 $ MSB=0 $ MTR=0
C
C READ THE PROGRAMS USED AND THE MAXIMUM SYSTEM DIMENSIONS
C
100 CONTINUE
READ(5,120)CARD
120 FORMAT(20A4)
IF(EOF(5))220,140
140 CONTINUE
DECODE(4,160,CARD(1))CC,DUMMY
160 FORMAT(A2,A2)
IF(CC.EQ.MCH)GO TO 100
C
C SET THE PROGRAM FLAGS
C
CODE=CARD(2)
IF(CODE.EQ.MK1HB)MS1F=1
IF(CODE.EQ.MK1HB)GO TO 100
IF(CODE.EQ.MK2BB)MS2F=1
IF(CODE.EQ.MK2BB)GO TO 100
IF(CODE.EQ.MK3BB)MS3F=1
IF(CODE.EQ.MK3BB)GO TO 100
IF(CODE.EQ.MK4BB)MS4F=1
IF(CODE.EQ.MK4BB)GO TO 100
IF(CODE.EQ.MKBBB)MS5F=1
IF(CODE.EQ.MKBBB)GO TO 100
C
C SET THE MAXIMUM SYSTEM DIMENSIONS
C
CODE=CARD(1)
DECODE(4,180,CARD(2))MAX,DUMMY
180 FORMAT(I3,A1)
IF(CODE.EQ.HNAME)NXM=MAX
IF(CODE.EQ.HNAME)GO TO 100
IF(CODE.EQ.HNRME)NHM=MAX
IF(CODE.EQ.HNRME)GO TO 100
IF(CODE.EQ.HNUME)NUM=MAX
IF(CODE.EQ.HNUME)GO TO 100
IF(CODE.EQ.HNYME)NYM=MAX
IF(CODE.EQ.HNYME)GO TO 100
IF(CODE.EQ.MMSBE)MSB=MAX
IF(CODE.EQ.MMSBE)GO TO 100
IF(CODE.EQ.MMTHE)MTR=MAX
IF(CODE.EQ.MMTHE)GO TO 100
C
C IF DATA CARD IS IN ERROR PRINT ERROR MESSAGE
C
PRECOM 2
PRECOM 3
PRECOM 4
PRECOM 5
PRECOM 6
PRECOM 7
PRECOM 8
PRECOM 9
PRECOM10
PRECOM11
PRECOM12
PRECOM13
PRECOM14
PRECOM15
PRECOM16
PRECOM17
PRECOM18
PRECOM19
PRECOM20
PRECOM21
PRECOM22
PRECOM23
PRECOM24
PRECOM25
PRECOM26
PRECOM27
PRECOM28
PRECOM29
PRECOM30
PRECOM31
PRECOM32
PRECOM33
PRECOM34
PRECOM35
PRECOM36
PRECOM37
PRECOM38
PRECOM39
PRECOM40
PRECOM41
PRECOM42
PRECOM43
PRECOM44
PRECOM45
PRECOM46
PRECOM47
PRECOM48
PRECOM49
PRECOM50
PRECOM51
PRECOM52
PRECOM53
PRECOM54
PRECOM55
PRECOM56
PRECOM57
PRECOM58
PRECOM59
PRECOM60
PRECOM61
PRECOM62
PRECOM63
PRECOM64

```

Figure 120. Program PRECOM Program Listing

# Contrails

```
WRITE(9,200)CARD
200 FORMAT(1H1,/,/,1X,25HEXOR IN PRECOMPILER DATA,/,/,1X
1,18HLAST CARD READ WAS,/,/,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=NXRM+NYM
M=ORD=17 $ NRSM=1
NDM11=MAX0(MWORD,NXM,NRM,NRSM)
NDM12=MAX0(NXUM,NRM)
NDM21=MAX0(NRM,NXM,NRSM)
NDM22=MAX0(NXM,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+MSM*(1+NXRUM+3)
MS11=MAX0(MS111,MS112)
MS121=MS111+MTB*15
MS122=MS112
MS12=MAX0(MS121,MS122)
MS131=MS111+MSM*(3+NXRUM+NXM)+NRM*(MSB+1)
MS132=MS112
MS13=MAX0(MS131,MS132)
MS14=MS11
MS15=1+NXM*(NRUM+NRM)+NRM*(2*NRM+3*NUM)+3*NXRUM
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+NRM*NUM+MTB*(4B+MTB+NRUM)
MS22=MAX0(MS221,MS222)
MS231=MS211
MS232=1+NRM*NUM+MSM*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
```

Figure 120. Program PRECOM Program Listing (Continued)

# Contrails

```
IF ((MS1F,NE,0).OR.(MS2F,NE,0).OR.(MS3F,NE,0).OR.(MS4F,NE,0).OR. PRECO131
1(MS5F,NE,0))GO TO 221 PRECO132
MS1F=1 $ MS2F=1 $ MS3F=1 $ MS4F=1 $ MS5F=1 PRECO133
C PRECO134
C CALCULATE MAXIMUM SCRATCH ARRAY DIMENSIONS NEEDED PRECO135
C PRECO136
221 CONTINUE PRECO137
IF (MS1F,EQ,1)GO TO 222 PRECO138
MS11=1 $ MS21=1 $ MS31=1 PRECO139
222 CONTINUE PRECO140
IF (MS2F,EQ,1)GO TO 224 PRECO141
MS12=1 $ MS22=1 $ MS32=1 PRECO142
224 CONTINUE PRECO143
IF (MS3F,EQ,1)GO TO 226 PRECO144
MS13=1 $ MS23=1 $ MS33=1 PRECO145
226 CONTINUE PRECO146
IF (MS4F,EQ,1)GO TO 228 PRECO147
MS14=1 $ MS24=1 $ MS34=1 PRECO148
228 CONTINUE PRECO149
IF (MS5F,EQ,1)GO TO 230 PRECO150
MS15=1 $ MS25=1 $ MS35=1 PRECO151
230 CONTINUE PRECO152
MS1=MAX0(MS11,MS12,MS13,MS14,MS15) PRECO153
MS2=MAX0(MS21,MS22,MS23,MS24,MS25) PRECO154
MS3=MAX0(MS31,MS32,MS33,MS34,MS35) PRECO155
MS4=1 PRECO156
C PRECO157
C COMPUTE MEMORY REQUIRED FOR SCRATCH ARRAYS PRECO158
C PRECO159
MST=MS1+MS2+MS3+MS4 PRECO160
C PRECO161
C SET THE MEMORY REQUIRED FOR THE PROGRAM CODE PRECO162
C PRECO163
MPT=30000 PRECO164
C PRECO165
C COMPUTE TOTAL MEMORY REQUIRED TO EXECUTE KONPACT=1 PROGRAM AND PRECO166
C PRINT THE FIELD LENGTH REQUIRED IN OCTAL BASE PRECO167
C PRECO168
MT=MST+MPT PRECO169
WRITE(9,240) PRECO170
240 FORMAT(////) PRECO171
WRITE(9,240) PRECO172
260 FORMAT(/,10X,56(1#)) PRECO173
WRITE(9,280)MT PRECO174
280 FORMAT(/,10X,50#FIELD LENGTH REQUIRED FOR EXECUTING KONPACT=1 = PRECO175
1.0#) PRECO176
WRITE(9,260) PRECO177
C PRECO178
C WRITE MAIN PROGRAM FOR KONPACT=1 PROGRAM ON COMP FILE PRECO179
C PRECO180
WRITE(6,300) PRECO181
300 FORMAT(38H PROGRAM MAIN(8INPUT,INPUT,NDATA, PRECO182
1,40HDATA,OUTPUT,TAPE5=8INPUT, PRECO183
2,/,36H 1TAPE6=INPUT,TAPE7=NDATA,TAPE8= PRECO184
3,40HDATA,TAPE9=OUTPUT,VDATA, PRECO185
4,/,38H 2TAPE4=VDATA,SCRATCH,TAPE3=SCRATCH)) PRECO186
WRITE(6,320) PRECO187
320 FORMAT(39HC ANALYSIS - A F KONAR / J K MAHESH PRECO188
1,40H - THE HONEYWELL INC PRECO189
2,/,44HC PURPOSE - TO SET UP MAXIMUM DIMENSIONS) PRECO190
WRITE(6,330) PRECO191
330 FORMAT(34H COMMON /DIM/ MS1,MS2,MS3,MS4 PRECO192
1,26H,MAXN,MAXM,NXM,NRM,NUM,NYM, / PRECO193
2,42H 1,MM,MP,MQ,MR,MSB,NR,MS,MN,MR,MST,MT) PRECO194
WRITE(6,340)MS1,MS2,MS3,MS4 PRECO195
340 FORMAT(22H COMMON /SC1/ S1(/,5,1H), /, PRECO196
```

Figure 120. Program PRECOM Program Listing (Continued)

# Contrails

```
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)
360  FORMAT(38HC      MAXIMUM SCRATCH ARRAY DIMENSIONS)  PREC0200
      WRITE(6,380)MS1,MS2,MS3,MS4                    PREC0201
380  FORMAT(6X,4HMS1= ,15,7H $ MS2= ,15,7H $ MS3= ,15,7H $ MS4= ,15) PREC0202
      WRITE(6,400)
400  FORMAT(31HC      MAXIMUM SYSTEM DIMENSIONS)        PREC0203
      WRITE(6,420)NXM,NPM,NUM,NYM,MSH,MTB            PREC0204
420  FORMAT(6X,4HMXM= ,13,7H $ NPM= ,13,7H $ NUM= ,13,7H $ NYM=
1,13,7H $ MSH= ,13,7H $ MTR= ,13)                    PREC0205
      WRITE(6,440)
440  FORMAT(40HC      CALL KONPACT ORGANIZING SUBROUTINE) PREC0206
      WRITE(6,460)
460  FORMAT(16H      CALL KORG),/,25HC      STOP EXECUTION  PREC0207
1,/,10H      STOP,/,9H      END)                    PREC0208
      STOP                                           PREC0209
      END                                           PREC0210
                                           PREC0211
                                           PREC0212
                                           PREC0213
                                           PREC0214
                                           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.