

AD-A100 285

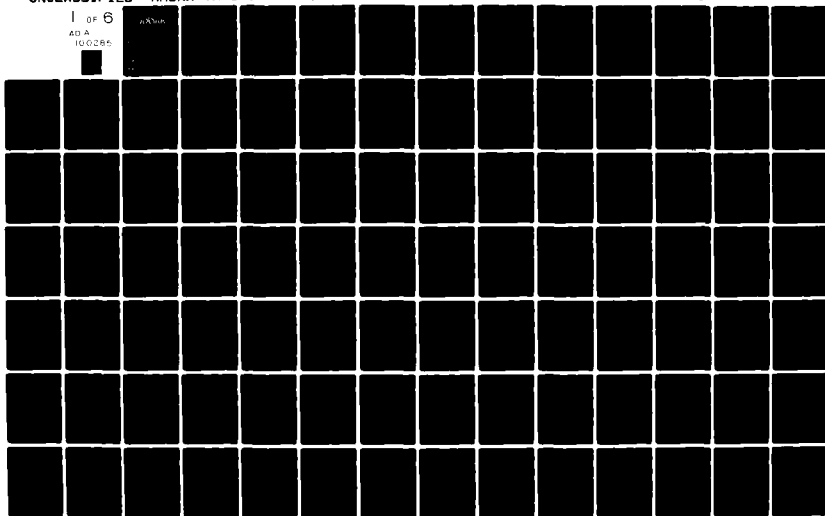
ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY ABERDEEN PROV--ETC F/G 19/1
COPPERHEAD OPERATIONAL PERFORMANCE EVALUATION (COPE): COMPUTER --ETC(U)
MAR 81 R S SANDMEYER

UNCLASSIFIED AMSAA-TR-318

NL

1 OF 6

AD A
100285





LEVEL II

AD

AMSAA

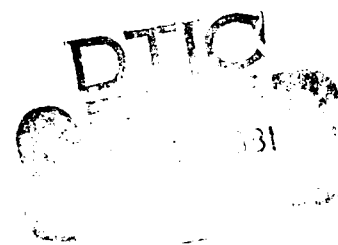
12

TECHNICAL REPORT NO. 318

AD A100285

COPPERHEAD OPERATIONAL PERFORMANCE EVALUATION (COPE):
COMPUTER PROGRAM USER AND ANALYST MANUAL

RICHARD S. SANDMEYER



MARCH 1981

DTIC FILE COPY

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

U. S. ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY
ABERDEEN PROVING GROUND, MARYLAND

81 6 15 054

DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position unless so specified by other official documentation.

WARNING

Information and data contained in this document are based on the input available at the time of preparation. The results may be subject to change and should not be construed as representing the DARCOM position unless so specified.

TRADE NAMES

The use of trade names in this report does not constitute an official endorsement or approval of the use of such commercial hardware or software. The report may not be cited for purposes of advertisement.

Accession For	
NTIS GMAI	X
DTIC 51P	
Unannounced	
Justification	
By	
Distribution	
Available	Codes
Dist	or
A	

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report No. 318	2. GOVT ACCESSION NO. AD-A100 285	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COPPERHEAD Operational Performance Evaluation (COPE): Computer Program User and Analyst Manual		5. TYPE OF REPORT & PERIOD COVERED
7. AUTHOR(s) Richard S. Sandmeyer		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Director US Army Materiel Systems Analysis Activity Aberdeen Proving Ground, MD 21005		8. CONTRACT OR GRANT NUMBER(s) AMSAH-TR-318
11. CONTROLLING OFFICE NAME AND ADDRESS Commander US Army Materiel Development & Readiness Command 5001 Eisenhower Ave., Alexandria, VA 22333		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DA Project # 1R765705M541
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE March 1981
		13. NUMBER OF PAGES 490
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release; Distribution Unlimited.		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
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) COPPERHEAD COPPERHEAD Operational Performance Evaluation Combat Operational Performance Evaluation COPE		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The COPE model simulates a COPPERHEAD fire mission under battlefield conditions. It takes into account such degrading effects on designator, fire direction center (FDC), artillery battery, and COPPERHEAD round performance as meteorological conditions, terrain shielding, smoke, dust, enemy artillery, preparatory fires, enemy direct fire against the designator, communication failures, and target location error as well as numerous factors that affect COPPERHEAD's ability to engage, hit, and kill a target.		

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

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

40291

19. Key Words - Continued

XM712	Operational Performance
Artillery	Laser Guided Weapons
Weapons Effectiveness	Smart Weapons
Item Level Performance	
Battlefield Environment	

20. Abstract - Continued

> This report includes detailed documentation of the computer programs that implement the COPE model. Sample cases complete with input and output as well as program source listings are included.

TABLE OF CONTENTS

	PAGE
PREFACE	17
NOTE ON CLASSIFICATION.	19
ACKNOWLEDGEMENTS.	21
<u>CHAPTER 1</u>	
1. INTRODUCTION.	1-1
1.1 COPE Background.	1-1
1.2 Overview of COPE Computer Programs	1-2
<u>CHAPTER 2</u>	
2. DESCRIPTION OF COPPERHEAD MISSION MODELING.	2-1
2.1 Application Area of Simulation	2-1
2.2 Description of Model	2-2
2.2.1 Outline of COPPERHEAD Mission	2-2
2.2.2 Preparatory Fires	2-4
2.2.3 Visibility Range.	2-7
2.2.4 Designator Maximum Range.	2-7
2.2.5 Target Not Detected	2-7
2.2.6 Smoke	2-8
2.2.7 Dust.	2-8
2.2.8 Designator Bail-Out (Precommo).	2-9
2.2.9 Commo-Out (Tests 8 and 9)	2-9
2.2.10 Designator Bail-Out After Communication	2-11
2.2.11 Line-of-Sight Lost Prior to Firing.	2-11
2.2.12 Line-of-Sight Lost During Mission	2-11
2.2.13 Designator Not Warned In Time	2-12
2.2.14 Designator in Direct Fire	2-12
2.2.15 Round Reliability Failures.	2-12
2.2.16 Target Obscured By Mini-Terrain	2-12
2.2.17 Round Did Not Engage Target	2-13
2.2.18 Round Engaged But Did Not Hit	2-13
2.2.19 Round Hit But Did Not Kill.	2-13
2.2.20 Subsequent Rounds	2-13
2.2.21 End of Replication.	2-14
2.2.22 Random Occurrence LOS Model Vs Shooting Gallery LOS Model	2-14
2.2.23 Note on Nominal Response Time	2-14

TABLE OF CONTENTS - Continued

	PAGE
 <u>CHAPTER 3</u>	
3. PERFORMANCE AND LIMITATIONS.	3-1
3.1 Computer Related Performance and Limitations.	3-1
3.2 Program Related Performance and Limitations	3-1
3.3 Model Related Performance and Limitations	3-2
3.3.1 Geometry of COPPERHEAD Fire Mission.	3-2
3.3.2 Target Modeling.	3-4
3.3.3 Direct Fire Effects Independent.	3-5
3.3.4 Only Initial Acquisitions (Unmaskings) Played	3-5
3.3.5 Other Limitations.	3-5
 <u>CHAPTER 4</u>	
4. INPUT DIRECTIVES AND "DATA" STATEMENTS	4-1
4.1 Input	4-1
4.2 Data Organization	4-1
4.3 Keyword Input: General Description.	4-1
4.4 Keyword Input: Specifics.	4-2
4.4.1 FIRST CASE	4-2
4.4.2 NEXT CASE.	4-3
4.4.3 END.	4-3
4.4.4 ENDF	4-3
4.4.5 PLOT	4-4
4.4.6 CONTROL.	4-5
4.4.7 LINE-OF-SIGHT MODE	4-5
4.4.8 MISSION CODE	4-6
4.4.9 NUMBER OF DIFFERENT TARGET POSTURES.	4-6
4.4.10 TARGET TYPE.	4-6
4.4.11 DESIGNATOR TYPE.	4-7
4.4.12 DO POSITION.	4-7
4.4.13 WEATHER.	4-8
4.4.14 ACQUISITION RANGE DISTRIBUTION	4-8
4.4.15 RESPONSE TIME.	4-9
4.4.16 DIRECT FIRE SUPPRESSION.	4-11
4.4.17 INVARIANT.	4-11
4.4.18 RUMC	4-11
4.4.19 SMOKE.	4-12
4.4.20 DO ARTILLERY PK.	4-13
4.4.21 TARGET VELOCITY.	4-13
4.4.22 MAXIMUM DESIGNATOR RANGE	4-13

TABLE OF CONTENTS - Continued

	PAGE
4.4.23 BAIL OUT RANGE.	4-14
4.4.24 TIME OF FLIGHT.	4-14
4.4.25 ANGLE T	4-14
4.4.26 DUST.	4-14
4.4.27 PROBABILITY OF SUCCESSFUL VOICE TRANSMISSION.	4-14
4.4.28 PROBABILITY OF SUCCESSFUL DIGITAL TRANSMISSION.	4-15
4.4.29 ROUND IN FLIGHT RELIABILITY	4-15
4.4.30 TIME BETWEEN ROUNDS	4-15
4.4.31 NUMBER OF REPLICATIONS.	4-15
4.4.32 NUMBER OF ROUNDS TO BE FIRED.	4-15
4.4.33 NUMBER OF VEHICLES PER TARGET	4-16
4.4.34 DISTANCE BETWEEN VEHICLES	4-16
4.4.35 GUN TARGET RANGE.	4-16
4.4.36 REFLECTIVITY.	4-16
4.4.37 DEFLECTION BIAS	4-17
4.4.38 TARGET HEADING.	4-17
4.4.39 SEEKER SENSITIVITY.	4-17
4.4.40 PROBABILITY OF A CORRECT MESSAGE.	4-18
4.4.41 PROBABILITY DO WARNED TO LASE	4-18
4.5 Special Keywords: OVERRIDE, TEMPORARY, RESET. . . .	4-18
4.5.1 Correspondence of Data Blocks to Keywords. .	4-18
4.5.2 OVERRIDE and TEMPORARY Features.	4-19
4.5.3 Using the OVERRIDE	4-19
4.5.4 Using TEMPORARY.	4-20
4.5.5 RESET.	4-21
4.6 Comments.	4-21
4.6.1 General Remarks on Comments.	4-21
4.6.2 \$ Comments	4-21
4.6.3 \$\$ Comments.	4-21
4.6.4 \$\$\$ Comments	4-22
4.7 DATA Statement Data	4-22
4.7.1 ABRLBL	4-22
4.7.2 ACHAR.	4-22
4.7.3 DESRNG	4-22
4.7.4 DISPLY	4-23
4.7.5 FLTTIM	4-23
4.7.6 FSRESP	4-23
4.7.7 HEADNG	4-23

TABLE OF CONTENTS - Continued

	PAGE
4.7.8 LOGFLG.	4-23
4.7.9 POINT	4-23
4.7.10 SMOKED.	4-23
4.7.11 STITLE.	4-23
4.7.12 SYMBOL.	4-23
4.7.13 XVALUE.	4-23
4.8 Side Effects of Certain Inputs	4-24
 <u>CHAPTER 5</u>	
5. PREPARING TO RUN COPE	5-1
5.1 Preprocessor Programs.	5-1
5.2 Word Addressable Mass Storage Data Base File (TAPE 11).	5-2
5.3 Record Names	5-2
5.3.1 Weather Data Record Name.	5-2
5.3.2 Acquisition Data Record Name.	5-3
5.3.3 Response Time Data Record Name.	5-3
5.3.4 Direct Fire Suppression Data Record Name.	5-3
5.3.5 Random Occurrence Data Record Name.	5-3
5.3.6 Target Posture Distribution Data Record Name.	5-4
5.3.7 Invariant Data Record Name.	5-4
5.3.8 Option Word Record Names.	5-4
5.3.9 Keyword Numbers, Abbreviated Forms, Etc.	5-6
5.4 PEDATA Record Names.	5-11
5.5 Control Cards for COPE and Its Preprocessors	5-13
 <u>CHAPTER 6</u>	
6. OUTPUT.	6-1
6.1 Title Page	6-1
6.2 "\$\$\$" Comment Page	6-1
6.3 Case Heading Page.	6-3
6.3.1 TARGET.	6-3
6.3.2 FIRING INFO	6-3
6.3.3 MISC.	6-6
6.3.4 DETECT, COMMO, and PROC TIMES	6-6
6.3.5 DESIGNATOR.	6-7
6.3.6 DESIGNATOR SUPPRESSED	6-7

TABLE OF CONTENTS - Continued

	PAGE
6.3.7 WEATHER	6-7
6.3.8 TERRAIN AND LOS	6-7
6.3.9 OBSCURANTS.	6-8
6.3.10 "\$\$" Comment Section.	6-8
6.4 Case Results Page.	6-8
<u>CHAPTER 7</u>	
7. PROGRAM GENERATED MESSAGES AND DEBUGGING PRINTS	7-1
7.1 Program Generated \$\$\$ Comments	7-1
7.2 Program Generated \$\$ Comments.	7-2
7.2.1 CHOICE OF Option Word FOR RESPONE TIME DATA IS INCONSISTENT WITH TIME OF Option Word FOR WEATHER DATA.	7-2
7.2.2 IN CCLOT FOR CASE nn WITH ICCPFG=1. NO PLOT PRODUCED.	7-3
7.2.3 IN PLOT FOR CASE nn WITH IPPFG=1. NO PLOT PRODUCED.	7-3
7.2.4 NO ENDF CARD FOUND BUT ONE ASSUMED SINCE END OF INPUT FILE ENCOUNTERED	7-3
7.2.5 NO 'FIRST CASE' OR 'NEXT CASE' CARD FOUND --- NEXT CASE, RD OPTION ASSUMED.	7-3
7.2.6 PEDATA FOR: TR=xxx. DT= n VEL= xx. GTR=zzz. REFL = a.aa ANGLET= bbb. DEFT= ccc. TGTID= dd. SKSEN=ee.	7-3
7.2.7 RECORD NAMED: Record Name USED TO OVERRIDE FOR Data Block Name	7-4
7.2.8 TEMPORARY OPTION IN EFFECT FOR Data Block Name.	7-4
7.2.9 YOUR INPUT LINE: Input Line Characters SUPERSEDES ONE OF YOUR PREVIOUS INPUT LINES FOR THIS CASE	7-4
7.3 Special Program Print Outs of a Warning Nature That Do Not Result in Program Termination	7-4
7.3.1 IN OUTPUT. PROBLEM 1 J= nnnnn RNDRED=xxxx NKILL = mmmmm	7-4
7.3.2 SERIOUS PROBLEM! KTEST AND NTEST ARRAYS DO NOT AGREE II= nnnnn JJ = Mmmm KTEST ARRAY: (Followed By A 20 x 6 Matrix)	7-5
7.4 Special Program Prints That Occur with Abnormal Program Termination and Which Give Information Useful in Locating the Cause of Termination.	7-4

TABLE OF CONTENTS - Continued

	PAGE
7.4.1 ERROR IN INPUT. LINE: Character String WITH IT1=nnnnn IS IN ERROR.	7-5
7.4.2 IN COMMNT. TOO MANY \$\$ COMMENTS (>20): (Followed By 20 Numbered Lines of Comments, Then:) THE FOLLOWING COMMENT CAUSED OVERFLOW OF CMENT ARRAY: (Followed By a 21st Comment Line)	7-5
7.4.3 INPUT ATTEMPTED TO READ RECORD NAMED Record name BUT NO SUCH RECORD WAS FOUND. . . .	7-6
7.4.4 IN FINDIT. KEYWORD: Keyword AND CORRESPONDING VALUE OF IT: Value ARE NOT ALLOWED WITH THE Option Word OPTION.	7-6
7.4.5 IN FINDIT. NO MATCH FOUND FOR LINE IB=nnn OF THIS B ARRAY: (Followed By Up to 10 Lines of Character Strings)	7-6
7.4.6 IN NUMRIC. LINE nnn EXCEEDS ALLOWABLE FIELD LENGTH FOR NUMERIC DATA: (Followed By Up to 10 Lines of Character Strings).	7-7
7.4.7 IN NUMRIC. LINE nnn HAD TOO MANY DECIMAL POINTS: (Followed By Up to 10 Lines of Charater Strings)	7-7
7.4.8 IN SMPLCD WITH RANDOM NUMBER GREATER THAN 1.0 RN=xx.xxxxxxx	7-7
7.5 STOP Messages That are Printed in the Dayfile.	7-7
7.5.1 STOP IN COMMNT: TOO MANY \$\$ COMMENTS.	7-7
7.5.2 STOP IN COPE: NORMAL PROGRAM TERMINATION. . . .	7-8
7.5.3 STOP IN DFCHK: ERROR NUMBER 1	7-8
7.5.4 STOP IN FINDIT: ERROR NUMBER 1.	7-8
7.5.5 STOP IN FINDIT: ERROR NUMBER 2.	7-8
7.5.6 STOP IN FINDIT: ERROR NUMBER 3.	7-8
7.5.7 STOP IN FINDIT: ERROR NUMBER 4.	7-8
7.5.8 STOP IN FPRCNT: ERROR NUMBER 1.	7-8
7.5.9 STOP IN GETRNG: TARGET VELOCITY=0.0	7-8
7.5.10 STOP IN GETVIS: RANDOM NUMBER TROUBLE 1	7-9
7.5.11 STOP IN GETVIS: RANDOM NUMBER TROUBLE 2	7-9
7.5.12 STOP IN HEADER: ERROR NUMBER 1.	7-9
7.5.13 STOP IN HITCHK: FAILED TO BRACKET RANGE	7-9
7.5.14 STOP IN INPUT: EOF ENCOUNTERED.	7-9
7.5.15 STOP IN INPUT: ERROR NUMBER 1	7-9
7.5.16 STOP IN INPUT: EXTRA NEXT CASE CARD	7-9
7.5.17 STOP IN INPUT: FIRST CASE CARD IN SUBSEQUENT CASE	7-9
7.5.18 STOP IN INPUT: PARAMETERIZED RESPONSE TIME <=0.0	7-10

TABLE OF CONTENTS - Continued

	PAGE
7.5.19 STOP IN LOSCHK: ERROR WITH RANDOM OCCURRENCE OPTION	7-10
7.5.20 STOP IN MINTRN: FAILED TO BRACKET RANGE	7-10
7.5.21 STOP IN NOREC: ATTEMPTED TO READ NON EXISTENT RECORD	7-10
7.5.22 STOP IN NUMRIC: ERROR NUMBER 1.	7-10
7.5.23 STOP IN NUMRIC: ERROR NUMBER 2.	7-10
7.5.24 STOP IN PECHK: FAILED TO BRACKET DELTA T.	7-10
7.5.25 STOP IN PENAME: ERROR NUMBER 1.	7-10
7.5.26 STOP IN PENAME: ERROR NUMBER 2.	7-10
7.5.27 STOP IN PENAME: ERROR NUMBER 3.	7-10
7.5.28 STOP IN PENAME (PEIDNT) ERROR NUMBER 4.	7-10
7.5.29 STOP IN SEPREC: ERROR NUMBER 1.	7-10
7.5.30 STOP IN SMPLCD: FAILED TO BRACKET RANDOM NUMBER	7-11
7.5.31 STOP IN SMPLCD: RANDOM NUMBER TOO LARGE	7-11
7.6 Echo Printing.	7-11
<u>CHAPTER 8</u>	
8. PROGRAM DESIGN.	8-1
8.1 Programming Principles of COPE	8-1
8.2 General Logic Flow	8-2
8.3 Subroutine and Function Calling.	8-2
<u>CHAPTER 9</u>	
9. SUBPROGRAMS	9-1
9.1 COPE Main Program.	9-1
9.2 BDATA1 Block Data Subprogram	9-3
9.3 FPRCNT Function.	9-3
9.4 GAMMA Function	9-4
9.5 IDCHAR Function.	9-4
9.6 NUMRIC Function (Logical Valued)	9-5
9.7 URAN31 Function.	9-5
9.8 ABRTTL Subroutine.	9-6
9.9 ADDTBR Subroutine.	9-6
9.10 ADDTOF Subroutine.	9-6
9.11 ARTCHK Subroutine.	9-6
9.12 BLOCHK Subroutine.	9-7
9.13 CCPLLOT Subroutine.	9-7
9.14 COMMNT Subroutine.	9-7
9.15 CREAD Subroutine	9-7

TABLE OF CONTENTS - Continued

	PAGE
9.16 DETCTN Subroutine.	9-7
9.17 DEFAULT Subroutine.	9-8
9.18 DFCHK Subroutine.	9-8
9.19 DUST Subroutine.	9-8
9.20 ECHO Subroutine.	9-8
9.21 FINDIT Subroutine.	9-8
9.22 GETRNG Subroutine.	9-10
9.23 GETTIM Subroutine.	9-10
9.24 GETVIS Subroutine.	9-11
9.25 HEADER Subroutine.	9-12
9.26 HITCHK Subroutine.	9-12
9.27 INITLZ Subroutine.	9-12
9.28 INPUT Subroutine.	9-13
9.29 LOSCHK Subroutine.	9-14
9.30 MINTRN Subroutine.	9-15
9.31 NOREC Subroutine.	9-15
9.32 OUTPUT Subroutine.	9-15
9.33 PECHK Subroutine.	9-16
9.34 PENAME Subroutine and PEIDNT Entry	9-17
9.35 PKCHK Subroutine.	9-17
9.36 PPLOT Subroutine.	9-18
9.37 PSMOKE Subroutine.	9-18
9.38 REINTZ Subroutine.	9-18
9.39 RNDREL Subroutine.	9-19
9.40 RNGCHK Subroutine.	9-19
9.41 SEPREC Subroutine.	9-19
9.42 SMOKE Subroutine.	9-20
9.43 SMPLCD Subroutine.	9-20
9.44 TIMCHK Subroutine.	9-21
9.45 TITLE Subroutine.	9-21
9.46 USET Subroutine.	9-21
9.47 VISCHK Subroutine.	9-21
9.48 WARND0 Subroutine.	9-21

CHAPTER 10

10. MEMORY ORGANIZATION: COMMON BLOCKS AND EQUIVALENCE CLASSES.	10-1
10.1 Common Blocks.	10-1
10.2 Equivalence Classes.	10-1

CHAPTER 11

11. FLOWCHARTS	11-1

CHAPTER 12

12. GLOSSARY OF MAIN COPE PROGRAM VARIABLES.	12-1

TABLE OF CONTENTS - Continued

<u>CHAPTER 13</u>	<u>PAGE</u>
13. CONVERSION TO OTHER COMPUTERS.	13-1
13.1 Conversion to Other CDC Computers.	13-1
13.2 Conversion to Non-CDC Computers.	13-1
13.2.1 Hollerith Delimiters and STOP's.	13-1
13.2.2 A FORMATS and Alphanumeric Arrays.	13-2
13.2.3 Word Addressable Mass Storage Files.	13-2
13.2.4 ENCODE and DECODE.	13-3
13.2.5 DATE and TIME.	13-3
13.2.6 STRACE	13-3
13.2.7 EOF.	13-3
13.2.8 COTANGENT.	13-4
13.2.9 Im.n FORMAT.	13-4
13.3 GGAMA Function (IMSL).	13-4
 <u>CHAPTER 14</u>	
14. PREPMS PREPROCESSOR PROGRAM.	14-1
14.1 General Remarks.	14-1
14.2 PREPMS Program Design.	14-1
14.3 Glossary of PREPMS Variables	14-3
14.4 PREPMS INPUTS.	14-6
14.4.1 Option Word Records.	14-6
14.4.2 Data Block Records	14-7
14.4.2(1) Weather Data Block Input	14-8
14.4.2(2) Acquisition Data Block Input	14-27
14.4.2(3) Response Time Data Block Input	14-37
14.4.2(4) Direct Fire Supporssion Data Block Input.	14-45
14.4.2(5) Random Occurrence Data Block Input.	14-50
14.4.2(6) PE Data Block Input.	14-55
14.4.2(7) Posture Data Block Input	14-58
14.4.2(8) Invariant Data Block	14-64
14.4.3 REPORT RECORD LIST Directive	14-70
14.4.4 INTERPRET PEDATA Directive	14-70
14.5 PREPMS Output.	14-70
14.6 PREPMS Diagnostic Messages	14-71
14.6.1 Messages of Informational Nature Not Terminating Program.	14-71
14.6.2 STOPS With Messages in Day Files	14-71

TABLE OF CONTENTS - Continued

	PAGE
<u>CHAPTER 15</u>	
15. PRBLOS PREPROCESSOR.	15-1
15.1 Method For Calculating PRBLOS.	15-1
15.2 Description of PRBLOS Program and Glossary of Variables	15-4
15.3 PRBLOS Inputs.	15-7
15.4 PRBLOS Output and Diagnostics.	15-15
15.4.1 Output	15-15
15.4.2 Diagnostic ("STOP") Messages in Dayfile.	15-15
<u>CHAPTER 16</u>	
16. PAM PREPROCESSOR	16-1
16.1 General Description of Program (PAM)	16-1
16.2 PAM Inputs	16-2
<u>APPENDIXES</u>	
APPENDIX A: Listing and Sample Case of PAM	A-1
APPENDIX B: Listing and Sample Case of PRBLOS.	B-1
APPENDIX C: Listing and Sample Case of PREPMS.	C-1
APPENDIX D: Listing and Sample Case of COPE.	D-1
APPENDIX E: Notational Conventions	E-1
APPENDIX F: Glossary of Abbreviations.	F-1
REFERENCES (BIBLIOGRAPHY)	REF-1
DISTRIBUTION LIST	DIS-1

LIST OF COMPUTER LISTINGS

<u>Listing</u>		<u>Page</u>
6-1	Title Page.	6-2
6-2	"\$\$\$" Comment Page.	6-4
6-3	Case Heading Page	6-5
6-4	Case Results Page	6-9
7-1	Short Form of Echo Print.	7-13
7-2	Long Form of Echo Print	7-16
A-1	FORTRAN Listing of PAM Program.	A-3
A-2A	Sample Input Set 1 for PAM.	A-14
A-2B	Sample Input Set 2 for PAM.	A-15
A-2C	Sample Input Set 3 for PAM.	A-16
A-3A	Sample Run Stream 1 for PAM	A-17
A-3B	Sample Runstream 2 for PAM.	A-18
A-3C	Sample Runstream 3 for PAM.	A-19
A-4	Sample Output from PAM.	A-20
B-1	FORTRAN Listing of PRBLOS Program	B-3
B-2	Sample Input for PRBLOS	B-6
B-3	Sample Runstream for PRBLOS	B-7
B-4	Sample Output from PRBLOS	B-8
C-1	FORTRAN Listing of PREPMS Program	C-3
C-2	Sample Input for PREPMS	C-15
C-3	Sample Runstream for PREPMS	C-30
C-4	Sample Output from PREPMS	C-31
D-1	FORTRAN Listing of COPE Program	D-3
D-2	Sample Input for COPE	D-85
D-3	Sample Runstream for COPE	D-86
D-4	Sample Output from COPE	D-87

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1-1	COPE and Its Preprocessors.	1-3
2-1	"Shooting Gallery" LOS Model.	2-5
2-2	"Random Occurrence" LOS Model	2-6
2-3	Third Communication Model	2-10
3-1	Geometry of COPPERHEAD Fire Mission	3-3
8-1	General COPE Logic Flow	8-3
11-1	Flow Chart of Main Program of COPE.	11-4
11-2	Flow Chart of Subroutine INPUT of COPE.	11-10
14-1	Logic Flow of PREPMS.	14-2

LIST OF TABLES

<u>Table</u>		<u>Page</u>
8-2	COPE Program Unit References.	8-4
8-3	System and IMSL Routine References.	8-7
8-4	COPE Subroutine "Tree".	8-8
10-1	Common Blocks in Each Subprogram of the COPE Program.	10-3
10-2	Subprograms of COPE in which Each COMMON Block Occurs.	10-5
10-3	COPE Equivalence Classes.	10-7

Next page is blank.

PREFACE

This report documents the COPE methodology and computer programs.

Various sections of this report are aimed at different audiences:

Chapter 1 is an introduction to COPE and to this report.

Chapters 2 and 3 are intended for the analyst who is concerned with the modeling methodology and assumptions.

Chapters 4, 5, 6, 7, 14, 15, and 16 are aimed at the user who wants to use the COPE model and its preprocessors in production runs, but who is not interested in programming details.

Chapters 8, 9, 10, 11, 12, and 13 are intended for the programmer who wants a thorough understanding of the COPE computer code in order to implement it on other machines or simply to make changes to it.

The first four appendixes (A, B, C, and D) include program listings and sample cases. The sample cases are cumulative in the sense that the three PAM sample input sets should be run using respectively the three PAM sample runstreams before running the PREPMS program. The sample PREPMS case should then be run using the TAPE 11 created by the PAM runs (this will result in the output of Appendix C). Finally, the COPE sample case should be run using the TAPE 11 created by PAM and PREPMS (this should result in the output of Appendix D). Running these sample cases and comparing the results with those in the appendixes provide a check for correct implementation of COPE and its preprocessors on other machines.

If discrepancies are found in this publication or the associated software, please send notice to:

Director
US Army Materiel Systems Analysis Activity
ATTN: DRXSY-GS (Mr. R. Sandmeyer)
Aberdeen Proving Ground, MD 21005

Suggestions for improvements to this report or the associated programs may also be sent to the above address.

In addition, requests for magnetic tape copies of the COPE programs and related files should be sent to the same address along with the appropriate paperwork for such requests.

This report may be updated by the issuing of revised pages. The frequency of such updates will, of course, depend on the number of errors requiring correction, the changes made in the model, and the demand for the report. Those desiring to maintain the latest versions of the report and the programs should be certain that they are on the COPE distribution list to be maintained by AMSAA.

Next page is blank.

NOTE ON CLASSIFICATION

This report is unclassified, however, when the COPE model is used for actual production runs, it requires some classified inputs. The inputs used in the sample cases (see Appendixes) are correct data except for false values which have been inserted in place of those data whose correct values are classified.

In addition, comment cards with the characters "CSEC" in columns 1-4 point out places in the program code where fictitious constants have been substituted to replace classified values in the listings of PAM and COPE (Listings A-1 and D-1).

The following data are classified (at the time of publication of this report) when correct values are used and identified as such:

In the PAM program: ETH, THEMH, and DISMH are classified by Reference 1 below. ED is classified by Reference 2.

In PREPMS and COPE programs, the values of PE, RNGTTF, and PKTBL, as well as the true seeker sensitivity value, are all classified by Reference 1.

¹Security Classification Guide, Projectile, 155mm, Cannon Launched, Guided, XM712, (CLGP), dated 24 July 1979.

²Security Classification Guide, Ground Laser Designators, dated 16 June 1978.

Next page is blank.

ACKNOWLEDGEMENTS

The COPE model was developed by the Support Weapons Analysis Branch (SWAB) of the Ground Warfare Division (GWD) of the US Army Materiel Systems Analysis Activity (AMSAA).

The requirement to create such a model came from David Hardison, Deputy Under Secretary of the Army for Operations Research (DUSA-OR). A preliminary model proposal was put forward by Herbert Fallin of the Office of DUSA-OR.

When AMSAA received the requirement and proposed model, a team of four analysts of the System Evaluation Section of SWAB was established to develop the model, oversee the collection of data, and, as it turned out, participate in the COPPERHEAD COEA by doing production runs and writing up the results. This team consisted of the section chief, Richard Scungio, who served as team leader and overall coordinator, Julian Chernick, Michael Starks, and the author of the present report, Richard Sandmeyer. Much of the design of the COPE model was accomplished by discussion within the team and individual credit cannot easily be assigned. However, COPE drew heavily on the probability of engagement work done previously by Mr. Chernick and Mr. Starks (see bibliography) including the modeling of weather; the PAM program written by Michael Starks was a derivative of a program used in that earlier work. Mr. Sandmeyer converted the conceptual COPE model into a working computer program (COPE) and also wrote the remaining two programs of this report (PREPMS and PRBLOS).

This report is, of course, concerned with the COPE model and its implementation as a computer program, but a word of appreciation to those who helped gather data for COPE is appropriate here even though this report does not deal with the data collection process (for the data sources used, see COPPERHEAD COEA in bibliography). Data for use in COPE was gathered by David Barnhart, Diana Frederick, Annie Young, and Edward Stauch (all of SWAB). In addition, David Kilminster of SWAB had previously assisted Mr. Chernick and Mr. Starks in obtaining and using weather data. Charles Cairns of AMSAA supplied LDWSS probability of hit numbers and MAJ Charles Williams of FT Sill, OK, provided advice and served as liaison between AMSAA and FT Sill. The SMOKE data was obtained from Sid Gerard (then of AMSAA, now of PM, SMOKE).

Special thanks is given to Mrs. Robin DeFranks, who edited and typed this lengthy report. Thanks also to Mary Webb of SWAB who typed the text in the flow charts of Chapter 11.

Next page is blank.

CHAPTER 1

1. INTRODUCTION

1.1 COPE Background.

Traditionally the item level performance of a weapon system has been expressed in terms of the firepower characteristics of the weapon. The weapon characteristics normally included in the computation of firepower were range, accuracy, lethality, and rate-of-fire. With the introduction of more sophisticated weapons, an awareness of the need to include additional factors in the assessment of a weapon's performance has emerged at both the decision and R&D levels. These additional factors have collectively been referred to as the "Battlefield Environment" and include natural phenomena as well as conditions imposed on the system by friendly and enemy forces.

The need for quantifying a weapon system's performance within the battlefield environment recently became most critical with the development of the 155mm XM712 COPPERHEAD artillery projectile. To study its potentially degraded performance under some battlefield environments, AMSAA developed a computer model which is known by the acronym COPE (COPPERHEAD Operational Performance Evaluation).

In addition to being a model for assessing the performance of a weapon system, the model and the manner of generating the input requirements constitute a novel approach to the evaluation of weapon systems. This approach results in a manner of assessing the performance of most weapon systems and potentially can quantitatively produce insight into the performance of one or more weapon systems which previously could only be treated subjectively.

This report describes the COPE model as currently used at AMSAA for the assessment of COPPERHEAD. AMSAA has modified the computer code to obtain models that analyze the performance of other weapon systems within the context of a COPE type "operational performance" analysis. The term COPE itself has now come to have two meanings: (1) "Combat Operational Performance Evaluation" which is the generic name for all of the derivative models of the original COPPERHEAD version and (2) "COPPERHEAD Operational Performance Evaluation" which is the first model in this family of "Operational Performance" models and is the one described in this report.

The models in the COPE family include:

(1) COPPERHEAD Operational Performance Evaluation (COPE) used to evaluate the XM712 COPPERHEAD artillery round,

(2) Tank Performance Evaluator (TAPE) used as part of the XM1 tank evaluation,

(3) TOW Operational Performance Evaluator (TOPE) used to evaluate the TOW anti-tank missile,

(4) SADARM Operational Performance Evaluator (SOPE) used to evaluate the proposed SADARM artillery round,

(5) Top Attack Evaluation Model (TAEM) used to evaluate candidates in the Top Attack Study,

(6) Ground Launched HELLFIRE Operational Performance Evaluation (GLHOPE) used to evaluate ground launched HELLFIRE,

(7) RPVCOPE used to evaluate COPPERHEAD when designation was performed by an RPV, and,

(8) HELLFIRE Operational Performance Evaluation (HOPE) used to evaluate HELLFIRE launched from helicopters.

The input requirements for these models range from moderate to massive. The TAPE and TOPE models for example require no preprocessors and take less input than the COPPERHEAD version. The TAEM program, on the other hand, requires as input results generated by the very costly, long running, TGSM (Terminally Guided Submunition Model) program.

This report is concerned only with the COPE model (in all remaining sections of this report, COPE shall have its second, narrower meaning: "COPPERHEAD Operational Performance Evaluation"); however, much of what is discussed applies to the other models in the family. Any of the models of the COPE family can be obtained from AMSAA, though at the present time only the COPPERHEAD and tank versions have been extensively documented.

1.2 Overview of COPE Computer Programs.

There are actually four computer programs used in the COPPERHEAD Operational Performance Evaluation. They are:

(1) The main COPE program (formerly called SLATCH for Simple Look at the COPPERHEAD) descended from the original proposal put forward by DUSA-OR.

(2) The PAM (Probability of Seeker Acquisition and Round Maneuver) model used to create the probability of engagement tables used as input to the main COPE program. This model is described and documented more thoroughly in a forthcoming AMSAA technical report.

(3) The PREPMS (Preprocessor Mass Storage) program that is used to create and modify the word addressable mass storage file used as the COPE "data base".

(4) The PRBLOS (Probability of Line-of-Sight) model used to obtain the probability of having line-of-sight for the terminal phase of the COPPERHEAD trajectory.

All four of these programs are documented in this report, though the PAM model is covered in much less detail than the other three since a more detailed treatment of it will be published in a separate report.

The relationships among the four programs and the word addressable "data base" file are shown in Figure 1-1.

The major data sets (called "data blocks" later in this report) are put on the data base by PAM and PREPMS. Then when one wishes to run COPE, it is necessary only to submit a string of simple input commands describing the cases to be run. For example, when running a case using December weather for 0600 (6 a.m.), one need merely submit a card: WEATHER,
DECEMBER, 0600 rather than read in thirty or forty cards filled with the appropriate weather data. This arrangement makes it extremely simple to run many cases of COPE with relatively few input cards once the data base file has been created.

In the event that one wishes to run cases that use a data set not on the data base file, there is a "temporary" option that allows one to read the data into COPE directly; however, if one intends to run multiple cases with such data, it will be easier in the long run to read it onto the permanent data base file once using the PREPMS program and then "call it up" for use as necessary in COPE by submitting the appropriate single card command.

All four programs are written in CDC FORTRAN Extended Version 4.

CHAPTER 2

2. DESCRIPTION OF COPPERHEAD MISSION MODELING

2.1 Application Area of Simulation.

The main COPE program is a simulation of a COPPERHEAD fire mission under battlefield conditions. It takes into account such degrading effects on designator, fire direction center (FDC), artillery battery, and COPPERHEAD round performance as meteorological conditions, terrain shielding, smoke, dust, enemy artillery preparatory fires, enemy direct fire against the designator, communication failures, and target location error as well as numerous factors that affect COPPERHEAD's ability to engage and hit a target.

The main COPE program was designed to evaluate the operational performance of the overall COPPERHEAD system under battlefield conditions.

The output of the program enables one to see which factors contribute most to the degrading of COPPERHEAD system performance and which are, therefore, areas in which performance improvement efforts should be concentrated.

In addition, by varying the input value(s) associated with a particular factor and then studying the results, one can perform the customary sensitivity analysis. A series of such analyses can be expected to show that changes (within realistic limits) in some factors have little effect on overall system operational performance whereas changes in other factors will result in major performance improvement or degradation.

By comparing those factors to which system operational effectiveness is most sensitive with those factors over which system designers and tacticians have some control, one can determine what, if any, design changes or employment doctrine changes might be made to improve overall COPPERHEAD system operational performance. This comparison and analysis, of course, is done outside the COPE program, but the resulting modified COPPERHEAD system and employment doctrine might then be evaluated in the COPE model or a descendent of it.

Finally, several derivative models have been developed from an earlier COPE program (SLATCH version 3.2) and are in use for evaluating tanks, TOW, and Hellfire under battlefield conditions. These models allow limited but informative comparisons of the operational performance of the different weapon systems.

2.2 Description of Model.

2.2.1. Outline of COPPERHEAD Mission. The heart of the main COPE program is the Monte Carlo simulation of a potential COPPERHEAD fire mission. This simulation consists of a series of steps that would occur in an actual COPPERHEAD fire mission. For a successful mission, these steps start with the unmasking from terrain shielding of a target and end with the killing of a target by COPPERHEAD. For an unsuccessful mission, the steps end before the target is killed.

The steps are grouped into three series bounded by four major events. The following outline gives the four major events as well as the steps (or tests) that must be passed to reach the next major event. All steps and events occur in the order listed.

This describes the tests when the so-called "shooting gallery" line-of-sight model is used. Following the list is a description of the modifications required when the "random occurrence" line-of-sight model is used.

Event 1. A target (one or more target vehicles) unmask (i.e., enters a region within line-of-sight of the designator operator's position). This is termed an "occasion" and begins the simulation of the potential COPPERHEAD mission.

Test 1: Has the designator operator survived enemy preparatory artillery fires up to the present time?

Test 2: Is the target within the meteorological visibility range limit of the designator operator for the present weather conditions?

Test 3: Is the target within the designation range of the laser designator system being played?

Test 4: Did the designator operator detect/acquire the target before it again became masked by terrain?

Test 5: Is the line-of-sight from designator-to-target sufficiently free of smoke that the designator operator can see and designate the target?

Test 6: Is the line-of-sight from designator-to-target sufficiently free from dust for the designator operator (D.O.) to see and designate the target?

Test 7: Is the target sufficiently far away from the designator operator that he will stay and designate rather than "bail-out?"

Event 2: The designator operator has acquired a target and decides to engage it (i.e., call for COPPERHEAD fire against it). This is termed an "attempted engagement."

Test 8: Is the designator operator able to communicate with the fire direction center (FDC)?

Test 9: Does the message which the designator operator sends to the fire direction center contain the correct information and does the FDC interpret it correctly?

Test 10: Is the target still (after time has been spent on communication and preparing to fire the mission) sufficiently far away that the designator operator will stay and designate rather than "bail-out?"

Test 11: Does the designator operator still have line-of-sight to the target?

If tests 8 through 11 are all passed, then conditions are such that a COPPERHEAD fire mission occurs. This leads to:

Event 3: The battery is ready to fire and the designator is ready to designate. A COPPERHEAD round is fired. This is termed a "shot".

Test 12: Does the designator operator still have line-of-sight to the target during the final critical time interval of the COPPERHEAD trajectory?

Test 13: Has the designator operator been warned to begin lasing the target or has he seen the first COPPERHEAD round of this fire mission already impact?

Test 14: Does the designator survive direct fire from the target being designated and is the line-of-sight to the target unobscured by direct fire rounds landing in front of the designator?

Test 15: Is the COPPERHEAD round just fired reliable?

Test 16: Is the line-of-sight from the designator operator to the target vehicle unobscured by small terrain features?

Test 17: Does the round come sufficiently close to the target for the seeker to receive enough reflected laser energy to engage the target and is the target within the maneuver footprint of the COPPERHEAD?

Test 18: Does the COPPERHEAD round hit the target?

Test 19: Does the COPPERHEAD round kill the target?

If tests 12 through 19 are all passed, then we have:

Event 4: The target vehicle is killed. This is termed a "kill."

If more COPPERHEAD rounds are to be fired for this mission, the model time is moved ahead to account for the time between rounds, and the model resumes at Event 3 for the next COPPERHEAD round.

To model the "random occurrence" line-of-sight case, tests 4 and 11 are not made and test 12 uses a different assumption.

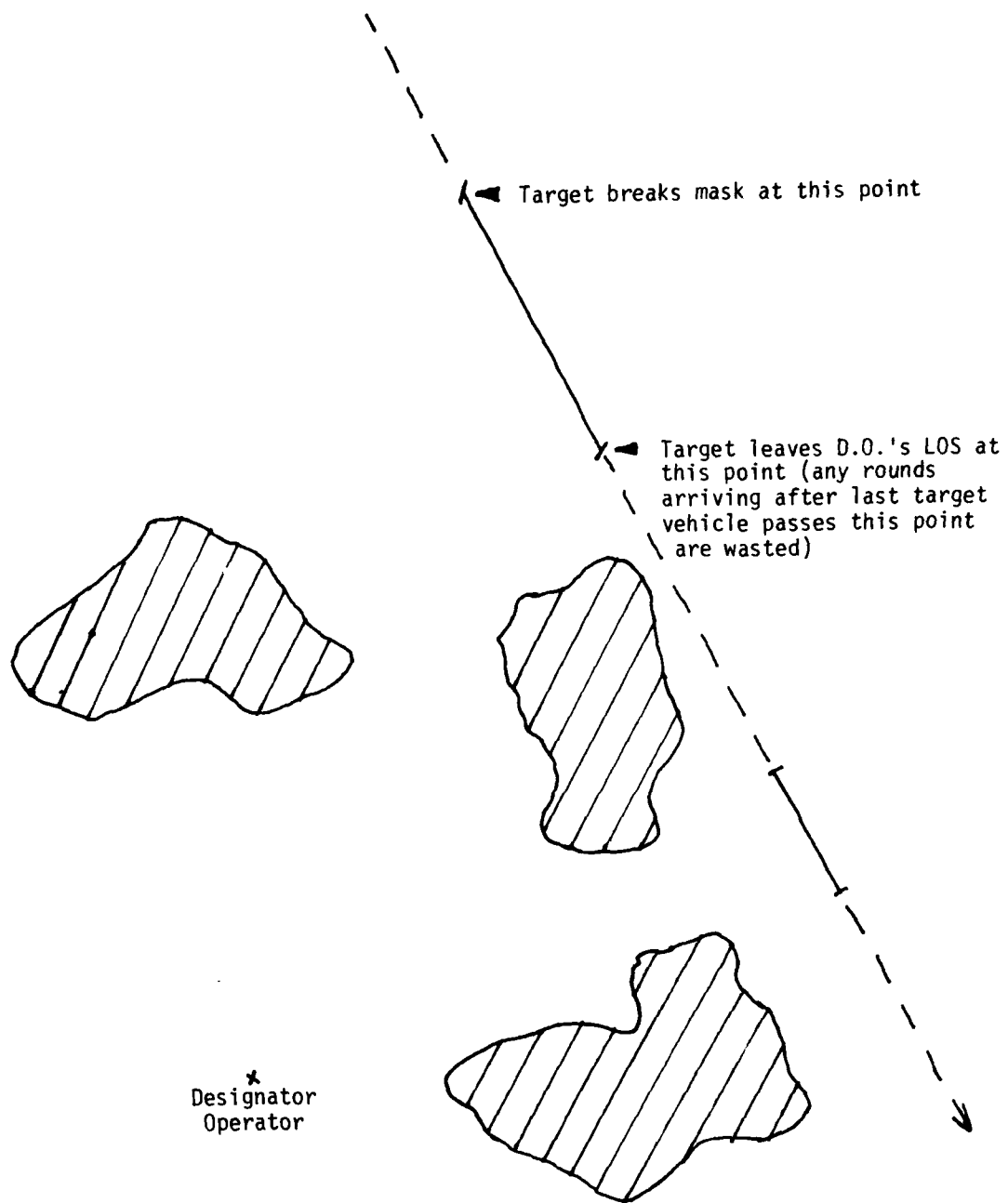
The two ways of handling the line-of-sight problem are called the "shooting gallery" method and the "random occurrence" method.

The "shooting gallery" method assumes that the entire COPPERHEAD mission must be completed during the time interval starting with the first target vehicle entering line-of-sight segment and ending with the last vehicle of the target leaving that same line-of-sight segment (see Figure 2-1).

The "random occurrence" method on the other hand assumes that target vehicles are passing in and out of view. Instead of assuming that all rounds fired after the last target vehicle leaves the line-of-sight segment in which it was acquired are wasted as in the "shooting gallery" technique, the "random occurrence" method does not check loss of line-of-sight until the last critical seconds of the COPPERHEAD trajectory and even then the check allows for the possibility that a target may have left the first line-of-sight segment but entered a second (or subsequent) line-of-sight segment where COPPERHEAD can be used. The "random occurrence" line-of-sight model, then merely checks whether there is at least one unkilld target vehicle that will be within line-of-sight of the designator for the final critical seconds of the COPPERHEAD trajectory (see Figure 2-2).

Now each step (test) in the model is explained in detail:

2.2.2 Preparatory Fires. The first test encountered by a potential COPPERHEAD mission is for designator operator survivability of preparatory fire. A probability that the designator operator is killed by enemy preparatory artillery is input (either explicitly or by default) and for each replication (i.e., each sample potential COPPERHEAD mission) of the given case, a uniformly distributed (pseudo-) random number is drawn. If that random number is less than the input probability that the designator operator is killed by enemy preparatory artillery fire, then the potential COPPERHEAD



Scale 0 250m

——— Target in D.O.'s LOS

- - - - - Target not visible to D.O.

FIGURE 2-1 "Shooting Gallery" LOS Model

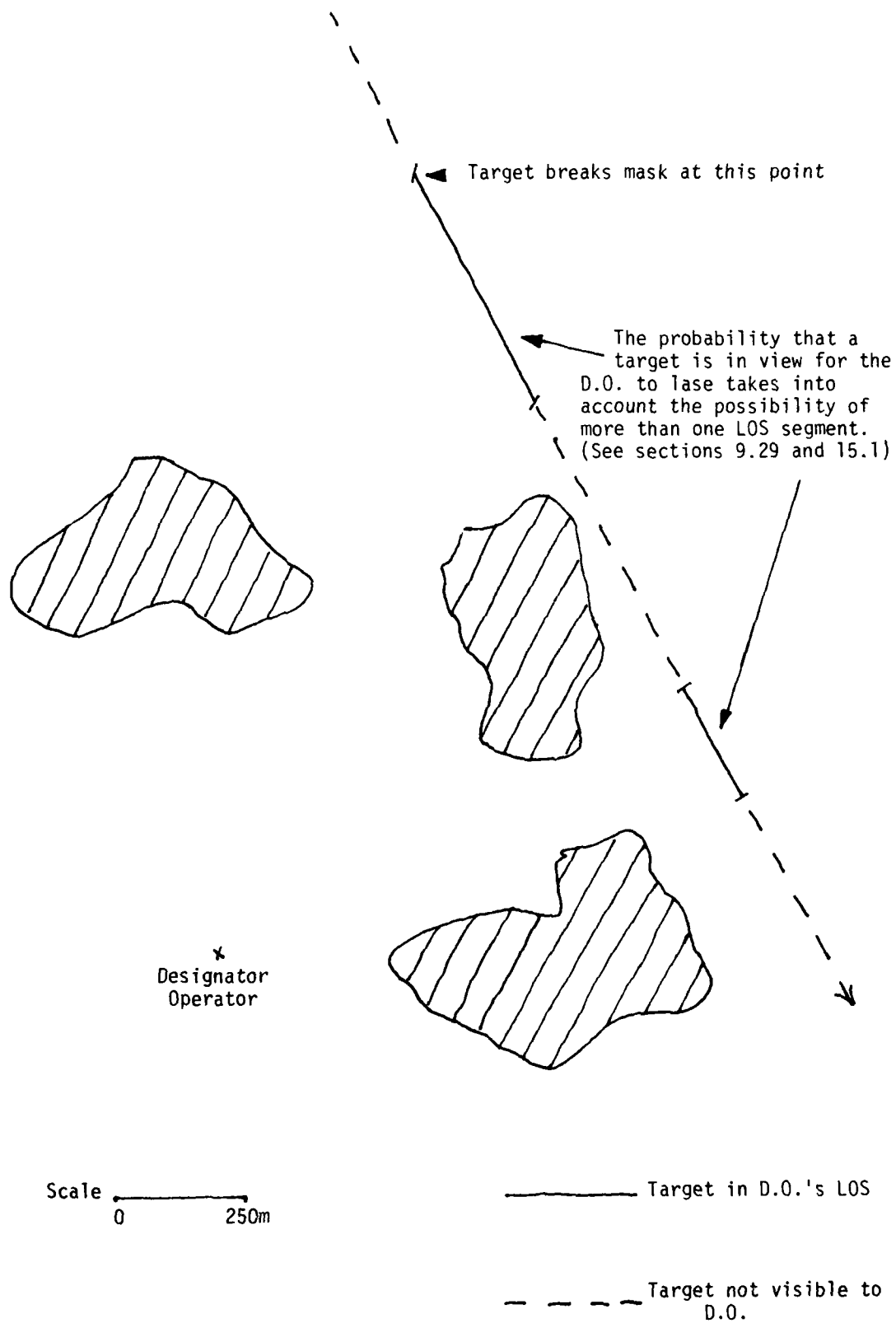


FIGURE 2-2 "Random Occurrence" LOS Model

mission aborts because the designator operator is killed; otherwise, the potential COPPERHEAD mission continues to the next test.

2.2.3 Visibility Range. The next test depends on the weather conditions. For the month and time of day that is used for a given case, there is a distribution of weather conditions which includes the probabilities of various cloud ceiling altitude and meteorological visibility range combinations as well as probabilities of various Pasquill atmospheric stability categories, windspeeds, and relative humidities. For each replication of a given case, the cloud ceiling altitude and meteorological visibility range distribution is sampled and the results (cloud ceiling altitude and meteorological visibility limit (range)) obtained are recorded for later use.

The program then samples the acquisition range distribution once for each replication to obtain the range from designator-operator (D.O.) to target terrain unmask for the current replication. If the target unmask from terrain shielding at a range from the designator-operator beyond the current replication's meteorological visibility limit, then the potential COPPERHEAD mission is aborted since the target cannot be seen and, therefore, cannot be lased by the designator operator; if, on the other hand, the target unmask at a range less than the visibility limit, then the next test is made.

2.2.4 Designator Maximum Range. Using the above random sample from the acquisition (terrain unmask) range distribution, the next test compares the designator-to-target range with the maximum designator range. If the target is beyond the maximum designation range, then the potential COPPERHEAD fire mission is aborted; otherwise, the tests continue.

At night, the maximum designator range is reduced to the range to which the thermal imagery night sight is usable for aiming the designator.

2.2.5 Target Not Detected. The next test is performed to determine whether the target was in view long enough to be acquired (the entire detection, recognition identification process is lumped into one test and called detection).

After a range from designator-operator to target terrain unmask is obtained as above, a line-of-sight segment length distribution is sampled to obtain the length of the section of the target's path extending from the point of initial terrain unmask to the first point along the path that is obscured from the D.O.'s view. From this line-of-sight segment length together with the target velocity and length of the target unit column, the duration of line-of-sight is calculated. This duration is the time from initial target unmask until the last vehicle in the target unit's column passes out of the D.O.'s view.

To determine whether the target is detected, a time is sampled from a distribution of detection times. This time represents the delay from the time of initial target unmask until the D.O. has detected, recognized, and identified the target and decided to call for COPPERHEAD fire. This is termed detection time.

If the detection time is greater than the duration of the line-of-sight on a particular replication, then the target is considered to have gone in and then out of view before the D.O. could detect it (or really before he could call for fire against it) and the mission aborts. If the detection time is less than the line-of-sight duration, then the D.O. detects and calls for fire on the target before it leaves his line-of-sight and the model continues to the next test.

2.2.6 Smoke. The next test is to determine whether there is sufficient smoke in the area of the target to prevent the D.O. from either seeing or lasing the target. The smoke model computes the fraction of targets that could be obscured from view by taking into account such factors as number and type of RED smoke rounds fired and weather conditions for the month and time of day considered in the current case.

As an alternative to specifying a volume of RED smoke rounds which is then used to calculate the fraction of targets obscured, one can simply input a fractional value which represents the fraction of targets obscured by smoke.

In either case, the fraction of targets (actually fraction of a specified front width) obscured by smoke is played as a "probability of abort due to smoke." For each replication that reaches the smoke test, a random number is drawn from a uniform distribution (from 0 to 1) and if it is less than the probability of abort due to smoke, then the mission is aborted at this step; if it is greater than the probability of abort due to smoke, then the next test is performed.

Note that while the D.O. has a limited target detection capability in smoke by using a thermal site on the designator, the laser beam itself could not penetrate the smoke in those cases where smoke was thick enough to require use of a thermal site. Consequently, COPE plays a "black-white" smoke situation: either smoke completely aborts the mission or has no effect. This is not the case for some of the other COPE variants (such as TAPE, SOPE) where there are three possibilities: (1) smoke is not present and, hence has no effect, (2) smoke is present and kills mission, (3) smoke is present but target can still be detected through thermal site and target is engaged but with reduced accuracy.

2.2.7 Dust. The next test is whether the dust from RED HE artillery fire is sufficient to prevent the D.O. from either seeing the target or lasing it. This is modeled in a very simplistic way in the current generation of COPE.

A probability that dust aborts the mission is estimated outside the COPE program and then input for a given COPE case (either explicitly or by default). Then for each replication that reaches the dust test, a random number is drawn from a uniform distribution (0 to 1) and compared to the probability that dust aborts the mission. If the random number is less than the probability that dust aborts the mission, then the mission is aborted; otherwise, the next test is made.

2.2.8 Designator Bail-Out (Precommo). The next test is to check whether the target is so close to the D.O. that in order to survive he would not engage the target. The assumption is that to avoid being killed by direct fire from potential COPPERHEAD targets, the D.O. would move to a new position whenever the enemy front line units closed to within a certain range; in the case of an isolated lead enemy unit, he might merely refrain from designating rather than move, but in either case he would not be calling for COPPERHEAD fire.

To perform this test, the model compares the D.O.-to-target range against a pre-set "bail-out" range. If the D.O.-to-target range is less than the bail-out range, the mission aborts; otherwise, the mission proceeds to the next test.

Note that this bail-out check is made prior to the D.O.'s call for COPPERHEAD fire. A second bail-out check (see section 2.2.10) is made later.

2.2.9 Commo-out (Tests 8 and 9). The next test determines whether the D.O. has a communication link with the F.D.C.

Communication delay time and probability of success (Test 8) can be played in any one of three ways depending on the input option selected:

(1) A single response time distribution is used together with a single probability of having a good communications link. In this case, a number is randomly sampled from the response time distribution to determine the total delay time from the D.O.'s call for Copperhead fire until the firing battery is ready to pull the lanyard to fire Copperhead. In addition, a second random number is drawn, this time from a uniform (0 to 1) distribution; if this random number is less than the probability of having a good communication link, then the mission proceeds to the Test 9; otherwise, the mission is aborted.

(2) The second way of playing communication delay time and probability of good communication link is called the "parameterized response time" option. This case is essentially the same as (1) above except that the delay time is set at a single fixed value that is the same for every replication (as opposed to sampling a distribution of delay times to obtain a new delay time value for each replication).

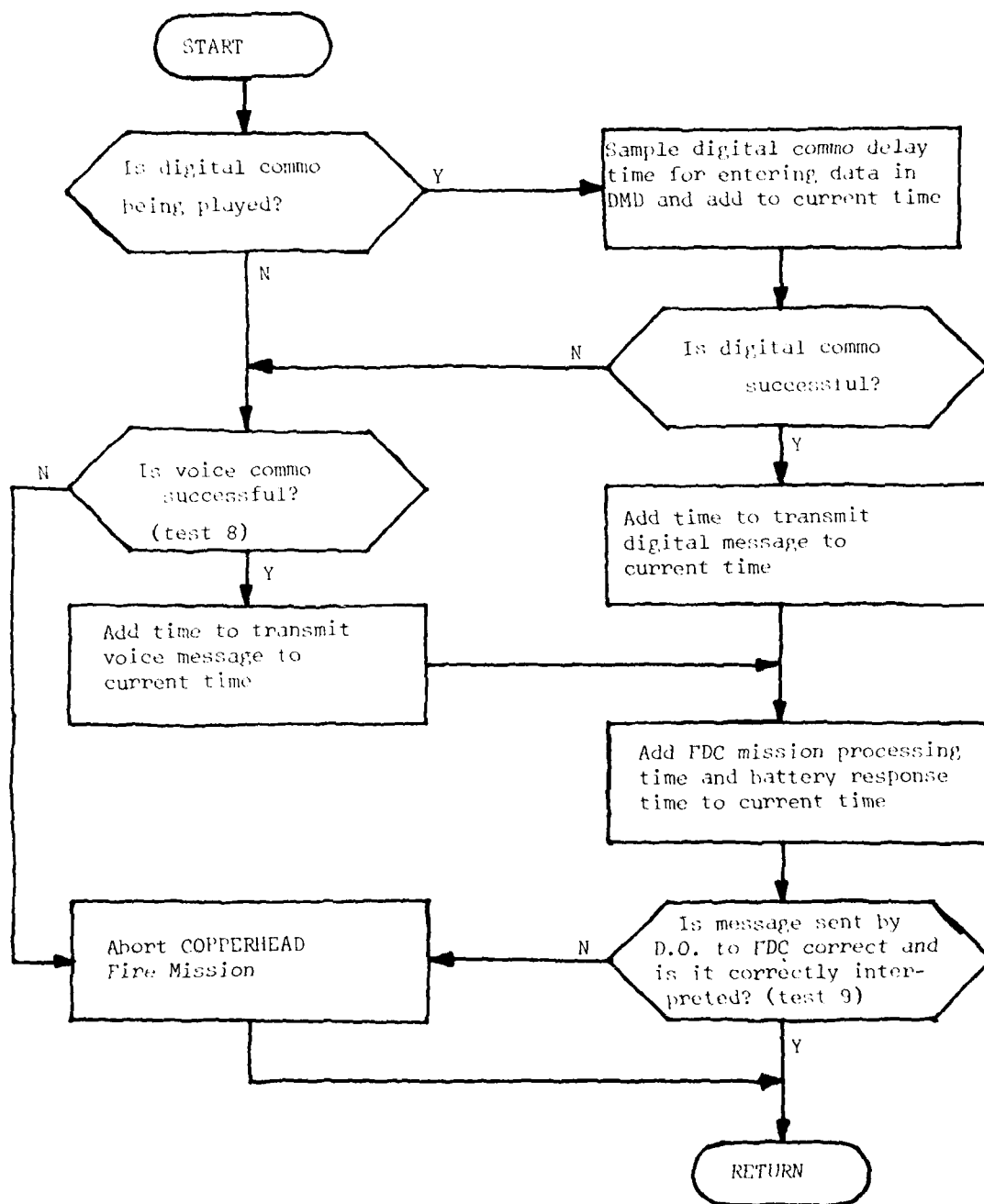


FIGURE 2-3 Third Communication Model

(3) The third way to model the communication delay time is to use a separate delay time for each part of the process from the D.O.'s calling in a fire mission to the battery's pulling the lanyard. Under this method, a delay time distribution is used to obtain digital communication times for D.O. to FDC messages. This delay time is a function of mission type and time of day (i.e., day or night).

The flowchart in Figure 2-3 gives the complete picture of the time delays when this third model is used.

No matter which of the three ways of modeling delay times is used, either communication fails and the mission aborts or communication is established and a delay time is added to the current time. Finally, if communication is established, another random number (uniform between 0 and 1) is drawn and compared to the probability of a successful (i.e., one containing the correct information) message (Test 9). If the random number is less than this probability, the mission proceeds to the next test; otherwise, the mission aborts because the message from the F.O. had bad data or was incorrectly interpreted.

2.2.10 Designator Bail-Out After Communication. After the communication from D.O. to FDC to firing battery is completed successfully, the D.O.-to-target distance is updated to account for target movement during the communication and processing delay. If the updated position brings the target within the "bailout" range of the D.O. (see 2.2.8) then the mission is aborted; otherwise, the mission continues with the next test.

2.2.11 Line-of-Sight Lost Prior to Firing. The next test is to determine whether the D.O. has lost LOS to the target. The current time (detection time plus communication and processing delay times) is compared to the duration of line-of-sight. If the current time is greater than the duration of line-of-sight, then the target has left the D.O.'s field of view prior to the firing of the first COPPERHEAD and the mission is aborted; otherwise, the mission is fired, the time of flight is added to the current time, and the next test is made.

2.2.12 Line-of-Sight Lost During Mission. The tests prior to this affected the entire mission; the sequence of tests after this point is performed separately for each round in the fire mission.

The current time (which is the time the first (next) COPPERHEAD round is to arrive) is compared to the duration of line-of-sight. If the current time is greater than the duration of line-of-sight, then the current round is aborted (ineffective); otherwise, the mission proceeds to the next test.

2.2.13 Designator Not Warned In Time.

At the time the first COPPERHEAD round is fired for a given mission, a message is sent to the D.O. by the FDC telling him to begin lasing the target. To determine whether the D.O. receives this message, a random number is drawn from a uniform (0 to 1) distribution and compared to the probability of D.O. warned to lase. If the random number is less than the probability, then the message gets through, the D.O. lases, and the next test is performed; otherwise, the message does not get through, the D.O. fails to lase, and the first round is wasted (aborted).

Even if the message warning the D.O. to lase does not get through for the first COPPERHEAD round, the model assumes that the D.O. would see the first round impact and hence know to begin lasing for the second and subsequent rounds; therefore, rounds after the first round are never lost because of D.O.'s failure to lase.

2.2.14 Designator in Direct Fire. The next test determines whether the D.O. is unable to lase due to incoming direct fire. It is assumed that the target being lased for the COPPERHEAD fire mission has a laser alarm as well as capability that allows it to locate the D.O.'s position (when his laser is on) with sufficient accuracy to fire.

A random number is drawn from a uniform (0 to 1) distribution and compared to probability of kill and probability of suppression values (which are a function of D.O.-to-target range). If the random number is less than the probability of kill, then the D.O. is considered killed and the current round and all subsequent rounds for this replication are aborted. If the random number is greater than the probability of kill but less than the sum of the probability of kill plus the probability of suppression (obscuration due to direct fire), then only the current round is aborted as a result of this test (the test will be performed separately on subsequent rounds of this replication).

If the random number is greater than the probability of kill or suppression, then the mission continues to the next test with no effect from the direct fire.

2.2.15 Round Reliability Failure. The round is next tested for inflight reliability. A random number is drawn from a uniform distribution (0 to 1) and compared to the round reliability. If the random number is less than the reliability, the round is reliable and the next test is made; otherwise, the round fails.

2.2.16 Target Obscured By Mini-Terrain. The D.O.-to-target range is computed allowing for target movement up to the current model time (actually, the time the first (or next) COPPERHEAD round is to arrive). A distribution of target exposures (due to mini-terrain) at that range is then sampled to determine whether the target is fully exposed, hull defilade (turret exposed), or completely obscured. If the target is completely obscured, then the round aborts; otherwise, the next test is made.

2.2.17 Round Did Not Engage Target. The next test determines whether the COPPERHEAD round receives sufficient reflected laser energy to initiate maneuver to a target and has the target in its maneuver footprint.

To perform this test, a random number is drawn from a uniform (0 to 1) distribution and then compared to a probability of seeker engagement value which is a function of designator-to-target range, visibility range limit, cloud ceiling altitude, delay time, and number of vehicles in column as well as reflectivity of target, seeker sensitivity, target heading, target velocity, nominal COPPERHEAD system response time, gun-target-D.O. angle, designator type, and deflection biases (offsets). The probability of seeker engagement is the probability that both the seeker receives sufficient reflected laser energy to initiate tracking and the target is within the maneuver footprint of the COPPERHEAD round. If the random number is less than the probability of seeker engagement, then the next test is performed; otherwise, the round fails because its seeker cannot engage the target (or can engage but cannot maneuver to the target).

2.2.18 Round Engaged But Did Not Hit. A probability that the round hits the target given that it engages (i.e., passes the previous test) is obtained as a function of range, exposure (fully exposed or hull defilade), and designator type. A table of these probabilities is obtained as output from the LDWSS program and is intended to account for the possibility that the COPPERHEAD seeker tracks laser over-spill or under-spill rather than the target itself.

Again a uniform distribution (0 to 1) is randomly sampled. If the resulting random number is less than the probability that the round hits, then the round hits the target and we proceed to the next test; otherwise, the round misses and aborts.

2.2.19 Round Hit But Did Not Kill. The final test is to determine whether the round killed the target given that it hit the target. To perform this test, a random number is drawn from a uniform distribution (0 to 1) and compared to a probability of kill given a hit value which is a function of target type and target exposure (fully exposed or hull defilade). If the random number is less than the probability of kill, then the target is killed; otherwise, the target is not killed and the round aborts.

2.2.20 Subsequent Rounds. When a round (not a mission) aborts, subsequent tests in the sequence are not made. Instead, the time is advanced to the next round's arrival time (if there are more rounds left to be considered) and the sequence of tests resumes at the "LOS lost during mission" (section 2.2.12) test for the next round.

If a round does not abort, but instead successfully kills a target, then the time is advanced to the next round's arrival time (if applicable) and the sequence of tests resumes at the "LOS lost during mission" (section 2.2.12) test for the next round.

When no more rounds remain to be fired for a given replication, the replication is ended.

2.2.21 End of Replication. When either a mission (not a round) aborts or all the rounds for a replication have been fired, the replication is considered ended. If more replications remain to be made, the next replication begins with the first test, "Preparatory fires" (section 2.2.2) and proceeds through the sequence of tests as described. If all replications for the current case are done, then the results are printed out and the subsequent case (if any) is begun.

2.2.22 Random Occurrence LOS Model vs Shooting Gallery LOS Model. The step by step description given above of the various tests that must be passed for a successful COPPERHEAD fire mission applies to the "Shooting Gallery" or single continuous LOS segment model. This model assumes that the entire COPPERHEAD mission must be completed during the time interval from the first target vehicle's breaking terrain mask until the last target vehicle leaves that same LOS segment.

The alternate LOS model is called the "Random Occurrence" LOS model and assumes that after the first target vehicle breaks terrain mask, the target vehicles will go in and out of mask at random. The sequence of tests for the random occurrence method is the same as for the shooting gallery except that tests 4 and 11 (sections 2.2.5 and 2.2.11) are omitted and test 12 is modified.

Test 12 (section 2.2.12) is modified so that the probability that at least one target vehicle is visible is given by $Q = 1 - (1 - P)^N$ where P is the probability of a single vehicle being visible if randomly placed along an approach path at the current D.O.-to-target range and N is the number of vehicles left unkilld in the target. A random number is sampled from a uniform distribution (0 to 1) and compared to Q . If the random number is greater than Q , the round is aborted; otherwise, the mission continues with test 13.

2.2.23 Note on Nominal Response Time. The probability of engagement numbers are a function of the time after the D.O.'s call for fire that the rounds arrive. The time is played by establishing a nominal response time $\bar{t}_r$ (an average value for the distribution of total mission communication and processing times which is input and to which is added the time of flight appropriate for the given gun-to-target range.) The model assumes that the D.O. estimates the target's location at $\bar{t}_r$ seconds in the future and calls for fire against that point. The actual COPPERHEAD mission response time is sampled and added to the time of flight to obtain the actual arrival time of the round (t_a). If t_a is greater than $\bar{t}_r$, then the probability of engagement is lowered because of the larger aiming error and TLE.

This is explained in greater detail in the report on the PAM model. The report on modeling target location error (TLE) discusses the problem of estimating target location and the error in the estimate as a function of time (see bibliography).

The only point important to the user at this stage is that COPE uses two response times: a nominal response time $\bar{t}_r$ and an actual response time t_a . The nominal response time is used to estimate target's future location and hence to choose the point at which to aim. The actual response time is used to determine the probability of seeker engagement (PE). Naturally, if t_a is very large, the probability of the target being at the point estimated (based on $\bar{t}_r$) is quite small.

Also, it was assumed that if the firing battery saw that t_a was likely to be less than $\bar{t}_r$ on a particular occasion, it would hold up firing to cause $t_a = \bar{t}_r$ rather than drop early rounds far in front of the target. This assumes a good communication capability between D.O. and battery.

CHAPTER 3

3. PERFORMANCE AND LIMITATIONS

3.1 Computer Related Performance and Limitations.

The COPE program (and its preprocessor programs) has been run primarily on the U.S. Army Ballistic Research Laboratory's Control Data Corporation CYBER 76 computer (CDC 7600). On that computer, a typical case with 10,000 replications and six rounds fired per successful COPPERHEAD mission requires 2.5 to 3.0 seconds per case. The computer memory requirement is about 24,000 words (57 K₂) of SCM and no LCM. The disk space required (which will vary with the size of the data base file) for TAPE 11 was about 155,000 words (456 K₂).

COPE has also been run on a CDC 6600 at Aberdeen Proving Ground and on a CDC 6400 and a CDC 6500 at FT. Leavenworth (using the remote terminal at FT. Sill). On these machines, the run time ranged from eight to sixteen times what it was on the CDC 7600. The memory requirements and disk space on the various CDC 6000 series computers were nearly the same as for the CDC 7600.

At the time of this writing, COPE has not been run on any non-CDC machines; however, chapter 13 of this manual considers some of the changes required to do so. Run time would, of course, vary from machine to machine (e.g., the author's experience with other FORTRAN programs suggests run time on the UNIVAC 1108 computer would be eight to ten times that on the CDC 7600).

Most current non-CDC computers use a word size of fewer than 60 bits (the CDC word size). This means that the program memory required to load the object program on most non-CDC machines will be somewhat larger in terms of number of words than for the CDC computers. Similarly, a slight increase in disk space may be required to put the data base file TAPE 11 on a smaller word machine.

3.2 Program Related Performance and Limitations.

The current version of COPE has been run several hundred times now with no errors detected other than those resulting from invalid input, insufficient run time requested, or insufficient print limit set.

The ability of the program to run multiple cases with relatively small streams of input directives has enabled the users to run hundreds of cases with relatively short set up time.

The main limitation encountered in running the program is the requirement that data selected to fill the eight data blocks be on the random access mass storage data base file created by the preprocessor

programs. Once these preprocessors are run, any case using data already on the data base file can be easily run. This subject is discussed in detail in chapter 5.

3.3 Model Related Performance and Limitations.

One of the main limitations of the COPE model is that much tactical detail is represented by distributions. For this reason, great care must be taken to interpret input distributions when reporting model results. The interpretation of results depends heavily on the conservatism and objectivity of the analyst.

The only attempt (and indeed the only opportunity) to verify the COPE model was by running it with case descriptors matching as closely as possible conditions present in the COPPERHEAD OT II. When this was done, the results predicted by COPE were compared to those obtained in OT II and found to be in quite close agreement. More precisely, the hypothesis that the probability of hit given a shot as obtained from COPE is equal to the true population probability of hit given a shot could not be rejected even with a significance level as high as 50 percent based on the sample of firings in OT II.

This good agreement of the model to test results should not lead to uncritical acceptance of all COPE results. For one thing, tests 1, 5, 6, and 14 were nulled out (that is, always passed). Also, since the OT II rounds were not live, the probability of kill values (test 19) in the model were set to one (so that kills and hits were considered equivalent).

There are some limitations to the COPE model that need to be explicitly pointed out:

3.3.1 Geometry of COPPERHEAD Fire Mission. The geometry of the gun, D.O., and target positions is only approximated. For any given simulated COPPERHEAD mission, the gun-to-target range, designator-to-target point-of-unmask range, target heading, and nominal gun-target-designator angle are fixed while the target itself moves (i.e., its position is a function of time).

Figure 3-1 represents the geometry of a COPPERHEAD fire mission.

U is the point of target unmask.

R is the point at which round is to arrive on target (provided nominal response time is met).

D is the designator operator location.

G is the gun location.

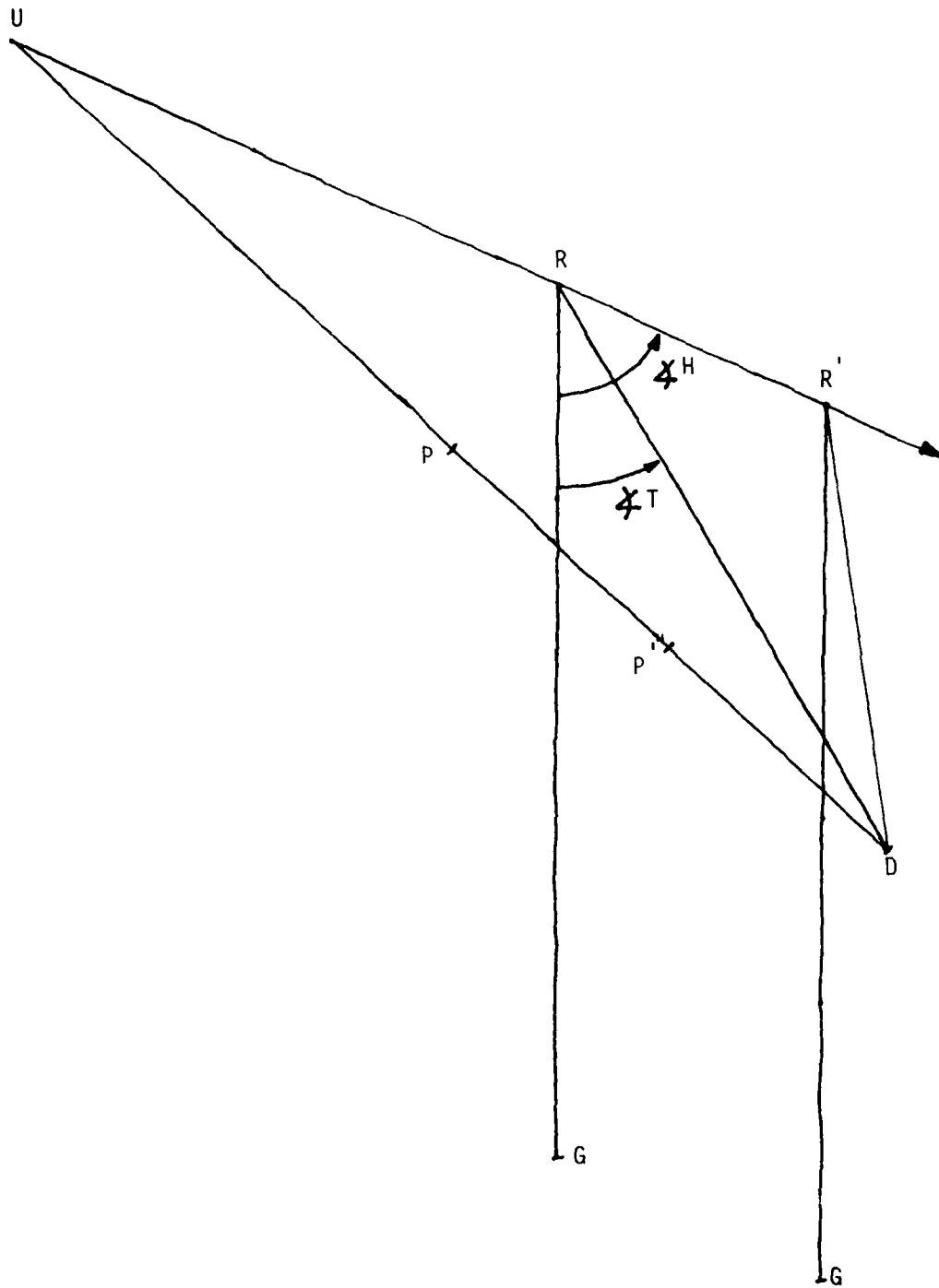


FIGURE 3-1 Geometry of COPPERHEAD Fire Mission

$\angle XH$ is the target heading (angle between gun-target line and target path at time round arrives on target).

$\angle XT$ is the angle between the gun-to-target line (GR) and the designator-to-target line (DR).

UR is the target's path (target travels from U to R).

By the COPE case conditions chosen, $\angle XT$, $\angle XH$, the length of GR , and the length of UD are constant (the first three are constant for a given case; the last, length of UD , is constant for a given replication, but is sampled anew for each replication). Hence, if the response time is such that the target is at R' on one replication rather than at R (as on, say, a previous replication) when the round arrives, one must move G to G' , and D to D' to preserve the fixed values of $\angle XT$, $\angle XH$, length of GR (now $G'R'$).

The requirement that G move to G' is, of course, unrealistic since one does not move the battery to a new site just because nominal response time is not met.

A second compromise in the geometric representation of the COPPERHEAD mission is in calculating the designator-to-target range. Rather than compute the designator-to-target range DR (or DR' in the second case) in the correct manner, the COPE model approximates the designator-to-target range by the distance PD . P is a point chosen on UD such that $UP = UR$ (or P' such that $UP' = UR'$ in the second case). This approximation is equivalent to the assumption that the target moves straight toward the designator.

One can see from the line segment lengths in Figure 3-1 that PD does not differ greatly from RD (in this case) whereas $P'D$ does differ considerably from $R'D$ in length. For typical range values and velocities used so far in COPE, these errors are considered minor. If one is to use greater target velocities or shorter acquisition ranges, then it would be worthwhile to improve this feature of COPE. (That is, incorporate the trigonometry required to correctly calculate DR (or DR') and use that value instead of PD (or $P'D$)).

Note: The designator operator-to-target distance is used not only at round arrival to calculate probability of engagement and probability of hit; but also throughout the simulation it is updated and used in checking bail-outs, direct fire suppression, line-of-sight probability for random occurrence, etc.

3.3.2 Target Modeling. The COPE model makes simplifying assumptions about the target. In particular, it assumes that the target is a column of vehicles moving at constant velocity with uniform spacing between successive vehicles. When a vehicle is killed, the length of the column is not reduced in length nor is any attempt made to keep a record of the position that the killed vehicle occupied in the column.

These limitations were not deemed serious in the COPE runs for the COPPERHEAD COEA; however, for the SADARM variant of COPE (called SOPE) the model was modified to keep track of individual vehicle status and to handle more general target layouts. (These modifications were essential to SADARM where there is no designator operator to pick out live targets instead of dead ones, and where the volley attacks an area instead of a point as with a single COPPERHEAD round).

3.3.3 Direct Fire Effects Independent. The model of direct fire effects of the target against the D.O. (see section 2.2.14) is oversimplified. The model has a single probability of kill (or obscuration) by direct fire for each target-to-D.O. range and this probability (corrected for then current range) is then applied during each COPPERHEAD round's firing to determine whether the D.O. is killed (or obscured) and thus prevented from lasing.

The shortcomings of this model are (1) subsequent rounds fired by the target at the D.O. should be more accurate both due to adjustment (assuming both that the D.O. survived previous rounds fired by the target and that the target survived previous COPPERHEAD's) by the target after seeing its first round impact and due to the D.O.'s giving away his position more times by the additional lasings, and (2) adjacent vehicles in the target unit, after seeing the lasings for the first few rounds, might join in firing at the D.O. on the latter rounds. Both effects (1) and (2) are omitted in COPE's model of direct fire against the D.O..

3.3.4 Only Initial Acquisitions (Unmaskings) Played. The COPE model only plays the first unmasking of each target unit. A more realistic assessment of COPPERHEAD's contribution would be obtained if the target unit were acquired (or, at least, unmasked to provide a potential acquisition) several times as it closes toward the BLUE position. This would require keeping track of the damage done to the RED target unit as the result of previous unmaskings (and the resulting COPPERHEAD fire missions) so as not to take credit for killing vehicles many times over. It would be a fairly straight forward task to modify the COPE program to allow for multiple acquisitions (and firing) of each target unit. The reason this was not done in the original COPE model is simply that all of the acquisition data (range to unmask and line-of-sight segment lengths) available to the model's creators were for first acquisition only.

3.3.5 Other Limitations. Other limitations to COPE are chiefly due to either inadequate data or to the desire to keep the model and program simple. These limitations include:

(a) Availability of artillery to fire the COPPERHEAD mission is not considered.

(b) The impact of the attrition of a D.O. or an F.O. (forward observer) on the ability of the direct support artillery to perform its mission is not considered. That is, how much of the lost D.O. capability could be

picked up by neighboring D.O.'s, other Fire Support Team (FIST) members, or platoon F.O.'s?

(c) The permanent loss of communications is not played. This could be partially simulated by reducing the probability of a successful transmission, although that value was intended to represent only degradation due to high communication traffic load and electronic warfare.

CHAPTER 4

4. INPUT DIRECTIVES AND "DATA" STATEMENTS

4.1 Input.

Considerable effort has gone into making the COPE input versatile enough to allow for the running of nearly any combination of input parameters while at the same time requiring a minimum of input changes to switch from one case to another. The stacking of multiple cases for a single run can be easily accomplished with relatively few input cards.

The existence of "default" (baseline case) values built into the program allows the program to be run with only the input cards (or card images) required to change those variables which it is desired to set at other than the default values. If further cases are desired one can continue with additional input changes applied to either the previous case or the baseline case.

The program also allows the insertion of up to three types of user comments which may appear in the output.

Finally, in the event that one wishes to use data not in the "data base" file, a "temporary" option using traditional formatted data is available.

4.2 Data Organization.

For each case that is run, there are numerous parameters whose values are set to provide the mathematical description of that case. These parameters are grouped into eight major blocks, six condition keys, and twenty odd individually selected numerically valued variables.

For the most part, a single input card is used to select each block, key, or variable value. There are some exceptions and some dependencies between choices which are explained below (section 4.8).

Finally, there are numerous data items whose values are set in the Block Data (BDATA1) subprogram because they are relatively fixed from case to case. Obviously, if one chose to vary these items frequently, they could be made into input rather than set via data statements. These items are defined below (section 4.7)

4.3 Keyword input: General Description.

Each input to select a case description (except when the "temporary" option is used) as well as various inputs to control program options is of the form:

keyword, option word 1, option word 2, ..., option word n
where $0 \leq n \leq 9$. Each such input must be contained on one card (or card image); no continuation to another card is allowed.

The keyword is an alphanumeric character string indicating which case descriptor is to be set by the card.

The option words are either alphanumeric character strings that indicates which option is chosen from a pre-defined set or numerical values that are to be used for certain variables.

Option words are order dependent (i.e., keyword, x, y is not equivalent to keyword, y, x). If an option word is omitted, a default value is set by the program. If a keyword takes two or more option-words and an option word other than the last one is omitted, then the proper number of separators (commas, for example) must be used to position the later optionwords (e.g.,

keyword, default value for option word 1, option word 2

is equivalent to

keyword, , option word 2

but not equivalent to

keyword, option word 2).

In most cases, blanks may be freely inserted within keywords or option words to improve readability. (The only exception is covered in the note of section 4.5.3.).

4.4 Keyword Input: Specifics.

Now each keyword is shown with the option words allowed. Note that new option words may be introduced by running the PREPMS pre-processor program to modify the data base file (section 5.3.8 and chapter 14).

4.4.1 FIRST CASE.

(a) general form: FIRST CASE

(b) This card is used only as the first non-comment card in an input deck. If it is omitted, the program will generate a message upon encountering the first non-comment card in the input and then proceed as though the FIRST CASE card were present.

4.4.2 NEXT CASE.

- (a) general form: NEXT CASE, option word 1
- (b) Currently allowed character strings for option word 1 are:
 - (1) RESET DEFAULTS or RD
 - (2) DON'T RESET or DR.
- (c) This card is the first non-comment card of each case other than the first case.

If the option word is RESET DEFAULTS, RD, or omitted, then the values for all case description parameters are reset to their default values and will differ from the baseline values only as determined by the remaining input cards for the current case.

If the option word is DON'T RESET or DR, then the values of the case description parameters remain at their values for the previous case except for those values changed by the remaining input cards for the current case.

If the NEXT CASE card is omitted, the program will generate a message and proceed as though a NEXT CASE, DR card had been encountered.

Note: The PLOT and CONTROL cards (sections 4.4.5 and 4.4.6) are not controlled by the option on the NEXT CASE card.

4.4.3 END.

- (a) general form: END
- (b) This card belongs at the end of the input for each case other than the final case.

If it is omitted, an error will usually result because the program will continue reading cards that are really part of the next case's input until it either hits an END card, ENDF card, or the end of the input file. This means that the case will run using its own input together with input from subsequent case (or cases). The user can detect this problem from the fact that the case descriptors will probably be incorrect (i.e., other than the intended values) in the case heading print-out, there may be unexpected messages stating that certain input cards have superseded others, and finally, the number of cases in the run will be wrong.

4.4.4 ENDF.

- (a) general form: ENDF or END OF FINAL CASE.

(b) This card should be the final card of the final case.

If it is omitted, the program will generate a message on reaching the end of input and then proceed as though an ENDF had been encountered.

If an END card is used instead, there will be no change in the result but the program will waste a few nano-seconds before realizing that there is no more input.

If an ENDF is used prior to the end of the last case, any cards following the ENDF card will be ignored.

4.4.5 PLOT.

(a) general form: PLOT, option word 1, option word 2

(b) Allowed character strings for option word 1 are:

(1) PRINTER

(2) CALCOMP.

Allowed character strings for option word 2 are:

(1) OFF

(2) ON.

(c) This keyword was designed to handle graphic output (bar charts), but has not been implemented in the current version of the program.

The cards PLOT, PRINTER, ON and PLOT, CALCOMP, ON would cause results to be displayed on bar graphs produced by the line printer and Cal-Comp plotter respectively. At the present time, however, these cards will only cause the program to generate a message saying that no plot was produced.

The cards PLOT, PRINTER, OFF and PLOT, CALCOMP, OFF turn off the production of the bar graphs produced by the line printer and Cal-Comp plotter respectively.

The default for option word 1 is PRINTER; the default for option-word 2 is OFF. So no bar graphs (or, at the present time, "no plot" messages) are produced unless positive action is taken by the user.

Note, that if and when this feature is implemented, the two types of graphs may be selected independently of each other.

A PLOT option once selected remains in effect until a further PLOT card is encountered changing the option (that is, ON to OFF or OFF to ON). It is not affected by a NEXT CASE, RD card.

4.4.6 CONTROL.

(a) general form: CONTROL, option word 1, option word 2

(b) Allowable choices for option word 1 are:

(1) ECHO

Allowable choices for option word 2 are:

(1) SHORT

(2) LONG.

(c) This keyword is designed to control miscellaneous program features. At the present time, the only choice (and the default) for option word 1 is ECHO. If option word 2 is SHORT or omitted, then the short form of the input echoing is used; if option word 2 is LONG, the long form of the input echoing is used (see section 7.6 for input echoing).

The default (which occurs if the card is omitted) is the short form.

A CONTROL option once selected remains in effect until a further CONTROL card is encountered changing the option (that is, LONG to SHORT or SHORT to LONG). It is not affected by a NEXT CASE, RD card.

4.4.7 LINE OF SIGHT MODE.

(a) general form: LINE OF SIGHT MODE, option word 1 or LOS MODE, option word 1

(b) Allowable choices for option word 1:

(1) SHOOTING GALLERY or SG

(2) RANDOM OCCURRENCE or RO.

(c) The choice of SHOOTING GALLERY or SG for option word 1 causes the current case to be run using the "shooting gallery" line of sight model. If option word 1 is RANDOM OCCURRENCE or RO, then the "random occurrence" line of sight model is used.

The default is the "shooting gallery" model.

4.4.8 MISSION CODE

- (a) general form: MISSION CODE, option word 1 or MSNCODE, option word 1
- (b) Allowable character strings for option word 1 are:
 - (1) PRIORITY PREPLANNED TARGET or PPPT
 - (2) PREPLANNED TARGET or PPT
 - (3) TARGET OF OPPORTUNITY or TOO.
- (c) If option word 1 is PRIORITY PREPLANNED TARGET, PPPT, or omitted, then the mission is considered priority preplanned; if option word 2 is PREPLANNED TARGET or PPT, then the mission is considered preplanned; and, if option word 2 is TARGET OF OPPORTUNITY or TOO, then the mission is considered a target of opportunity.

The mission type affects the response time parameters used (see section 4.4.15) and the modeling of range from unmask to centroid (RUMC, section 4.4.18). See section 4.8 for side effects of mission code choice.

4.4.9 NUMBER OF DIFFERENT TARGET POSTURES.

- (a) general form: NUMBER OF DIFFERENT TARGET POSTURES, option word 1 or NUMDIFTGTP, option word 1
- (b) Allowable character strings for option word 1 are:
 - (1) THREE or 3
 - (2) TWO or 2.
- (c) If option word 1 is THREE, 3, or omitted, the target exposure (mini-terrain) model plays three exposures or postures (fully exposed, hull defilade, and completely obscured).

If option word 1 is TWO or 2, the target exposure model plays two exposures (fully exposed and hull defilade) with the completely obscured divided between the other two postures in proportion to their relative fractions.

The default option word 1 is THREE.

4.4.10 TARGET TYPE.

- (a) general form: TARGET TYPE, option word 1
- (b) Allowable character strings for option word 1 are:

- (1) 1
- (2) 2
- (3) 3
- (4) 4
- (5) 5
- (6) 6
- (7) 7

- (c) If option word 1 is i, then the vulnerability values (probability of kill given a hit as a function of target exposure) for target type i are used.

The default target type is 1.

4.4.11 DESIGNATOR TYPE.

- (a) general form: DESIGNATOR TYPE, option word 1 or DESIGTYPE, option word 1
- (b) Allowable choices for option word 1 are:
 - (1) GLLD or G
 - (2) MULE or M
 - (3) LTD or L
- (c) If option word 1 is GLLD, G, or omitted, the designator type is the GLLD; if option word 1 is MULE or M, the designator type is the MULE; and, if option word 1 is LTD or L, the designator type is the LTD.

The default designator type is GLLD.

See section 4.8 for side effects of Designator Type Choice.

4.4.12 DO POSITION.

- (a) general form: DO POSITION, option word 1
- (b) Allowable choices for option word 1 are:
 - (1) VANTAGE POINT or VP
 - (2) MANEUVER UNIT or MU.

- (c) If option word 1 is VANTAGE POINT, VP, or omitted, then the target posture table gives break down into target exposure as seen from a vantage point.

If option word 1 is MANEUVER UNIT or MU, then a target posture table is used which gives the breakdown into target exposures as seen from a position down with the maneuver units.

The default for DO POSITION is VP.

4.4.13 WEATHER.

- (a) general form: WEATHER, option word 1, option word 2 or W, option word 1, option word 2
- (b) Allowable choices for option word 1 are:
- (1) JUNE or J
 - (2) DECEMBER or D
 - (3) MARCH or M
 - (4) SEPTEMBER or S

Allowable choices for option word 2 are:

- (1) 1400 or 14
 - (2) 0600 or 06
 - (3) 2200 or 22.
- (c) The weather data used in COPE is grouped into sets of data; each set corresponding to a certain month and time-of-day combination.

The choice for option word 1 determines the month whose weather data are to be used for the current case.

The choice for option word 2 determines the time-of-day whose weather data are to be used for the current case.

The defaults are JUNE or J for option word 1 and 1400 or 14 for option word 2.

See section 4.8 for side effects of weather choice.

4.4.14 ACQUISITION RANGE DISTRIBUTION.

- (a) general form: ACQUISITION RANGE DISTRIBUTION, option word 1 or ACQRNGDIST, option word 1

(b) Allowable choices for option word 1 are:

- (1) AVERAGE or A
- (2) CLOSE or C
- (3) OPEN or O.

(c) If option word 1 is AVERAGE, A, or omitted, then the acquisition range and line-of-sight segment length distributions used are those for TETAM "average" terrain. If option word 1 is CLOSE or C, then distributions for TETAM "close" terrain are used. If option word 1 is OPEN or O, then distributions for TETAM "open" terrain are used.

The default is to use TETAM "average" terrain.

See section 4.8 for side effects of acquisition data choice.

4.4.15 RESPONSE TIME.

(a) general form: RESPONSE TIME, option word 1, option word 2, option word 3, or RESPTIME, option word 1, option word 2, or option word 3

(b) Allowable choices for option word 1 are:

- (1) FTSILL or FS
- (2) STAUCH or S
- (3) SCUNGIO
- (4) PARAM

Allowable choices for option word 2 are:

- (1) DIGITAL or D
- (2) VOICE or V
- (3) a positive numerical value (which represents response time in seconds).

Allowable choices for option word 3 are:

- (1) DAY or D
- (2) NIGHT or N.

- (c) If option word 1 is FT SILL, FS, or omitted, then the response time distribution supplied by USAFAS at FT Sill is used. This plays the first way of modeling delay time as described in Section 2.2.9.

If option word 1 is STAUCH or S, the response time numbers are obtained from the BCS DT II for some processes and TACFIRE tests for others. These data were supplied by Ed Stauch of AMSAA and use the third way of modeling delay time as described in Section 2.2.9.

If option word 1 is SCUNGIO, then the response time distribution used is a hypothetical distribution created by Richard Scungio of AMSAA to test the sensitivity of COPPERHEAD performance to longer response time. In this case the third modeling of response time (see Section 2.2.9) is used.

If option word 1 is PARAM, then the second way of modeling response time delay (see Section 2.2.9) is employed (so-called "parameterized" response time). In this case, the response time is a constant equal to the positive numerical value used as option word 2.

If option word 2 is DIGITAL, D, or omitted, then digital communication is used (with voice back-up when using the third way of modeling in Section 2.2.9).

If option word 2 is VOICE or V, then voice communication is used (this will have an affect only when used with the third way of modeling in Section 2.2.9)

If option word 2 is a positive numerical value, then this value is used as the "parameterized" response time. This can only be used with the PARAM choice for option word 1. A zero or blank option word 2 value used with PARAM as option word 1 will cause the program to stop (see section 7.5.18).

(Note: If one really wishes to run a zero parameterized response time value, a value of .001 could be used. The difference between these results and those obtained with a true zero response time would be far below the noise in the model).

If option word 3 is DAY, D, or omitted, day time response times are used. If option word 3 is NIGHT or N then nighttime response times are used. In addition, the choice of day or night influences the target velocity and maximum designator range (see Sections 4.4.21 and 4.4.22).

The defaults are: for option word 1, FTSILL; for option word 2, DIGITAL; for option word 3, DAY.

See section 4.8 for side effects of response time data choice.

4.4.16 DIRECT FIRE SUPPRESSION.

- (a) general form: DIRECT FIRE SUPPRESSION, option word 1
or DFIRESUPPR, option word 1
- (b) Allowable choices for option word 1 are:
 - (1) HIGH or H
 - (2) NONE or N
- (c) If option word 1 is HIGH, H or omitted, then the direct fire suppression (obscuration from direct fire) and kill table used is for a high level of D.O. suppression and kill (corresponding to tank fire at the D.O.)

If option word 2 is NONE or N, then there is no effect of direct fire on the D.O..

The default is HIGH.

4.4.17 INVARIANT.

- (a) general form: INVARIANT, option word 1
- (b) Allowable choices for option word 1:
none
- (c) The only currently allowed choice for invariant data is the default. Provision for other choices is built into the program for possible future use and may be activated by appropriate preprocessor run(s).

4.4.18 RUMC.

- (a) general form: RUMC, option word 1
- (b) Allowable choices for option word 1 are:
 - (1) NO or N
 - (2) a numerical value.
- (c) The RUMC (which stands for range from unmask to centroid) value is the distance from the point at which the target breaks terrain mask to the centroid of the COPPERHEAD footprint.

If option word 1 is NO, N or omitted, then RUMC is assumed to vary with each replication in such a way that the centroid of the footprint is set at the predicted target (first vehicle) location for first round arrival. This predicted location is based on the nominal (estimated) response time (including time-of-flight) and the target velocity.

If option word 1 is a numerical value, then RUMC remains at that value for each replication.

The RUMC keyword has an effect only when the MISSION CODE is PPPT or PPT (see Section 4.4.8). If the MISSION CODE is TOO, then the program behaves as though option word 1 were NO regardless of what may be input on a RUMC card.

The default for RUMC is NO.

4.4.19 SMOKE.

(a) general form: SMOKE, option word 1, option word 2, option word 3

(b) Allowable choices for option word 1:

(1) SMOKE ROUNDS FIRED or SMKRND

(2) PROBABILITY SMOKE KILLS MISSION or PRBSMOKE.

Allowable choices for option word 2:

(1) a numerical value

Allowable choices for option word 3:

(1) a numerical value.

(c) If option word 1 is SMOKE ROUNDS FIRED, SMKRND, or omitted, then option words 2 and 3 respectively give the numbers of type 1 smoke rounds fired and type 2 smoke rounds fired.

If option word 1 is PROBABILITY SMOKE KILLS MISSION or PRBSMOKE, then option word 2 gives the probability that smoke kills the potential COPPERHEAD fire mission.

The default for option word 1 is SMKRND. The default for option word 2 is 484 when option word 1 is SMOKE ROUNDS FIRED or SMKRND, but it is .40 when option word 1 is PROBABILITY SMOKE KILLS MISSION or PRBSMOKE.

The default for option word 3 is 846 when option word 1 is SMOKE ROUNDS FIRED or SMKRND; it is unused otherwise.

Note: When the first alternative for SMOKE option word 1 is chosen, the probability that smoke aborts the potential COPPERHEAD mission is calculated as a function of the number of smoke rounds of each type fired and the weather conditions.

When the second alternative is used, the value input for option word 2 (a probability between 0 and 1) is used as the probability that smoke aborts the potential COPPERHEAD mission. (See section 2.2.6)

4.4.20 DO ARTILLERY PK.

- (a) general form: DO ARTILLERY PK, option word 1 or DOARPK, option word 1
- (b) Option word 1 must be either a numerical value between 0 and 1 or omitted.
- (c) The numerical value input for option word 1 is used as the probability that the D.O. is killed by RED preparatory artillery fire.

The default value is .01.

4.4.21 TARGET VELOCITY.

- (a) general form: TARGET VELOCITY, option word 1 or TGTVEL, option word 1
- (b) Option word 1 is a positive number which gives the target velocity in meters per second. (The program is currently set up to accept only 2, 3, 5, 8, or 9 as allowed velocity).
- (c) The value of option word 1 is used as the target velocity.

The default target velocity is obtained from a table (part of the acquisition range data block) giving target velocities as a function of day or night, weather, and terrain type.

Note: For side effects of velocity choice, see section 4.8.

4.4.22 MAXIMUM DESIGNATOR RANGE.

- (a) general form: MAXIMUM DESIGNATOR RANGE, option word 1 or MAXDESRNG, option word 1
- (b) Option word 1 is a positive number giving the maximum designator range in meters. This is the designator range mentioned in section 2.2.4.

The default maximum designator range is obtained from a table giving ranges as a function of designator type and day or night.

Note: For side effects of designator range value, see section 4.8.

4.4.23 BAIL OUT RANGE.

- (a) general form: BAIL OUT RANGE, option word 1 or BAILOUTRNG, option word 1
- (b) Option word 1 is a positive number giving the bail-out range in meters (see section 2.2.8 and 2.2.10). The default bail-out range is 1000m.

4.4.24 TIME OF FLIGHT.

- (a) general form: TIME OF FLIGHT, option word 1 or TOF, option word 1
- (b) Option word 1 is a positive number giving the time of flight of the COPPERHEAD round in seconds.

The default value for time of flight is obtained from a table giving times of flight as a function of gun-to-target range.

Note: For side effects of time of flight value, see section 4.8.

4.4.25 ANGLE T.

- (a) general form: ANGLE T, option word 1
- (b) Option word 1 is a number giving the angle (in degrees) between the gun-to-target line and the designator-to-target line. The default value is 25 degrees.

Note: For side effects of angle T value, see section 4.8.

4.4.26 DUST.

- (a) general form: DUST, option word 1
- (b) Option word 1 is a number from 0 to 1 whose value is the probability that dust kills the potential COPPERHEAD mission.

The default value is .37.

4.4.27 PROBABILITY OF SUCCESSFUL VOICE TRANSMISSION.

- (a) general form: PROBABILITY OF SUCCESSFUL VOICE TRANSMISSION, option word 1 or PRBVOTRAN, option word 1

(b) Option word 1 is a number from 0 to 1 whose value is the probability that voice communication is established between the D.O. and the battery FDC.

The default value is .975.

4.4.28 PROBABILITY OF SUCCESSFUL DIGITAL TRANSMISSION.

- (a) general form: PROBABILITY OF SUCCESSFUL DIGITAL TRANSMISSION, option word 1 or PRBDGTTRAN, option word 1
- (b) Option word 1 is a number from 0 to 1 whose value is the probability that digital communication is established between the D.O. and the battery FDC.

The default value is .915.

4.4.29 ROUND IN FLIGHT RELIABILITY.

- (a) general form: ROUND IN FLIGHT RELIABILITY, option word 1 or RNDFLTREL, option word 1
- (b) Option word 1 is a number between 0 and 1 whose value is the probability that the round is reliable (functions properly).

The default value is .96.

4.4.30 TIME BETWEEN ROUNDS.

- (a) general form: TIME BETWEEN ROUNDS, option word 1 or TBR, option word 1
- (b) Option word 1 is a number whose value is the time (in seconds) between the firing of successive COPPERHEAD rounds.

The default value is 20 seconds.

4.4.31 NUMBER OF REPLICATIONS.

- (a) general form: NUMBER OF REPLICATIONS, option word 1 or NUMREP, option word 1
- (b) Option word 1 is the number of replications (i.e., Monte Carlo sample size of potential COPPERHEAD fire missions) to be used for the current case.

4.4.32 NUMBER OF ROUNDS TO BE FIRED.

- (a) general form: NUMBER OF ROUNDS TO BE FIRED, option word 1 or NUMRND, option word 1

(b) Option word 1 is the number of rounds to be fired on each potential COPPERHEAD fire mission of this case that reaches the point where firing occurs (i.e., a "shot" event). Since each round is fired and completes its flight (successfully or not) before the next round arrives, one may choose to think of the number of rounds to be fired as the number of one round volleys to be fired.

The default value is 6 rounds.

4.4.33 NUMBER OF VEHICLES PER TARGET.

- (a) general form: NUMBER OF VEHICLES PER TARGET, option word 1 or NUMVEHTGT, option word 1
- (b) Option word 1 is the number of vehicles in the target column.

The default value is 10 vehicles.

4.4.34 DISTANCE BETWEEN VEHICLES.

- (a) general form: DISTANCE BETWEEN VEHICLES, option word 1 or DISTBVEH, option word 1
- (b) Option word 1 is the mean distance (in meters) between adjacent vehicles in the target column.

The default value is 61 meters.

4.4.35 GUN TARGET RANGE.

- (a) general form: GUN TARGET RANGE, option word 1 or GTRNG, option word 1
- (b) Option word 1 is the gun-to-target range in kilometers.

The default value is 8 kilometers.

Note: For the side effects of gun-to-target range, see section 4.8.

4.4.36 REFLECTIVITY.

- (a) general form: REFLECTIVITY, option word 1 or REFLECT, option word 1
- (b) Option word 1 is a number between 0 and 1 whose value is the reflectivity of the target to laser energy (i.e., it is the fraction of the laser energy at the designator's frequency that is reflected by the target).

The default value is .10.

Note: The side effects of reflectivity are discussed in section 4.8.

4.4.37 DEFLECTION BIAS.

- (a) general form: DEFLECTION BIAS, option word 1 or DEFB,
option word 1
- (b) Option word 1 is the value in meters of the deflection bias which is to be defined as the minimum distance from the target column's path (a line) to the footprint centroid (a point).

The default value is 0.

Note: The side effects of deflection bias are discussed in section 4.8.

4.4.38 TARGET HEADING.

- (a) general form: TARGET HEADING, option word 1 or TGTHDG,
option word 1
- (b) Option word 1 is a number whose value is the angle (in degrees) between the target's line of travel and the target-to-gun line.

The default value is 0.

Note: The side effects of target heading are discussed in section 4.8.

4.4.39 SEEKER SENSITIVITY.

- (a) general form: SEEKER SENSITIVITY, option word 1 or
SEEKSENS, option word 1
- (b) Option word 1 is a number whose value is the minimum amount (threshold value) of energy per area (in joules per square meter) that must reach the COPPERHEAD seeker in order to initiate maneuver.

The default value is 60 joules/m². (The true value is not used as the default to avoid any possible classification problem.)

Note: The side effects of the seeker sensitivity value are discussed in section 4.8.

4.4.40 PROBABILITY OF A CORRECT MESSAGE.

- (a) general form: PROBABILITY OF A CORRECT MESSAGE, option word 1 or PROBCORMSG, option word 1
- (b) Option word 1 is a number whose value is the probability that the D.O. sends correct information to the FDC and the FDC interprets it correctly.

The default value is .975.

4.4.41 PROBABILITY DO WARNED TO LASE.

- (a) general form: PROBABILITY DO WARNED TO LASE, option word 1 or PROBDOWARN, option word 1
- (b) Option word 1 is a number whose value is the probability that the message warning the D.O. to begin lasing for the first COPPERHEAD round is transmitted and received successfully.

The default value is .975.

4.5 Special keywords: OVERRIDE, TEMPORARY, RESET.

4.5.1 Correspondence of Data Blocks to Keywords. As mentioned previously, a large part of the data used in each case is grouped into eight blocks. Each of these blocks is normally filled by reading a record from a word addressable mass storage file that is created by the PREPMS and PAM programs. The choice of which particular record to read from the file to fill each data block depends on the various keywords and option words for the case.

This correspondence of data blocks to keywords can be summarized as follows:

Record to be read for data block of this type	Depends on options for this (these) keywords
(1) Weather	WEATHER
(2) Acquisition range data	ACQRNGDIST
(3) Response time data	RESPTIME
(4) Direct fire suppression data	DFIRESUPPR
(5) Random occurrence data	ACQRNGDIST and TGTVEL
(6) Probability of engagement data	RESPTIME, MSNCODE, DESIGTYPE, TGTVEL, GTRNG, REFLECT, ANGLE T, DEFLB, TGTHDG, SEEKSENS

- (7) Target posture distribution FO POSITION
- (8) Invariant data INVARIANT

As can be seen, there is but one keyword influencing the choice of record to be read for each data block except in the cases of random occurrence data and probability of engagement data.

4.5.2 OVERRIDE and TEMPORARY Features. Two additional methods of filling in a data block are provided. The first is to choose a record from the word addressible mass storage file to be read into the data block without regard to the options chosen for the keyword(s) that would normally determine the record choice; this is the OVERRIDE feature. The second is to actually supply alternative data as traditional formatted FORTRAN inputs to be read into the data block; this is the TEMPORARY feature.

The OVERRIDE option is provided primarily so that one can choose to use probability of engagement (PE) data that has been generated with one set of conditions under other, not too different, conditions without having to re-run the PAM program. For example, PE data for a case with nominal response time, 130 sec. could be used to run a case identical (in those factors affecting PE) except for a nominal response time of 150 sec. The advantage of this procedure is that one can avoid re-running the relatively expensive PAM program in those cases where the factors affecting PE have changed very little. The disadvantage, of course, is that there are no efficient objective criteria available to determine whether the errors introduced by using wrong (but, it is hoped, approximate) PE data are significant in a given situation.

The TEMPORARY option is provided so that one can directly load data into any data block without having to use the PREPMS preprocessor program. This option allows one to use data for a given run without running the PREPMS program to make that data a "permanent" record in the data base file. This option was provided for flexibility, but was not exercised at all in the original COPPERHEAD Operational Performance Evaluation done for the COEA.

4.5.3 Using the OVERRIDE.

- (a) general form: OVERRIDE , keyword, option word 1 or
OVERIDE, keyword, option word 1
- (b) Choices for keyword are:
 - (1) WEATHER or W
 - (2) ACQUISITION RANGE DISTRIBUTION or ACQRNGDIST
 - (3) RESPONSE TIME or RESPTIME

- (4) DIRECT FIRE SUPPRESSION or DFIRESUPPR
- (5) RANDOM OCCURRENCE DISTRIBUTION or RODIST
- (6) PEDATA
- (7) TARGET POSTURE DISTRIBUTION or TGTPSTDIST
- (8) INVARIANT

Option word 1 is the name of any record (on the random access mass storage data base file) containing data for the data block type specified by the keyword choice.

- (c) The OVERRIDE causes the selected data block (or blocks, if more than one OVERRIDE card is used in a case) to be filled with data from the record whose name is option word 1.

There is no default for OVERRIDE. An error will result from the omitting of either the keyword or option word 1.

Note: If the record name includes one or more blanks or separators then the record name must be preceded by a "\$" (dollar sign). The ten characters forming the record name must immediately follow the dollar sign, any blanks in the name must be included in their proper positions, and no excess blanks may be inserted. For present purposes, separators include commas, slashes, question marks, and parentheses.

4.5.4 Using TEMPORARY.

- (a) general form: TEMPORARY, keyword or TEMP, keyword
- (b) Choices for keyword are as in section 4.5.3.
- (c) The use of the TEMPORARY option will cause the program to begin reading data in FORTRAN formatted fashion. Hence, each use of the TEMPORARY option must be immediately followed by the formatted data cards to be read to fill the data block selected by the keyword.

The formats to be used and the data to be supplied for each block are identical to those used by the PREPMS program for the corresponding block (see section 14.4.2).

Note: If the keyword is PEDATA, then one has the further option of reading the formatted cards from a separate file (TAPE 4). To use this option, a card: USE TAPE 4 must be the next card after the TEMPORARY, PEDATA card; then the formatted data should be on a file to be read by I/O unit 4.

There is no default for TEMPORARY.

4.5.5 RESET.

- (a) general form: RESET, keyword
- (b) Allowable keywords are as in section 4.5.3.
- (c) This card will reset the default conditions for the keyword chosen. Its use is mainly to undo an OVERRIDE or a TEMPORARY without having to use NEXT CASE, RD which might reset other defaults not desired.

Note: The effect of an OVERRIDE or TEMPORARY used on PEDATA or RANDOM OCCURRENCE DISTRIBUTION (RODIST) can be undone only by a RESET, a NEXT CASE, RD, or another OVERRIDE or TEMPORARY for the same data block.

The effect of an OVERRIDE or TEMPORARY for any other data block can also be undone by a card of the usual form for that data block's keyword (i.e., keyword, option word 1, option word 2, -----).

There is no default on the RESET card.

4.6 Comments.

4.6.1 General Remarks on Comments. The program allows three types of user comment cards to be included in the input. In addition, the program itself generates some comments under certain conditions. The program generated comments are detailed in sections 7.1 and 7.2.

User comments in the input card deck (or card image file) are identified by having "\$" (dollar sign) in the first one, two, or three columns of the card.

4.6.2 \$ Comments. Cards having a "\$" in column 1 but not in column 2 are user comment cards which are totally ignored by the program. When one is read, the program merely goes to the next input card and reads that.

This type of comment is provided only so that the user can leave reminders, separate sections of input, or otherwise label inputs without having an effect on program printouts.

4.6.3 \$\$ Comments. Cards having "\$" in columns 1 and 2 but not in column 3 are user comment cards which are reproduced within the current case's printout only.

The comment (contents of the card) will be reproduced (less the first two columns) with the symbols "*** USER ***" preceding it in the comment section of the case's printout. (See sample case, Appendix D.)

These comments should be used to note any special details about a case.

4.6.4 \$\$\$ Comments. Cards having "\$" in columns 1, 2, and 3 are user comments that are printed out immediately upon being encountered by the program. Again the symbols "*** USER ***" are appended to the front of the comment and the \$'s in columns 1-3 are removed.

These comments are printed on a special page separate from any case's printout. It was originally intended that "\$\$\$" comments be put at the beginning of the input before the FIRST CASE card; however, wherever they are put a special page will be printed containing them, and this page will occur in the printout immediately after the previous case's printout; or, (if there has been no case run yet) immediately after the title page.

4.7 DATA Statement Data.

There are many data which are set in data statements rather than read in from cards or disk file with each run or case.

As a general rule, these data are either labels (alphanumeric data) or numbers that do not change from run to run.

All of these data statements have been collected into the BLOCK DATA sub-program BDATA1. They are now described common block by common block:

4.7.1 ABRLBL. This common block contains the abort labels (i.e., alphanumeric texts) that identify the abort causes on the printouts. They reside in the IDABRT array and are entered as Hollerith constants.

4.7.2 ACHAR. This common block consists of the CHAR array which contains the keywords used to input COPE case descriptors. Each keyword is entered as a string of Hollerith constants. In addition, corresponding to each keyword character string are two numbers, the first of which is the keyword number and the second of which is the number of distinct option word character strings allowed with that keyword. (Note: The second number as entered in the data statement line for each keyword is overwritten by numbers taken from the record named OPTNNUMS on the random access mass storage data base file. Thus, when the number of distinct optionword character strings allowed with a given keyword changes, there is no need to adjust the data statement; the pre-processor program PREPMS is used to adjust the value in OPTNNUMS to the correct value. (See section 14.5.)

Also, NREC, the total number of keyword character strings, is set in this common block's data statements.

4.7.3 DESRNG. This common block contains the DESRNG array which consists of the default values for maximum designator range.

DESRNG (I, J) is the maximum designator range for designator type I under condition J where J=1 for day and J=2 for night.

4.7.4 DISPLY. This common block contains the DISPT, DISPM, RECLBL, and TMLBL arrays, all of which are defined in the variable glossary (see chapter 12). Each of these arrays consists of alphanumeric constants entered as Hollerith data.

4.7.5 FLTTIM. This common block consists of the TOFARY array. The array gives the default times of flight: TOFARY (I) is the COPPERHEAD time of flight for a gun-to-target range of I (currently only I=8 and I=12 are used).

4.7.6 FSRESP. This common block consists of the FSRT array which contains the FT Sill supplied response time distribution.

4.7.7 HEADNG. This common block contains the ACQLBL, DAYLBL, DFSLBL, DGTLBL, DOLBL, ROLBL, and RSPLDL arrays all of which are defined in the glossary of variables (chapter 12).

Each of these arrays consists of alphanumeric constants entered as Hollerith data.

4.7.8 LOGFLG. This common block contains various logical flags: FIRSTL, SEQNML, SPCL array, and SHRTEC all of which are defined in the glossary of variables (chapter 12).

These are all logical type variables.

4.7.9 POINT. This common block contains the IPOINT array used in Subroutine INPUT to direct control to various parts of a computed GOTO.

4.7.10 SMOKED. Only the SMK2 and SMK5 arrays plus the variable FRONT of this common block are filled in via data statements. They are all defined in the variable glossary (chapter 12).

4.7.11 STITLE. This common block contains SLVERS and the VERDAT array. Both are used in the title page printout.

4.7.12 SYMBOL. This common block consists of the ALFBET, ANUMBR, and SEP arrays plus the single variables BLANK, DECPNT, and DOLLAR. They are all alphanumeric constants filled in as Hollerith data and defined in the glossary of variables (chapter 12).

4.7.13 XVALUE. This common block contains the XVALUE array which is used to generate PEDATA record names. (See sections 5.4 and 9.34.)

4.8 Side Effects of Certain Inputs.

There are a number of dependencies and side effects connected with some of the inputs.

First, the choice of response time data (4.4.15) and mission code (4.4.8) together with time-of-flight determine the nominal response time. (Time-of-flight itself may be determined either by direct input (4.4.24) or as a result of the choice of gun-to-target range (4.4.35)).

The nominal response time, designator type (4.4.11), target velocity (4.4.21), angle T (4.4.25), gun-to-target range (4.4.35), reflectivity (4.4.36), deflection bias (4.4.37), target heading (4.4.38), and seeker sensitivity (4.4.39) are all used to determine the name of the PE data record (see section 5.4). Indeed, angle T, reflectivity, deflection bias, target heading, and seeker sensitivity play no other roles in COPE than to affect the PE data (by determining the PE data record name) and to appear on the case heading page (see section 6.3) for identification purposes.

The maximum designation range (4.4.22) depends on the time of day (day or night) input with response time data (4.4.15) and the designator type (4.4.11). The maximum designator range can also be set by direct input (4.4.22); however, the user must take care to see that the PE data as well as TTF and RGTTF arrays contain values appropriate for the maximum designator range used.

The target velocity (4.4.21) can be input directly or as a result of the time of day (day or night) used for the response time data (4.4.15), the month used for weather (4.4.13), and the terrain type (4.4.14) used for acquisition data. (The VELTBL array may need adjusting if additional choices are allowed for acquisition data or weather.)

Finally, the weather data (4.4.13) has a strong side effect on the effectiveness of smoke when the SMOKE option (4.4.19) that plays a specified number of smoke rounds is used.

CHAPTER 5

5. PREPARING TO RUN COPE

Prior to running the main COPE program, it is necessary to run the preprocessor programs to create the word addressable mass storage data base file (TAPE 11). This chapter describes the role of each preprocessor, the structure of TAPE 11, and the interfacing of TAPE 11 with each of the programs by means of the record names.

5.1 Preprocessor Programs.

There are three programs that may be regarded as preprocessors to the main COPE simulation program. These three programs are called:

- (1) PREPMS which stands for preprocessor mass storage,
- (2) PRBLOS which stands for probability of line-of-sight, and
- (3) PAM which stands for probability of seeker acquisition and round maneuver.

More detailed information on PREPMS, PRBLOS, and PAM is included in chapters 14, 15, and 16 respectively of this report. In addition, more thorough documentation of PAM is to be published in a future separate AMSAA Technical Report.

As mentioned in chapter 4 of this report, much of the data used in each COPE case is grouped into eight data blocks. These data blocks are filled by reading records from TAPE 11 (except in the case of the TEMPORARY option described in section 4.5). The main purpose of the PREPMS program (and one of the purposes of the PAM program) is to fill values into the records on TAPE 11 in such a way that they are accessible to the main COPE program.

The PREPMS program is used to fill in records for data blocks for weather data, acquisition range data, response time data, direct fire suppression data, random occurrence data, target posture data, and invariant data. It can also be used to fill in records for PE (probability of engagement) data, but this feature has never been used because the PAM program which generates PE data has been modified to write its results directly to TAPE 11 and the choice of record name has been entirely automated in this case. Finally, the PREPMS program is also used to establish and modify the option words allowed as choices with each keyword (described in sections 4.4.1 through 4.4.19 except for those option word choices described as "a numerical value" in sections 4.4.18 and 4.4.19).

The PRBLOS program is used to calculate the probabilities of line-of-sight at various ranges. These values are then used in the random occurrence line-of-sight model. They are read from the PRBLOS printout and entered by hand as input to PREPMS. Because the PRBLOS program does not directly act on TAPE 11, it shall not concern us further in this chapter (see chapter 15 for complete discussion of PRBLOS).

The PAM program takes as input the parameter values describing the conditions under which probability of engagement is to be computed. It then computes the PE data values, creates a record name for its current block of PE data by encoding the input conditions according to a scheme also used in the main COPE program, and then writes the PE data to TAPE 11 under that record name.

5.2 Word Addressable Mass Storage Data Base File (TAPE 11).

The word addressable mass storage data base file (TAPE 11) consists of the records containing the data to be used by each of the eight data blocks mentioned in chapter 4, the records of option words allowed with each keyword (applies only to keywords described in sections 4.4.1 through 4.4.19 and excludes any option words described as "a numerical value"), and a record called OPTNNUMS that records the number of option word character strings allowed for each keyword.

TAPE 11 is a CDC word addressable mass storage file using a name type master index. (This file type and the way it is interfaced with a FORTRAN program is explained in the CDC Fortran Extended Version 4 Reference Manual chapter 8, a section titled "Mass Storage Input/Output"). The important feature to note here is that this file type (with a name type master index) is organized as a set of records each consisting of one or more words of data and each having a unique record name of one to ten characters. This unique record name is set when the record is created and must be used by any program that either reads or modifies the record. It is by using the same system of record names in the COPE program as in the preprocessors PREPPE and PREPMS that the data records written by the preprocessors can be read by the main COPE program.

5.3 Record Names.

5.3.1 Weather Data Record Name. The record name of the data to be used in the weather data block (the WDATA array) has the form : mmmmtttt021 where mmm are the first three letters of the name of the month of the weather data, tttt are four numbers giving the time of day (24 hour military time) of the weather data, and the 021 are three numbers signifying that WEATHER has keyword number 21.

Examples:

JUN0600021 is the record name for weather data for a
June day at 0600 (6 AM).

SEP2200021 is the record name for weather data for a September day at 2200 (10 PM).

5.3.2 Acquisition Data Record Name. The record name of the data to be used in the acquisition range data block (the ACQDAT array) has the form: ACQDATnn22 where nn are two digits giving the number of the choice for option word 1 used with keyword 22 (section 4.4.14).

Examples:

ACQDAT0122 is the record name of the acquisition range and LOS segment length data to be used when option word AVERAGE is used with keyword 22 (ACQRNGDIST)

ACQDAT0322 is the record name of the data to be used with OPEN terrain.

5.3.3 Response Time Data Record Name. The record name of the data to be used in the response time data block (RSPDAT array) is of the form: RSPDATnn23 where nn are two digits giving the number of the choice for option word 1 used with keyword 23 (section 4.4.15)

Examples:

RSPDAT0123 is the name of the record containing the response time data to be used when the FTSILL option is selected.

RSPDAT0223 is the name of the record containing the response time data for the STAUCH option.

Note: There is no data record corresponding to the PARAM choice for option word 1. This is because the PARAM choice for option word 1 causes the program not to require any of the data in the response time data block.

5.3.4 Direct Fire Suppression Data Record Name. The record name of the data to be used in the direct fire suppression data block (DFSDAT array) is of the form: DFSDATnn24 where nn are two digits giving the number of the choice for option word 1 used with keyword 24 (section 4.4.16)

Example:

DFSDAT0124 is the record name for HIGH direct fire suppression data.

5.3.5 Random Occurrence Data Record Name. The record name of the data to be used in the random occurrence data block (RODATA array) is of the form: RODAVmTRnn where m is the target velocity in meters per second (obviously between 0 and 9) and n is the terrain type number (corresponds

to number of option word 1 on ACQRNGDIST card, i.e., 1 for AVERAGE, 2 for CLOSE, 3 for OPEN).

Example:

RODAV8TRN2 is the record name for the random occurrence data to be used when target velocity is 8 m/s and terrain type is 2 (CLOSE).

Note: The program is currently set up only for velocities of 2, 3, 5, 8, and 9. Furthermore, 2 and 8 are treated as 3 and 9 respectively when it comes to looking up RODATA and PEDATA.

5.3.6 Target Posture Distribution Data Record Name. The record name of the data to be used in the target posture data block (PSTDAT array) is of the form: PSTDATnn27 where nn is the number of the choice for option word 1 used with keyword 17 (section 4.4.12).

Example:

PSTDAT0127 is the record name for the target posture distribution used when FO POSITION is VANTAGE POINT.

Note: Keyword 17 (FO POSITION) and keyword 27 (TARGET POSTURE DISTRIBUTION) could each be used to control the choice of target posture distribution data, but only keyword 17 (FO POSITION) has been provided with option words and is the only one of the two explained in section 4.4.

5.3.7 Invariant Data Record Name. The record name of the data used in the invariant data block (AINVDA array) is of the form: AINVDAAnn28 where nn is the number of the choice for option word 1 used with keyword 28.

Example:

AINVDA0128 is the record name for the invariant data used with the default (omitted) option word 1.

Note: The keyword INVARIANT is currently set up to accept no option words (other than a blank which is equivalent to the default).

5.3.8 Option Word Record Names. For each keyword numbered less than 41 (see section 5.3.9 for keyword numbers) there may be some option words that are not numerical values (see sections 4.4.1 through 4.4.19 for the current option word choices available with each keyword).

To create, modify, or add to the list of character strings that can be used with any keyword, it is necessary to run PREPMS with input specifying the keyword (in its abbreviated form of ten characters or less), the total number of option word character strings that can be used

with that keyword and the keyword number as well as the option word number, the option word choice number and the option word character string for each option word choice. The formatting of this data is explained in chapter 14, but an explanation of its meaning is provided by the sample input following:

```
OPTION WORDS, W, 10, 21
1, 2, JUNE
1, 2, J
2, 2, DECEMBER
2, 2, D
1, 3, 1400
1, 3, 14
2, 3, 0600
2, 3, 06
3, 3, 2200
3, 3, 22
```

where the first line indicates that option word allowable choices are being set for keyword W (or WEATHER), that ten optionword character strings are being input, and that W (or WEATHER) is keyword number 21.

Each of the next ten lines establishes a character string as a possible option word choice. For example, 1, 3, 1400 establishes 1400 as a character string corresponding to the first choice for option word 2 of any input line beginning with keyword W (or WEATHER). The first digit (1) indicates that 1400 is the first choice allowed, the second digit 3 indicates that 1400 is a choice allowed for the third item on the input line (recall that an input line is of the form: Keyword, option word 1, option word 2, - - -, option word n so that option word 2 is actually the third item on an input line), and, of course, 1400 indicates that the input character string to select option 1 for option word 2 is 1400. The fact that the next line is 1, 3, 14 means that 14 has the same effect (first choice for option word 2) as 1400. No more than two lines are allowed to be input with the same ordered pair (choice number, option word number + 1) under any given keyword (though one may input just one line for such a pair if desired).

The PREPMS program then places this option word information on a record whose record name is the keyword (W in the case above). It is for this reason that the keyword used must be ten or fewer characters. Each keyword of length more than ten characters also has an abbreviated form (see the appropriate section of chapter 4 for each keyword; even some keywords of length less than ten characters have abbreviated forms). For example, the keyword ACQUISITION RANGE DISTRIBUTION is suitable for input to COPE, but cannot be used as input to PREPMS because it is too long to be a record name; instead, the abbreviated form ACQRNGDIST must be used when setting option words via PREPMS.

The rule is always to use the abbreviated form of the keyword with PREPMS input (even if the long form of the keyword is itself ten or fewer characters) and to use either form with input to COPE. For keywords with only one form, use it as input to both programs.

Finally, for each new option word choice that is allowed, the user must consider what COPE will do when it encounters the new option word. In the case of option words that control the choice of records to be read from TAPE 11 into data blocks, the user must also have run PREPMS to add the possible data block records into TAPE 11. In our example above, input to PREPMS must have created records named JUN0600021, JUN1400021, JUN2200021, DEC0600021, DEC1400021, and DEC2200021 containing the weather data to be used with each of the possible option word combinations. For the case of option words that control program actions rather than select a data record name, it will usually be necessary to modify the program code so that it will "know" what to do when encountering a new option.

5.3.9 Keyword Numbers, Abbreviated Forms, Etc. The following table gives the current list of allowed keywords, each keyword's number, abbreviated form, and relevant notes. (This overlays some of the information in chapter 4, but it was felt desirable to have all the keywords listed in one place for quick reference; chapter 4, sections 4.4.1 - 4.4.38 are still essential for allowed optionwords and detailed effects of keywords). When two consecutive keywords have the same number, the second is the abbreviated form. (Recall that blanks may be arbitrarily inserted in (or removed from) keywords.) Notes are explained at end of listing.

<u>Keyword Number</u>	<u>Keyword</u>	<u>Notes</u>
1	FIRST CASE	*1 Sec. 4.4.1
2	NEXT CASE	*2 Sec. 4.4.2
3	END OF CASE	*1 Sec. 4.4.3
3	END	*1 "
4	END OF FINAL CASE	*1 Sec. 4.4.4
4	END F	*1 "
5	PLOT	*2 Sec. 4.4.5
6	OVERRIDE	*3 Sec. 4.5.3
6	OVERIDE	*3 "
7	TEMPORARY	*3 Sec. 4.5.4
7	TEMP	*3 "

<u>Keyword Number</u>	<u>Keyword</u>	<u>Notes</u>
8	RESET	*3 Sec. 4.5.5
9	CONTROL	*2 Sec. 4.4.6
11	LINE OF SIGHT MODE	*2 Sec. 4.4.7
11	LOS MODE	*2 "
12	MISSION CODE	*2, *9 Sec. 4.4.8
12	MSN CODE	*2, *9 "
13	NUMBER OF DIFFERENT TARGET POSTURES	*2 Sec. 4.4.9
13	NUMDIFTGTP	*2 Sec. 4.4.9
14	TARGET TYPE	*2 Sec. 4.4.10
16	DESIGNATOR TYPE	*2, *9 Sec. 4.4.11
16	DESIGTYPE	*2, *9 "
17	FO POSITION	*4 Sec. 4.4.12
21	WEATHER	*4 Sec. 4.4.13
21	W	*4 "
22	ACQUISITION RANGE DISTRIBUTION	*4 Sec. 4.4.14
22	ACQRNGDIST	*4 "
23	RESPONSE TIME	*2, *4, *9 Sec. 4.4.15
23	RESPTIME	*2, *4, *9 "
24	DIRECT FIRE SUPPRESSION	*4 Sec. 4.4.16
24	DFIRESUPPR	*4 "
25	RANDOM OCCURRENCE DISTRIBUTION	*10
25	RODIST	*10
26	PEDATA	*10
27	TARGET POSTURE DISTRIBUTION	*10

<u>Keyword Number</u>	<u>Keyword</u>	<u>Notes</u>
27	TGTPSTDIST	*10
28	INVARIANT	*5 Sec. 4.4.17
31	RUMC	*6 Sec. 4.4.18
32	SMOKE	*7 Sec. 4.4.19
41	DO ARTILLERY PK	*8 Sec. 4.4.20
41	DOARPK	*8 "
42	TARGET VELOCITY	*9 Sec. 4.4.21
42	TGTVEL	*9 Sec. 4.4.21
43	MAXIMUM DESIGNATOR RANGE	*8 Sec. 4.4.22
43	MAXDESRNG	*8 "
44	BAIL OUT RANGE	*8 Sec. 4.4.23
44	BAIL OUTRNG	*8 "
45	TIME OF FLIGHT	*8 Sec. 4.4.24
45	TOF	*8 "
46	ANGLE T	*9 Sec. 4.4.25
47	DUST	*8 Sec. 4.4.26
48	PROBABILITY OF SUCCESSFUL VOICE TRANSMISSION	*8 Sec. 4.4.27
48	PRBVOCTRAN	*8 "
49	PROBABILITY OF SUCCESSFUL DIGITAL TRANSMISSION	*8 Sec. 4.4.28
49	PRBDGTTRAN	*8 "
50	ROUND IN FLIGHT RELIABILITY	*8 Sec. 4.4.29
50	RNDFLTREL	*8 "
51	TIME BETWEEN ROUNDS	*8 Sec. 4.4.30

<u>Keyword Number</u>	<u>Keyword</u>	<u>Notes</u>
51	TBR	*8 Sec. 4.4.30
52	NUMBER OF REPLICATIONS	*8 Sec. 4.4.31
52	NUMREP	*8 "
53	NUMBER OF ROUNDS TO BE FIRED	*8 Sec. 4.4.32
53	NUMRND	*8 "
54	NUMBER OF VEHICLES PER TARGET	*8 Sec. 4.4.33
54	NUMVEHTGT	*8 "
55	DISTANCE BETWEEN VEHICLES	*8 Sec. 4.4.34
55	DISTBVEH	*8 "
56	GUN TARGET RANGE	*9 Sec. 4.4.35
56	GTRNG	*9 "
57	REFLECTIVITY	*9 Sec. 4.4.36
57	REFLECT	*9 "
58	DEFLECTION BIAS	*9 Sec. 4.4.37
58	DEFLB	*9 "
59	TARGET HEADING	*9 Sec. 4.4.38
59	TGTHDG	*9 "
60	SEEKER SENSITIVITY	*9 Sec. 4.4.39
60	SEEKSENS	*9 "
61	PROBABILITY OF A CORRECT MESSAGE	*8 Sec. 4.4.40
61	PROBCORMSG	*8 "
62	PROBABILITY OF A WARNING TO LAUNCH	*8 Sec. 4.4.41
62	PROBDOWARN	*8 "

Explanation of notes:

*1 These keywords do not take option words.

*2 These keywords set certain flags or indices that cause the program to function in one of several alternate ways or else serve as indices for array "look-ups." Any increase in the number of options corresponding to one of these keywords might require code changes to COPE (at the least, redimensioning of some arrays) to specify the actions to be taken for the new options.

*3 These keywords take other keywords (numbers 21 through 28) as their second fields (i.e., as the first item following the separator after the keyword). The items to be used in the second fields are built into the program in data statements (CHAR array) and any changes would require code modifications to COPE. (See sections 4.5.1 through 4.5.5)

*4 These keywords set the names of records on TAPE 11 to be read into the program data blocks. Any increase in the number of options allowed must be accompanied by input to PREPMS creating the records with the names that would be called for if the new options were exercised. (See chapter 14.)

Also, keyword 23 takes some further option words which determine digital or voice communications and day or night visibility. If more options are allowed for these keywords, then code changes are required for COPE.

*5 Currently takes no option words, but if option words were added, it would fall in the same class as keywords with note *4.

*6 The addition of any new options for RUMC would require code changes to COPE.

*7 The addition of any new options for SMOKE would require code changes to COPE.

*8 These keywords all take numerical values for option word 1. Any numerical value that could be read in F10.4 format is allowable though for some option words negative values or values greater than 1.0 may cause program malfunction (as in the case when the value represents a probability). Any attempt to modify one of these keywords to accept options other than numbers would require program code changes in COPE.

*9 These keywords affect the choice of the PEDATA record name to be used (see section 5.4). Any increase in the number of allowed choices here would require changing values in the XVALUE array (in both COPE and PAM) and then running PAM to create any new PEDATA records that may be called for as a result of these changes.

In short, one can see from the above notes that while it is easy to add new optionword choices for a particular keyword (by running the PREPMS program when necessary), it often requires code changes to COPE to make the new choices have the desired effect. About the only keywords for which options can be conveniently changed are those of notes 4, 5, and, to a lesser extent, 9.

5.4 PEDATA Record Names.

The record name for the PEDATA (probability of engagement data) to be used with a given case is a function of ten keywords and their associated optionwords (except, of course, when using the OVERRIDE or TEMPORARY keywords with PEDATA).

When the PAM program is run, nine parameters are input. The values of these parameters determine the record name of the PEDATA created by that run. When COPE is run, the same nine parameters are assigned values (as the result of keyword inputs or defaults), the same function for determining record name is applied, and the record with the resulting name is read from TAPE 11 and filled into the PEDATA data block.

Because there are only a finite number of ten character record names and because the record name depends on the values of nine different parameters, it is necessary to limit the choices allowed for each parameter. This section is devoted mainly to a description of the function that determines record name when given the nine parameter values and to the changes required to introduce a new allowed parameter value.

The nine parameters are:

- (1) nominal response time in seconds (determined by keywords 23 and 12),
- (2) designator type (determined by keyword 16),
- (3) target velocity (determined by keyword 42),
- (4) gun-to-target range (determined by keyword 56),
- (5) target reflectivity (determined by keyword 57),
- (6) angle T (i.e., angle between gun-target line and designator target line; determined by keyword 46),
- (7) deflection bias (determined by keyword 58),
- (8) target heading (determined by keyword 59),
- (9) seeker sensitivity (determined by keyword 60).

The values currently allowed are (by parameter number):

- (1) any time t in seconds such that $0.5 \text{ second} \leq t < 999.5$ seconds
- (2) GLLD, MULE, LTD (which COPE and PAM call types 1, 2, and 3 respectively)
- (3) 2, 3, 5, 8, and 9 meters per second,
- (4) 8 and 12 kilometers,
- (5) .05, .10, .20, and .30,
- (6) 0, 25, 30, 60, 90, and 120 degrees,
- (7) -200, -100, 0, 100, and 200 meters,
- (8) -60, -30, 0, 30, and 60 degrees,
- (9) 24, 36, 48, 60, and 72 joules per square meter.

The record name function works as follows:

-First, it rounds the nominal response time (parameter one to the nearest integer (call the result t' , then $1 \leq t' \leq 999$).

-Next, for each parameter 2 through 9, the function matches the parameter value to one of the allowed values. Say for the j th parameter, the value is found to match the I_j th allowed value (if no match is found, an error stop results).

-Next, the record name is made by converting t' to octal (call it t'' where $1 \leq t'' \leq 17478$), by letting $N_j =$ a function of I_j (usually $I_j - 1$), and then defining

record name = 20053333xxxx $N_2N_3N_4N_5N_6N_7N_8N_9$ in octal

where: 20053333 is octal for the alphanumeric "PE00", xxxx are the four octal digits of t'' (possibly with leading zero(s) if t'' is small enough), $N_2, \dots, N_9$ are as defined above (note that when $1 \leq I_j \leq 8$, we have $0 \leq N_j \leq 7$ so that each N_j is but a single octal digit. It is for this reason that we shall never allow more than eight different choices for the value of any of the parameters 2 through 9).

The values allowed for each parameter 2 through 9 are to be found in the XVALUE array which is defined by data statements in both COPE and PAM (also, in PREPMS). In COPE, the data statement is in BDAT1, in PAM it is in subroutine PENAME, and in PREPMS it is in BDATA2.

The XVALUE array is set up so that XVALUE (I,J,K) is the Ith allowed value for the Jth parameter when K=1 and XVALUE (I,J,K) is the N_J value (as defined above a function of I_J with 0<N_J<7) corresponding to the Ith allowed value of the Jth parameter when K=2.

Notes: (1) Because the first parameter (nominal response time) is allowed a wider range of values, the XVALUE entries with J=1 are not used. (2) Because XVALUE is dimensioned XVALUE (8,9,2) and because not all parameters 2 through 9 actually use the maximum of eight different allowable values, those array spaces XVALUE (I,J,1) and XVALUE (I,J,2) for I greater than the number of allowable values for parameter J are filled with 0's and 8's respectively. Hence, if an N_J value of 8 were to occur, the program has made an invalid parameter value match and a stop with error message will occur. (3) The programs do not distinguish between target velocities of 2 and 3 m/s or between 8 and 9 m/s or between designator types MULE and LTD when it comes to PE data. (There are still distinctions made in other parts of COPE such as calculating current designator target ranges or looking up LDWSS probability of hit numbers as function of designator type). This failure to distinguish can be observed in the XVALUE array: the XVALUE (I,J,2) values are identical for target velocities of 2 and 3 m/s, for example.

To allow more choices for one of the parameters 2 through 9, one need merely modify the data statements defining the XVALUE array so that XVALUE (I, J, 1) with appropriate I and J has the new value and XVALUE (I,J,2) has the corresponding N_J (octal digit). This must be done in both COPE and PAM (and should be done in PREPMS to keep it current also). Then one must run PAM with the parameter combinations using the new allowed value(s) to create the PEDATA corresponding to the case(s) one intends to run in COPE with the new parameter values.

To allow a new choice for parameter 1, one needs merely run PAM with the new value for parameter 1 to create the PEDATA for the case(s) using the new nominal response time and the desired allowed values for the other parameters. Of course, nominal response times outside .5 seconds to 999.5 seconds are not allowed.

Note: The PREPMS program uses the XVALUE array only to decipher PE record names. That is, one can direct that program to list the PEDATA records currently on TAPE 11 and then decipher the PE record names and print the values of the 9 parameters to which each such record corresponds. It is for this reason that the XVALUE array data statement in PREPMS should be updated along with those in COPE and PAM whenever a new value is introduced for any of the parameters 2 through 9.

5.5 Control Cards for COPE and Its Preprocessors.

The appendixes A, B, C, and D at the end of this report include respectively, the control cards used to run PAM, PPBLOS, PREPMS, and COPE. They are explained in these appendixes and hence require no further explanation here. These control card sets, of course, apply only to BRL's computer system and would require minor changes for other CDC installations and major changes for non-CDC computers.

CHAPTER 6

6. OUTPUT

This chapter describes the normal COPE output prints. Special messages and diagnostic or error prints that may occur are described in Chapter 7.

The normal COPE output consists of four types of pages, three of which will always be present and one of which will be present only if one or more "\$\$\$" comments is included in the input.

The page types will ordinarily occur in this order:

Title Page
"\$\$\$" comment page (if present)
First case heading page
First case results page
Second case heading page
Second case results page

nth case heading page
nth case results page

final case heading page
final case results page

An additional "\$\$\$" comment page will occur following the results page of the last case whose END card precedes any "\$\$\$" comment card (or cards).

Each page type is now described in detail.

6.1 Title Page.

This page includes the program name, the names of the program author and the various contributors to the model design and data collection. In addition, it identifies the activity, division, branch, and section responsible for the model.

Of greater practical value, this page also includes the version number of the program and the date that version was created. Finally, the page gives the date and time of the current run of the program.

For an example, see Listing 6-1.

6.2 "\$\$\$" Comment Page.

This page merely lists the user's "\$\$\$" comments (see section 4.6.4) with the symbol "** USER **" to the left of each comment line. In addition, any program generated comments (see section 7.1) are listed with "**COPE**" to the left of each comment line.

C O P P E R H E A D O P E R A T I O N A L P E R F O R M A N C E E V A L U A T I O N

-PROGRAM BY RICHARD SANDMEYER
-MODEL DEVELOPMENT AND DATA COLLECTION BY:
DAVID BARNHART JULIAN CHEPNICK DIANA FREDRICK
RICHARD SANDMEYER RICHARD SCUNGIO MICHAEL STARKS
EDWARD STAUCH ANNIE YOUNG
-BASED ON A PROPOSAL BY HERBERT FALLIN
-DONE AT THE REQUEST OF DAVID HARDISON (DUSA-DR)

UNITED STATES ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY
GROUND WARFARE DIVISION
SUPPORT WEAPONS ANALYSIS BRANCH
SYSTEMS EVALUATION SECTION

CODE VERSION 6.2 CURRENT AS OF 6 MARCH 1980
THIS RUN DONE ON 08/14/80 AT TIME 15.41.16.

LISTING 6-1 TITLE PAGE

For an example, see Listing 6-2.

6.3 Case Heading Page.

The first page for each case is the heading page. The heading page gives a description of the conditions in effect for the case whose results follow. In addition, the heading page contains a section for any "\$\$" comments (either user or program generated). The case heading page (Listing 6-3) begins with a line indicating the case number. The next line below that gives the number of replications (i.e., number of potential Copperhead missions generated and tested by the current case of the Monte Carlo COPE model).

The next area of the case heading page is broken up into nine boxes each of which is surrounded by asterisks and contains descriptors of one aspect of the current case. These boxes are now described one at a time.

Note: When an OVERRIDE or TEMPORARY option is in effect, the case heading information may be unreliable for those items connected with the data block for which the OVERRIDE or TEMPORARY is in effect.

In the special case of a PE data block input via the OVERRIDE option, the values given in the comment of section 7.2.6 take precedent over those in the case heading when it comes to determining PE data but not for other purposes (for example, velocity used for calculating designator-to-target range is the velocity printed in the TARGET box rather than that in the comment).

If in doubt, the user should look at the contents (Data Check) of TAPE 8 for the case in question (see section 7.6) to determine which values were actually used for various parameters.

6.3.1 TARGET. The first box is labeled "TARGET" and contains data describing the target. These data are:

- (a) Target type
- (b) number of vehicles in the target unit,
- (c) mean distance between target vehicles,
- (d) target velocity (speed)
- (e) target heading (where a heading of zero means the target is moving along the gun-to-target line toward the gun. Angles are measured counterclockwise from this zero heading).
- (f) target reflectivity.

6.3.2 FIRING INFO. Moving vertically down the page, one next finds the FIRING INFO box which gives various information about the firing

of the COPPERHEAD round. These items are:

- (a) number of rounds fired per engagement,
- (b) round in-flight reliability
- (c) time between successive rounds
- (d) time of flight of round
- (e) nominal gun-to-target range
- (f) angle T (angle between target-to-gun line and target-to-designator line)
- (g) deflection bias (which is the minimum distance between the COPPERHEAD footprint centroid and the target's path).

6.3.3 MISC. The next box below the FIRING INFO is labeled MISC and contains miscellaneous data relevant to the case. These data are:

- (a) either a note that DELTA T3 is zero (in which case RUMC (see section 4.4.18) is optimally chosen for each fire mission or a note that DELTA T3 is not assumed to be zero in which case RUMC is a fixed value for each fire mission.
- (b) The value of RUMC (if DELTA T3 is not assumed zero; otherwise, this line is not printed.)
- (c) The seeker sensitivity value used for the COPPERHEAD seeker in this case.

6.3.4 DETECT, COMMO, and PROC TIMES. The next box which is located at the top of the second column of boxes contains data pertaining to detection, communication, and processing times. These data are:

- (a) The mission type
- (b) The type of communication (i.e., digital or voice)
- (c) day or night indicator
- (d) nominal response time (which is the sum of the median communication and processing times and the time of flight)
- (e) probability of successful digital communication link being established when needed (not printed when item (b) above is VOICE)

- (f) probability of successful voice communication link being established when needed (printed only when applicable - see section 2.2.9)
- (g) probability of a correct message,
- (h) probability that the D.O. is warned to lase.

If the parameterized response time option is used, then the parameterized response time (which does not include time of flight) is listed in this box first.

6.3.5 DESIGNATOR. The next box gives information about the designator (both the soldier and the piece of equipment). This information consists of:

- (a) designator type,
- (b) maximum designator range,
- (c) bail-out range
- (d) designator location.

6.3.6 DESIGNATOR SUPPRESSED. The next box gives information about suppression of the designator. This information is:

- (a) The probability that the FO (D.O.) is killed by Red preparatory artillery fires.
- (b) A label giving the level of direct fire suppression achieved by the target against the D.O.

6.3.7 WEATHER. The first box of the third column gives weather data. These data are:

- (a) The month whose weather is being used in the current case,
- (b) The time of day (on a 0 to 2400 clock) whose weather is being used in the current case.

6.3.8 TERRAIN AND LOS. The next box moving down has terrain and line-of-sight information. This information includes:

- (a) The terrain type used for the acquisition range distribution and line-of-sight segment length distribution.
- (b) The line-of-sight model used in the current case ("SHOOTING GALLERY" or "RANDOM OCCURRENCE").

6.3.9 OBSCURANTS. The final box gives data on obscurants (smoke and dust). These data are:

- (a) Probability that smoke is sufficiently dense to kill a potential COPPERHEAD mission.
- (b) Number of smoke rounds fired of types one and two respectively to obtain the level of smoke in (a) above. (This is printed only if smoke effects are played as a function of number of smoke rounds fired (see section 4.4.19).
- (c) Probability that dust from RED HE artillery fire is sufficient to abort a potential COPPERHEAD mission.

6.3.10 "\$\$" Comment Section.

The final section of the heading page for each case consists of a comment section. This comment section includes both any user comments for this case as well as any program generated comments.

If a case has no comments of either type, then the comment section is omitted.

6.4 Case Results Page.

Following the heading page for a given case is the case results page. This page is labeled in such a fashion as to be almost self-explanatory (see Listing 6-4).

The page has one line corresponding to each mission abort condition and two lines for each round abort condition. A mission abort occurs whenever one of the tests that affect the entire potential COPPERHEAD mission is failed (see sections 2.2.2 through 2.2.11); a round abort occurs whenever one of the tests that affect the functioning of a single round is failed (see sections 2.2.12-2.2.20).

Beginning at the top of the results page, one finds the number of occasions, which is the number of potential COPPERHEAD missions sampled for the current case. This number should equal the number of replications on the heading page for the same case.

The first seven abort causes (or tests) are those which would prevent the D.O. from even attempting to employ COPPERHEAD against a target.

Each of these causes of abort has one line giving the following information:

- (a) The number of the abort cause (or test),
- (b) a label describing the cause of abort,

CAUSE OF ABORT

***** OCCASIONS (10000) *****						
1	DESIGNATOR IN INDIRECT FIRE	84(0.84%)	NUMBER TESTED (% PASSED):	10000(100.16%)		
2	TARGET BEYOND VISIBILITY RANGE	415(4.15%)	NUMBER TESTED (% PASSED):	9916(95.81%)		
3	TARGET BEYOND DESIGNATOR RANGE	0(0.00%)	NUMBER TESTED (% PASSED):	9501(100.0%)		
4	TARGET NOT DETECTED	0(0.00%)	NUMBER TESTED (% PASSED):	9501(100.0%)		
5	SPOKE KILLED MISSION	7123(71.23%)	NUMBER TESTED (% PASSED):	9501(25.03%)		
6	POST KILLED MISSION	887(8.87%)	NUMBER TESTED (% PASSED):	2378(62.70%)		
7	DESIGNATOR FAILED-DUT PPF-COMM	484(4.84%)	NUMBER TESTED (% PASSED):	1491(67.54%)		
***** ATTEMPTED ENGAGEMENTS (1007) *****						
8	COMM NOT (BOTH VOICE & DIGIT)	70(10.70%)	NUMBER TESTED (% PASSED):	1007(93.05%)		
9	FPDP IN TRANSMISSION	21(10.21%)	NUMBER TESTED (% PASSED):	937(97.76%)		
10	DESIGNATOR FAILED-DUT POST-COM	159(11.59%)	NUMBER TESTED (% PASSED):	916(82.64%)		
11	LOS LOST - NO FIRING	55(10.55%)	NUMBER TESTED (% PASSED):	757(92.73%)		
***** SHOTS (702) *****						
POUND NUMBER :		1	2	3	4	5
12	LOS LOST DURING MISSION	30(10.30%)	62(10.62%)	86(10.96%)	151(11.51%)	242(12.42%)
	NUMBER TESTED (% PASSED)	702(95.73%)	702(91.17%)	702(86.32%)	702(78.49%)	702(65.53%)
13	DESIGNATOR NOT WARNED IN TIME	15(10.15%)	0(0.00%)	0(0.00%)	0(0.00%)	0(0.00%)
	NUMBER TESTED (% PASSED)	67(97.77%)	64(100.0%)	60(100.0%)	551(100.0%)	460(100.0%)
14	DESIGNATOR IN DIRECT FIRE	176(11.76%)	173(11.73%)	194(11.94%)	183(11.83%)	164(11.64%)
	NUMBER TESTED (% PASSED)	657(73.21%)	640(72.97%)	606(67.99%)	551(66.79%)	460(64.35%)
15	POUND RELIABILITY FAILURE	20(10.20%)	16(10.16%)	15(10.15%)	16(10.16%)	9(0.09%)
	NUMBER TESTED (% PASSED)	481(95.84%)	467(96.57%)	412(96.36%)	368(95.65%)	296(96.96%)
16	TARGET OBLSCURED BY MINI-TEPPH	54(10.54%)	61(10.61%)	83(10.83%)	55(10.55%)	46(10.46%)
	NUMBER TESTED (% PASSED)	461(98.20%)	451(96.47%)	397(99.09%)	352(84.38%)	287(83.97%)
17	POUND DID NOT ENGAGE TARGET	1(0.01%)	1(0.01%)	4(0.04%)	2(0.02%)	0(0.00%)
	NUMBER TESTED (% PASSED)	407(99.74%)	30(99.74%)	314(99.73%)	297(99.33%)	241(100.0%)
18	POUND ENGAGED BUT DID NOT HIT	105(11.05%)	83(10.83%)	74(10.74%)	54(10.54%)	43(10.43%)
	NUMBER TESTED (% PASSED)	406(74.14%)	389(78.66%)	310(76.13%)	295(81.69%)	241(82.16%)
19	POUND HIT BUT DID NOT KILL	156(11.56%)	159(11.59%)	104(11.04%)	128(11.28%)	96(10.96%)
	NUMBER TESTED (% PASSED)	301(46.17%)	306(46.04%)	236(25.93%)	241(46.80%)	198(51.57%)
***** KILLS *****						
		145	147	137	113	102
PPOR(KILL/DCSN)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/ATMP ENG)=		14.40%	14.60%	13.11%	11.22%	10.13%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77%	84.64%	80.05%	72.70%	60.77%
PPOR(SHT/DCSN)=		7.00%	6.97%	6.97%	10.07%	10.07%
PPOR(SHT/ATMP ENG)=		1.45%	1.47%	1.37%	1.13%	1.02%
PPOR(KILL/HIT)=		20.66%	20.94%	18.80%	16.10%	14.53%
PPOR(LDS)=		88.77				

- (c) the number of times the test was failed,
- (d) the percentage of the total occasions that failed the test
- (e) The number of potential COPPERHEAD fire missions on which the test was performed, (this number is equal to the total number of potential COPPERHEAD fire missions minus the number of potential missions that failed prior tests).
- (f) The percentage of those potential COPPERHEAD fire missions on which the test was performed that passed the test.

Following the seventh abort cause is a line giving the number of attempted engagements. This is the number of replications (out of the total original number of potential COPPERHEAD missions) that passed the first seven tests and have, therefore, reached the point where the D.O. attempts to call for COPPERHEAD fire.

The next four cause-of-abort lines (abort causes numbers 8 through 11) consist of those tests that must be passed before the battery will fire a COPPERHEAD mission. Each of these abort causes has one line giving information as described in (a) through (f) above in this section.

Following the eleventh cause-of-abort line is a line giving number of shots. This is the number of replications for which the mission reached the point where the battery actually fired COPPERHEAD (i.e., the mission passed tests 1 through 11). The number of shots is the number of missions fired, not the number of individual rounds fired.

The final eight abort causes (numbers 12 through 19) are round aborts which have two lines each in the printout.

The first line of each round abort cause gives:

- (a) The number of the abort cause (test),
- (b) a label describing the cause of abort,
- (c) the number of times the test was failed for the first round,
- (d) the percentage of the total occasions for which the first round failed the test.

The line continues with items (c) and (d) repeated but for the second and subsequent rounds up to a total of NRF rounds (NRF_≤6).

The second line of each round abort gives

- (e) the number of COPPERHEAD mission first rounds on which the test was performed,
- (f) the percentage of those COPPERHEAD fire mission first rounds on which the test was performed that passed the test.

The second line continues with items (e) and (f) repeated but for the second and subsequent rounds of the COPPERHEAD fire mission.

Following the final (19th) abort cause line is a line giving kills. This line gives the number of replications for which the first round of the COPPERHEAD mission achieved a kill, the number of replications for which the second round of the COPPERHEAD mission achieved a kill, and so on through the final round of the COPPERHEAD mission.

Next, the printout gives a number of probabilities derived from the case just simulated: (these probabilities are printed as percentages).

The probability of kill given an occasion for the nth round of the fire mission is printed and is calculated by dividing the number of kills for the nth round by the number of occasions.

The probability of kill given an attempted engagement for the nth round equals the number of kills for the nth round divided by the number of attempted engagements.

The probability of kill given a shot for the nth round equals the number of kills for the nth round divided by the number of shots.

The probability of line-of-sight (LOS) for the nth round is the probability of having LOS when the round arrives (test 12) given that LOS existed immediately prior to firing (test 11). (Note: The preceding applies to the "shooting gallery" LOS model; for random occurrence, the probability of having line-of-sight for the nth round is just equal to the success rate of test 12).

The probability of shot given an occasion is the fraction of potential COPPERHEAD fire missions that actually reach the point where rounds are fired.

The probability of shot given an attempted engagement is the number of shots divided by the number of attempted engagements.

The probability of attempted engagements given an occasion is the number of attempted engagements divided by the number of occasions.

The final line of the results page gives information on designator survivability. This line gives the number of replications in which the designator was killed by direct fire from the target and the percentage of the total replications that number represents. The line also gives the number of replications for which the test of designator kill was performed and the percentage of those replications in which the designator was not killed.

Note: The number of kills shown on the results page differs slightly in its meaning depending on whether "shooting gallery" or "random occurrence" line-of-sight model is used. When "shooting gallery" is used, no attempt is made to account for multiple kills of the same target vehicle (i.e., the number of kills on the nth round is calculated under the assumption that the vehicle the round hit had not already been killed by a previous round).

When "random occurrence" is used, the number of kills on the nth round takes into account the number of vehicles surviving to the nth round.

This difference, of course, has no effect in comparing first round kills computed using the two differing LOS models. Furthermore, for round n there will be no effect from this difference in bookkeeping if (1) $n < \text{number of vehicles in the target}$, (2) $\text{distance between target vehicles} / \text{target velocity} < \text{time between rounds}$, and (3) the D.O. is assumed always to designate the front most target vehicle in view. In other cases, the results obtained by the two methods will have some differences attributable to the difference in bookkeeping between the two methods.

CHAPTER 7

7. PROGRAM GENERATED MESSAGES AND DEBUGGING PRINTS

The COPE program has been designed so that most of the common errors will cause the program to come to a stop and print an error message. Of course, it is not possible to anticipate every possible error and the user can expect occasionally to encounter a run that terminates with a dump or a system generated error message. However, the current version of COPE has been used for hundreds of runs now and has not terminated in anything other than the normal program termination or one of the anticipated error stops.

The messages generated by COPE can be divided into five classes:

- (1) program generated \$\$\$ comments,
- (2) program generated \$\$ comments,
- (3) special program print-outs of a warning nature that do not result in program termination
- (4) special program prints that occur with abnormal program terminations and which give information useful in locating the cause of termination.
- (5) STOP messages that are printed in the day file. (These STOPS include the subroutine or function name in which the stop occurs.)

The messages are now explained. They are ordered alphabetically (by first line) within each class. (Note that some messages include more than one line).

7.1 Program Generated \$\$\$ Comments.

YOU HAVE SEQUENCE NUMBERS IN COLUMNS 73-78 OF TAPE 5. THEY
WILL BE REMOVED BY THE PROGRAM;
HOWEVER, IF THE TEMPORARY OPTION WAS USED TO READ IN ANY
DATA, THE RESULTS MAY BE SUSPECT.

This message is generated in subroutine SEPREC and is part of a modification added to handle a Control Data peculiar problem. Users of COPE frequently create TAPE 5 (the file of case descriptors) using the CDC editor in its default FORTRAN format and then save the resulting file by the command SAVE, LFN (where LFN is the local file name). The problem is that the editor then inserts sequence numbers in columns 73-78 of the saved file. Now one could easily avoid this problem by using SAVE, LFN, N rather than SAVE, LFN (the additional N

causing no sequence numbers to be saved). However, to avoid aborting runs in which sequence numbers are present, COPE was modified to check the first non-comment card (or card image) of file TAPE 5 for sequence numbers: if columns 73-78 are filled with numbers and columns 63-72 and 79-80 are blank, then the computer concludes that sequence numbers are present, issues the above comment, and ignores columns 73-78 on all cards of the current run.

As mentioned in the message itself, this ignoring of columns 73-78 could adversely affect the results if the TEMPORARY option is used, since that option quite likely has important information in columns 73-78. In this event, the file TAPE 5 should be created in the editor with 80 character lines (i.e., use CDC command F,CH=80 before creating the file). Then when it is saved, the sequence numbers will be in columns not read by the program.

For keywords other than TEMPORARY, there is not likely to be any adverse effect since the case descriptor character strings on the cards will usually terminate before column 63 and hence the deletion of columns 73-78 will have no effect.

Finally, if the computer should erroneously conclude that sequence numbers are not present when they really are (as could happen if any of columns 63-72 or 79-80 of the first non-comment card were not blank), then the computer would treat the sequence numbers as part of the keyword or optionword and a different error would bring the program to a halt when that keyword or optionword was not found (see section 7.4.5).

One may wish to remove this feature if using a non-CDC computer.

7.2 Program Generated \$\$ Comments.

7.2.1 CHOICE OF option word FOR RESPONSE TIME DATA IS INCONSISTENT WITH TIME OF option word FOR WEATHER DATA.

The first option word is DAY or NIGHT while the second one is 0600, 1400, or 2200.

This message merely warns the user that one of the following pairs of a RESPONSE TIME data option word and a WEATHER data option word is being used:

RESPONSE TIME option word	WEATHER option word
DAY	2200
NIGHT	0600
NIGHT	1400

AD-A100 285

ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY ABERDEEN PROV--ETC F/G 19/1
COPPERHEAD OPERATIONAL PERFORMANCE EVALUATION (COPE): COMPUTER --ETC(U)
MAR 81 R S SANDMEYER

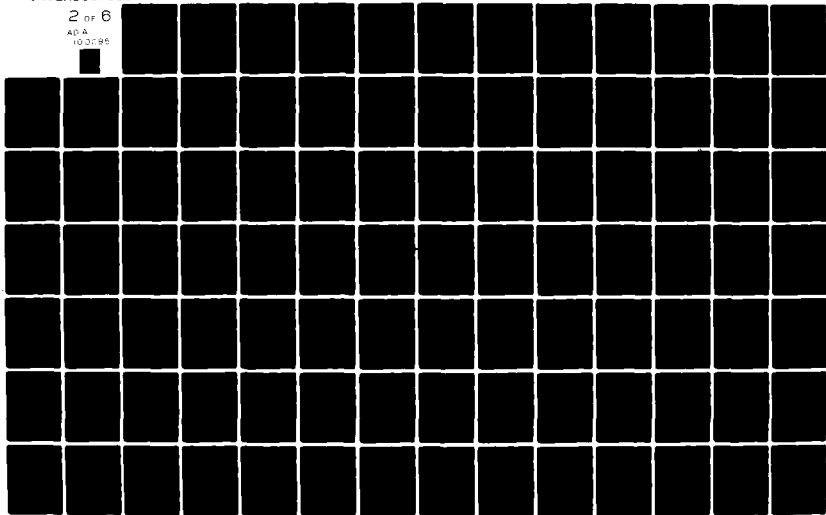
UNCLASSIFIED

AMSAA-TR-318

NL

2 of 6

AD-A
10 01 85



These are all considered inconsistent pairs since daylight is usually gone by 2200. On the other hand, day light is usually present at 0600 and 1400.

The choice of one of these inconsistent pairs will not abort the program, but merely result in the generating of this message and the executing of the case with the inconsistent data selected.

7.2.2 IN CCLOT FOR CASE nn WITH ICCPFG= 1. NO PLOT PRODUCED.

nn is the case number.

This message merely means that the user wanted a Cal Comp plot for case number nn, but since the plot routine is not yet implemented, no plot was produced.

7.2.3 IN PLOT FOR CASE nn WITH IPPFG = 1. NO PLOT PRODUCED.

nn is the case number.

This message merely means that the user wanted a printer plot for case number nn, but since the plot routine is not yet implemented, no plot was produced.

7.2.4 NO ENDF CARD FOUND BUT ONE ASSUMED SINCE END OF INPUT FILE ENCOUNTERED.

This message occurs when no ENDF card has been put at the end of TAPE 5 (INPUT). It merely means that the program reached the end-of-file without encountering an ENDF card and then proceeded as though it had found one there.

7.2.5 NO 'FIRST CASE' OR 'NEXT CASE' CARD FOUND --- NEXT CASE, RD OPTION ASSUMED.

This message means that inputs for a new case (possibly the first case) have been encountered, but were not preceded by a 'FIRST CASE' or 'NEXT CASE' card. In this event, the program proceeds as though a NEXT CASE, RD card had been encountered prior to the new case's inputs.

7.2.6 PEDATA FOR: TR=xxx. DT= n VEL= yy. GTR= zzz. REFL= a.aa
ANGLET= bbb. DEFT= ccc. TGTHD= dd. SKSEN= ee.

xxx., n, yy., zzz., a.aa, bbb., ccc., dd., and ee. are respectively the response time, designator type number, target velocity, gun-to-target range, reflectivity of target, angle T, deflection bias, target heading, and seeker sensitivity used to generate the PEDATA used in this run. This message occurs only when the OVERRIDE option is used with PEDATA.

The units used with each quantity are those normally used on the corresponding keyword cards.

7.2.7 RECORD NAMED: Record Name USED TO OVERRIDE FOR Data Block Name.

Here record name is the name of the record on the word addressible mass storage file that is selected by the OVERRIDE option and data block name is the name of the data block to be filled by this override.

This print occurs only with each data block in a given case for which the OVERRIDE is in effect.

7.2.8 TEMPORARY OPTION IN EFFECT FOR Data Block Name.

Here data block name is the name of the data block filled by use of the TEMPORARY OPTION.

This print occurs only with each data block in a given case for which the TEMPORARY option is in effect.

7.2.9 YOUR INPUT LINE: Input Line Characters SUPERSEDES ONE OF YOUR PREVIOUS INPUT LINES FOR THIS CASE.

The input line characters are the contents of the card that supersedes a previous line. This message is only a warning to the user that he has included two (or more) cards with the same keyword in a single case. When this happens, only the last such card has an effect on the choice of case descriptors.

If two PLOT cards (section 4.4.5) are included in a case (one for CALCOMP and one for PRINTER), then this message will be produced. In such a case, both plot types are set according to the input cards even though the message says the last one superseded the earlier one; this is the one case in which this message can be safely ignored.

In most cases, one will want to check the input cards and probably re-run the case so that no more than one card is used for each keyword. Note that an OVERRIDE or TEMPORARY card applied to a certain keyword counts as an input card with that keyword for purposes of this message (for example, putting an OVERRIDE, W, JUN1400021 card and a W, J, 0600 card in the same case will produce the message of this section).

7.3 Special Program Print-Outs of a Warning Nature That Do Not Result in Program Termination.

7.3.1 IN OUTPUT. PROBLEM 1 J= nnnnn RNDRED=xxxx. NKILL = mmmmm.

nnnnn is the number of the round, xxxx. is the number of kills on the Jth round for this case calculated by adding one each time a kill occurs, and mmmmm is the number of kills on the Jth round for this case calculated by subtracting the number of aborted missions and Jth rounds from the number of replications.

This print occurs only if the Jth round kills (J runs from 1 to NRF) computed by the two methods differ. It indicates a serious problem with the program.

7.3.2 SERIOUS PROBLEM! KTEST AND NTEST ARRAYS DO NOT AGREE
II= nnnnn JJ = mmmmm KTEST ARRAY: (Followed By A
20 x 6 Matrix).

nnnnn is the abort number and mmmmm is the round number (or if nnnnn corresponds to a mission abort, mmmmm is 1).

II and JJ are the indices of the first entry in the KTEST and NTEST arrays that do not agree.

The 20 x 6 matrix is the KTEST array where the element in row I column J is KTEST (I, J). The NTEST array elements correspond to the 'number tested' values printed on the case results page where NTEST (I, J) corresponds to the Jth round Ith abort condition (when the abort is a mission abort, J=1).

The KTEST array is filled by incrementing KTEST (I,J) each time the Ith test is made for the Jth round (if the test is for a mission abort J=1). The NTEST array is filled by setting NTEST (I,J) equal to the number of replications minus the number of missions (and Jth rounds) aborted by tests 1 through I-1.

If NTEST and KTEST differ, then the program is not behaving correctly.

7.4 Special Program Prints That Occur with Abnormal Program
Termination and Which Give Information Useful in Locating
the Cause of Termination.

7.4.1 ERROR IN INPUT. LINE: Character String WITH ITI=nnnnn
IS IN ERROR.

The character string is the card image read from TAPE 5 (INPUT) that is in error and nnnnn is the keyword number.

The error occurs when a keyword match is found for the input, but no code has been provided to handle the input. This is a serious error and means that a program change is required.

This message corresponds to STOP in section 7.5.15.

7.4.2 IN COMMNT. TOO MANY \$\$ COMMENTS (>20): (Followed By 20
Numbered Lines of Comments, Then:) THE FOLLOWING COMMENT
CAUSED OVERFLOW OF CMNT ARRAY: (Followed by a 21st
Comment Line).

The CMNT array is dimensioned to allow up to 20 \$\$ comments per case. This message means that 21 \$\$ comments were generated for the current case and consequently the CMNT array overflowed. The \$\$ comment count includes both program generated and user comments.

The corrective action is either to reduce the number of comments for the case or increase the second dimension of the CMENT array.

Note: One possible cause of too many comments is the omission of one or more END cards.

This message corresponds to the STOP of section 7.5.1.

7.4.3 INPUT ATTEMPTED TO READ RECORD NAMED Record Name
BUT NO SUCH RECORD WAS FOUND.

The record name is the name of the record which the program attempted to read from the word addressible mass storage file but which could not be found on that file.

This message means the desired record does not exist on the mass storage file. In order to create it, one of the pre-processor programs must be run. If the OVERRIDE feature was used, try preceding the record name with a \$ (dollar sign), as described in section 4.5.3.

This message is accompanied by a trace back print of the subroutines called to produce the message. This can be used to locate which call to READMS actually caused the abnormal termination.

This message occurs with the STOP in section 7.5.21.

7.4.4 IN FINDIT. KEYWORD: Keyword AND CORRESPONDING VALUE
OF IT: Value ARE NOT ALLOWED WITH THE Option Word OPTION.

Here keyword is the keyword character string, value is the keyword number, and option word is an option word character string.

This message occurs when one attempts to use OVERRIDE, TEMPORARY, or RESET features with an option word that is a keyword other than the name of one of the data blocks.

This message corresponds to the STOP of section 7.5.5.

7.4.5 IN FINDIT. NO MATCH FOUND FOR LINE IB= nnn OF THIS B
ARRAY: (Followed By Up to 10 Lines of Character Strings).

Here nnn is the value of IB.

IB is the number of the character string for which no match was found. If IB=1, there was no keyword match (this corresponds to the stop of section 7.5.4). If IB=2, then one could be in the situation of the STOP of section 7.5.4, section 7.5.6 or section 7.5.7.

This error usually means the keyword or option word found in line IB was misspelled or does not exist.

7.4.6 IN NUMRIC. LINE nnn EXCEEDS ALLOWABLE FIELD LENGTH
FOR NUMERIC DATA: (Followed By Up to 10 Lines of
Character Strings).

Here nnn is the number of the line in the listing of character strings which contains a numerical string of more than 10 characters (including "-" sign and "." if present).

The corrective action is to change the number to one of less than 10 characters (when including "-" and "."). With the parameters used in COPE there should never be occasion to use a number requiring more than 10 characters.

(Note: If the number is written on input without a decimal point, then the program adds a decimal point after the last digit. Hence, if the decimal point is omitted the number should not exceed nine characters).

This error corresponds to the STOP of section 7.5.23.

7.4.7 IN NUMRIC. LINE nnn HAD TOO MANY DECIMAL POINTS:
(Followed By Up To 10 Lines of Character Strings).

Here nnn is the number of the line in the listing of character strings which contains more than one ".".

The corrective action is to rewrite the input line so that no option word has more than one decimal point.

This error corresponds to the STOP of section 7.5.22.

7.4.8 IN SMPLCD WITH RANDOM NUMBER GREATER THAN 1.0 RN=xx.xxxxxx.

Here xx.xxxxxx is the random number generated for use in SMPLCD.

This error will occur only if the random number generator malfunctions and produces a number 1.0. If this error occurs, a trace of subroutine calls is produced and the STOP of section 7.5.31 occurs.

7.5 STOP Messages That are Printed in the Dayfile.

7.5.1 STOP IN COMMNT: TOO MANY \$\$ COMMENTS. This message occurs when too many \$\$ comments are encountered in one case. See section 7.4.2.

7.5.2 STOP IN COPE: NORMAL PROGRAM TERMINATION. This is the desirable stop message. It means that the program terminated after processing all the inputs and that no abnormal stopping conditions were encountered. Of course, it does not guarantee that the program did what the user expected, and there may be some warning diagnostic messages (of types 1, 2, or 3 above) produced.

7.5.3 STOP IN DFCHK: ERROR NUMBER 1. This stop occurs when the designator-to-target range cannot be bracketed by the range values in the DFFOKL array. This indicates either an error in the range calculation or a need for larger range values in the DFFOKL array.

7.5.4 STOP IN FINDIT: ERROR NUMBER 1. This message means that a search of the CHAR array has found no keyword to match that of the current input card. (If the VERRIDE, TEMPORARY, or RESET option is used, then this message will also occur if the second character string on the card cannot be matched from the CHAR array).

This stop corresponds to the print-out of section 7.4.5.

7.5.5 STOP IN FINDIT: ERROR NUMBER 2. This message occurs when one attempts to use the VERRIDE, TEMPORARY, or RESET option with anything other than one of the eight data block names. (See section 4.5.)

See section 7.4.4.

7.5.6 STOP IN FINDIT: ERROR NUMBER 3. This message occurs when no match for a given option word from an input card can be found among the allowable option words for the input card's keyword.

See section 7.4.5.

7.5.7 STOP IN FINDIT: ERROR NUMBER 4. This message means the same as that in 7.5.6, but occurs at a different point in subroutine FINDIT.

7.5.8 STOP IN FPRCNT: ERROR NUMBER 1. This message means that function FPRCNT was called to format into a percentage a number greater than or equal to 100.05. There should never be any percentages greater than 100 in this program, so if this message is obtained, there are major problems in the bookkeeping processes. In particular, the NABORT array and NTEST array are probably filled with wrong numbers.

7.5.9 STOP IN GETRNG: TARGET VELOCITY = 0.0. This message means a target velocity of zero was input. This is not allowed since division by zero in calculating line-of-sight duration would cause division overflow on the computer.

If one really wishes to use zero velocity, it is suggested that one use a velocity of, say, .001 which will create line-of-sight durations so great that the end result will not differ significantly from those of a stationary target.

A subroutine call trace is also produced.

7.5.10 STOP IN GETVIS: RANDOM NUMBER TROUBLE 1. This message means that the attempt to sample weather data failed under cloud free line-of-sight conditions.

This probably means that there is an error in the PRGCFL array or the random number generator is malfunctioning.

A subroutine call trace is also produced.

7.5.11 STOP IN GETVIS: RANDOM NUMBER TROUBLE 2. This message means that the attempt to sample weather data failed under no cloud free line-of-sight conditions.

This probably implies that there are invalid numbers in the W array or the random number generator is malfunctioning.

A subroutine call trace is also produced.

7.5.12 STOP IN HEADER: ERROR NUMBER 1. This error occurs only if IOVER (J) has a value other than 0, 1, or 2 for a value of J = 21, 22, 23, 24, 25, 26, 27, or 28.

This should never happen. If it does there is a serious problem somewhere in the input or the random access mass storage file.

7.5.13 STOP IN HITCHK: FAILED TO BRACKET RANGE. This stop occurs when the designator-to-target range cannot be bracketed by the range values in the RNGPST array. This indicates either an error in the range calculation or a need for longer range values in the RNGTHF array.

7.5.14 STOP IN INPUT: EOF ENCOUNTERED. This stop occurs when an end-of-file is encountered on TAPE 5 after an END card (instead of after the expected ENDF card).

7.5.15 STOP IN INPUT: ERROR NUMBER 1. This stop corresponds to the message of section 7.4.1.

7.5.16 STOP IN INPUT: EXTRA NEXT CASE CARD. This stop occurs when two (or more) NEXT CASE cards occur within the same case (i.e., without an intervening END card). The program will not accept more than one NEXT CASE card per case. (All cards lying between consecutive END cards are considered to form a case. For this purpose, the beginning and end of TAPE 5 may both be regarded as END cards.)

7.5.17 STOP IN INPUT: FIRST CASE CARD IN SUBSEQUENT CASE. This stop occurs if a FIRST CASE card is encountered anywhere on TAPE 5 other than in the first case. Cases other than the first case should be headed by NEXT CASE cards rather than FIRST CASE cards.

7.5.18 STOP IN INPUT: PARAMETERIZED RESPONSE TIME ≤ 0.0 . This stop occurs when attempting to play a parameterized response time less than or equal to zero. Such times are not allowed in the program.

7.5.19 STOP IN LOSCHK: ERROR WITH RANDOM OCCURRENCE OPTION. This message means that the designator-to-target range cannot be bracketed by the range value in the RNGPLS array. This indicates either an error in the range calculation or a need for larger range values in the RNGPLS array.

7.5.20 STOP IN MINTRN: FAILED TO BRACKET RANGE. This stop occurs when the designator-to-target range cannot be bracketed by the range values in the RNPST array. This indicates either an error in the range calculation or a need for larger range values in the RNPST array.

7.5.21 STOP IN NOREC: ATTEMPTED TO READ NON EXISTENT RECORD. This stop corresponds to the message of section 7.4.3.

A subroutine call trace is also produced.

7.5.22 STOP IN NUMRIC: ERROR NUMBER 1. This stop corresponds to the error message of section 7.4.7.

7.5.23 STOP IN NUMRIC: ERROR NUMBER 2. This stop corresponds to the error message of section 7.4.6.

7.5.24 STOP IN PECHK: FAILED TO BRACKET DELTA T. This message means that the time delay cannot be bracketed by the DLTT array time values. This indicates either an error in calculating DELTAT or a need for larger time values in the DLTT array.

7.5.25 STOP IN PENAME: ERROR NUMBER 1. This stop means that one of the input values that determines which PEDATA record is to be used had a value not found in the XVALUE array. (See section 5.4.)

7.5.26 STOP IN PENAME: ERROR NUMBER 2. This stop means that an illegal index value was assigned to NP in subroutine PENAME. (See section 5.4.)

7.5.27 STOP IN PENAME: ERROR NUMBER 3. This stop means that a response time less than zero or greater than or equal to 999.5 was input. This is outside the allowable range of response times.

7.5.28 STOP IN PENAME: (PEIDNT) ERROR NUMBER 4. This stop means that the PEDATA record name could not be decoded successfully. (See section 5.4.)

7.5.29 STOP IN SEPREC: ERROR NUMBER 1. This error occurs only if the number of keywords and/or option words on an input card is greater than 10 or less than or equal to zero.

7.5.30 STOP IN SMPLCD: FAILED TO BRACKET RANDOM NUMBER. This stop occurs when the random number obtained from the uniform random number (0 to 1) generator cannot be bracketed by the cumulative probability values describing the distribution to be sampled. It indicates either a malfunctioning random number generator or an incomplete description of the distribution to be sampled.

A subroutine call trace is also produced.

7.5.31 STOP IN SMPLCD: RANDOM NUMBER TOO LARGE. This stop corresponds to the error message of section 7.4.8.

A subroutine call trace is also produced.

7.6 Echo Printing.

The COPE program uses TAPE 8 (i.e., logical output unit 8) to echo inputs, to print the values of the case parameters, and to print the initial random number seeds. TAPE 8 is normally a scratch file that is lost when the job is completed; however, if one wishes to see the contents of TAPE 8 for debugging or verification purposes, then one need merely list it or catalog it for future reference.

A sample of TAPE 8 contents is shown in Listings 7-1 and 7-2.

For each case that is run there are two sections of output on TAPE 8. The first section is a listing of the input options (i.e., the case descriptor cards on TAPE 5) that apply to the case.

The second section (Data Check) displays the values of most of the data that describe the case. This section includes data from the word addressable mass storage file (or its equivalent obtained by using the TEMPORARY option), data obtained directly from INPUT cards, and a few computed values. Together, these data give the complete numerical description of the case to be simulated. (This echo print data check has two forms, a long form and a short form. The long form gives all of the data describing the case; the short form gives it all except the PEDATA (i.e., the INDEX and PETBL arrays). (See section 4.4.6.)

Listing 7-1 gives an example of what is printed on TAPE 8 when the short form of the echo print is used. Listing 7-2 shows the TAPE 8 printout when the echo print long form is used.

The values listed in the Data Check section of the TAPE 8 printout are not identified by any printout labeling. Instead, the user must locate the value of a particular variable on the printout by matching the printout lines with the WRITE statements in subroutine ECHO. If the user intends to use this Data Check section frequently, then perhaps some labeling should be added.

In addition to the two sections of output for each case, TAPE 8 also includes the random number seeds. These are printed immediately following the first case data check. Since the random number seeds used for each case are the same, the print of random number seeds is not repeated following the data check prints for subsequent cases.

Note that the column numbers at the top and bottom of each page of Listings 7-1 and 7-2 are not part of the output produced on TAPE 8. They are included here only for the reader's convenience.

[illegible][illegible]

7-17

[illegible]

LISTING 7-2 LONG FORM OF ECHO PRINTS - CONT'D

(PAGE 2 OF 17)

[illegible]

PE APPAY FQ? 1- 37

PE APRAY FND 1 38

PE AFPAV FC3 I= 39

PE APPAY FB3 I= 40
,0300 .0100[illegible]

LISTING 7-2 LONG FORM OF ECHO PRINTS - CONT'D

(PAGE 11 OF 17)

	COLUMN NOS.

[illegible][illegible]

CHAPTER 8

8. PROGRAM DESIGN

8.1 Programming Principles of COPE.

The main COPE program has been designed according to the following principles:

- (1) Input is designed to allow multiple cases per run with only a few mnemonically keyed input lines per case. For flexibility, it is also possible to read in data as traditional FORTRAN formatted input.
- (2) Output is designed so that the printout of results is nearly self-explanatory. In addition, user generated comments can easily be made to appear in the printout for any case.
- (3) The program is highly modularized. The main program consists almost entirely of calls to various subroutines together with the logic necessary to initialize and control the execution of three nested loops (see section 8.2).
- (4) Most subroutines or functions are designed for a single purpose (e.g., perform a test, read input, write output, etc.) so that when one wishes to change any aspect of the program it is usually necessary to change only a small number of the subroutines (usually the one performing a particular test plus the input routine).
- (5) The principles of structured programming have been adhered to in some degree. Because the program is written in FORTRAN IV (extended CDC version) which does not have if-then-else, do-until, do-while constructs, it is still necessary to use GO TO's and statement labels. However, blank comment cards have been inserted between logical blocks within each program to improve readability, and unnecessary GO TO's have been minimized. In addition, statement labels are in ascending numerical order within each subroutine.
- (6) All data statements have been grouped into a single BLOCK DATA subprogram.
- (7) Mnemonically meaningful variable and subroutine names have been selected.
- (8) FORTRAN code has been written to minimize the number of non-standard (i.e., CDC peculiar) features used. Nevertheless, there are some CDC peculiar code features that were impractical to omit (e.g., READMS, SYSTEMC, etc.). These are cited in Chapter 13.

8.2 General Logic Flow.

The COPE main program can be viewed as three large nested loops on case, replication, and round. Figure 8-1 illustrates the general COPE logic flow including the three loops.

This figure is necessarily oversimplified and a more detailed flow chart is to be found in Chapter 11, but the overall picture is accurately summed up here.

8.3 Subroutine and Function Calling.

Table 8-2 gives a listing of the names of the main program, all of the subroutines, and all of the functions in the COPE program.

COPE is the main program name.

FPRCNT, GAMMA, IDCHAR, NUMRIC, and URAN31 are the function names.

ABRTTL, ADDTBR, ADDTOF, ARTCHK, BLOCHK, CCPLT, COMMNT, CREAD, DETCTN, DEFAULT, DFCHK, DUST, ECHO, FINDIT, GETRNG, GETTIM, GETVIS, HEADER, HITCHK, INITLZ, INPUT, LOSCHK, MINTRN, NOREC, OUTPUT, PECHK, PENAME, PKCHK, PLOT, PSMOKE, REINTZ, RNDREL, RNGCHK, SEPREC, SMOKE, SMPLCD, TIMCHK, TITLE, USET, VISCHK, and WARNFO are the subroutine names.

PEIDNT is an entry point name in subroutine PENAME.

For each list entry (subprogram name) in Table 8-2, there are sections giving the names of subprograms calling the given list entry, the names of subprograms (in COPE) that the given list entry calls, and the names of system or IMSL routines that the given list entry calls.

There is a discrepancy that occurs in the list. Subroutine NOREC is called by system routines LOCF (as an external argument) and SYSTEMC (by address loaded in IRAY array) and hence does not show as "CALLED BY" any routine in Figure 8-2 (see section 9.31).

Table 8-3 lists the system (CDC, standard FORTRAN, and International Mathematical and Statistical Library) routines used in COPE and the subprograms of COPE that call each of them.

Table 8-4 is a subprogram calling tree. Each subprogram (subroutine or function) in column two can be called by the COPE main program directly. Each subprogram in column N calls those subprograms in column N+1 to which it is linked by lines (V's, I's, -'s, and >'s).

NOREC calls no other subprograms in COPE. It is called by SYSTEMC whenever CDC FORTRAN IV Extended execution error 104 is encountered in INPUT, INITLZ, or FINDIT.

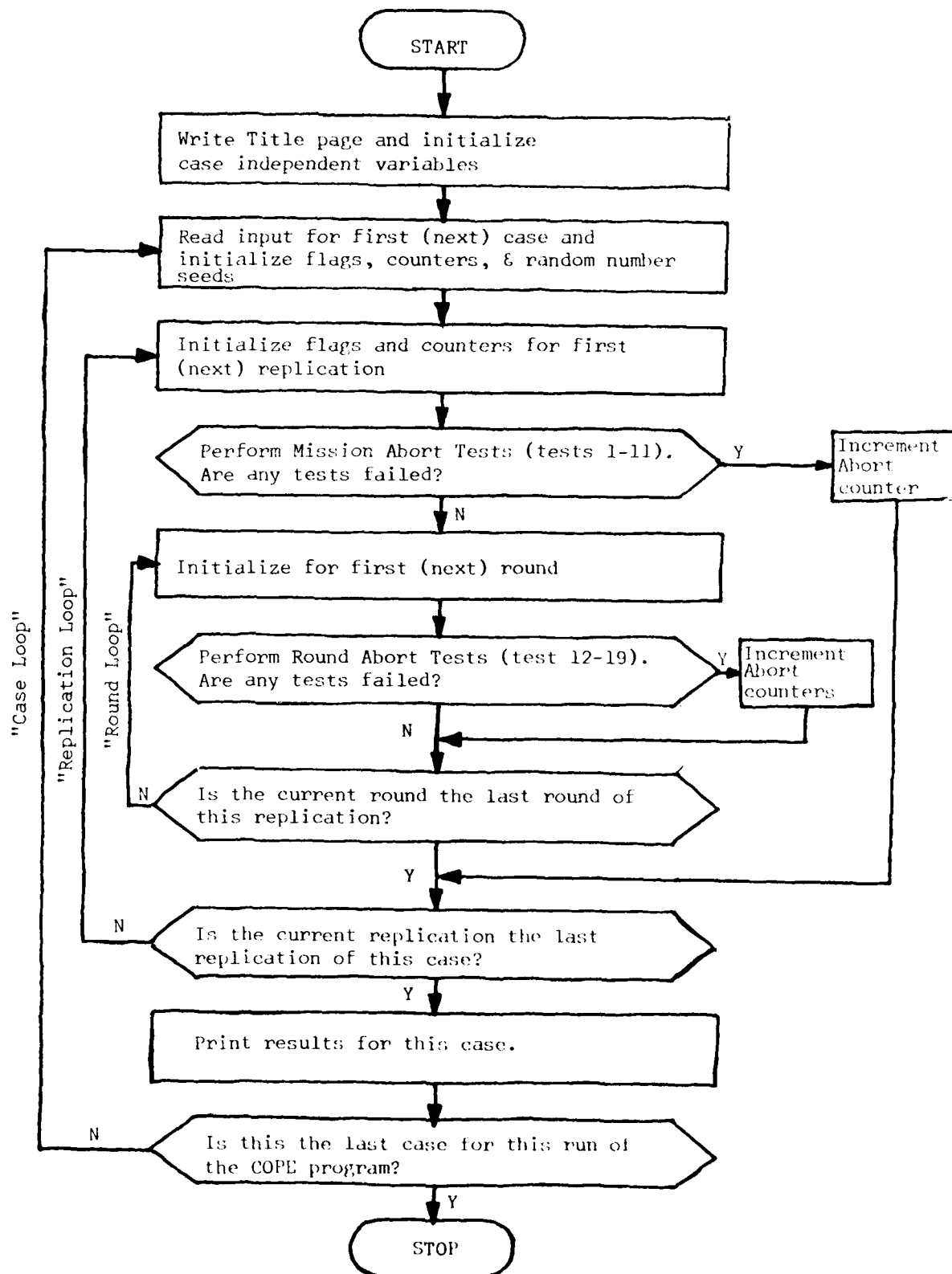


FIGURE 8-1 General COPE Logic Flow

TABLE 8-2 PROGRAM UNIT REFERENCES (SUBROUTINES, FUNCTIONS, AND MAIN PROGRAM)

ABRTTL	CALLED BY	ARTCHK HITCHK RNGCHK	BLOCHK LOSCCHK SMOKE	DETCIN MINTRN TIMCHK	DECHK PECHK VISCHK	DUST PKCHK WARNDG	GETTIM RNDREL
ADDTRR	CALLED BY	COPE					
ADDTRF	CALLED BY	COPE					
ARTCHK	CALLED BY CALLS	COPE ABRTTL	URAN31				
BLOCHK	CALLED BY CALLS	COPE ABRTTL					
CCPLOT	CALLED BY CALLS CALLS (SYS)	INPUT COMMNT ENCODE					
COMMNT	CALLED BY CALLS (SYS)	CCPLOT ENCODE	HEADER WRITE	INPUT	PLOT	SEPRED	
COPE	CALLS	ADDTRR DUST INPUT REINTZ VISCHK	ADDTRF GETRNG LOSCCHK RNDREL WARNDG	ARTCHK GETTIM MINTRN RNGCHK	BLOCHK GETVIS OUTENT SMOKE	DETCIN HITCHK PECHK TIMCHK	DECHK INITI2 PKCHK TITLE
CREAD	CALLED BY CALLS CALLS (SYS)	INPUT SEPRED DECODE	ENCODE	READ			
DETCIN	CALLED BY CALLS	COPE ABRTTL	SMPLOC				
DEFAULT	CALLED BY	INITLZ					
DECHK	CALLED BY CALLS CALLS (SYS)	COPE ABRTTL AMAX1	URAN31				
DUST	CALLED BY CALLS	COPE ABRTTL	URAN31				
ECHO	CALLED BY CALLS (SYS)	INPUT WRITE					
FINDIT	CALLED BY CALLS CALLS (SYS)	INPUT NUMRIC IFIX	LOOP	READMS	SYSTEMC	WRITE	
FPRONT	CALLED BY CALLS (SYS)	OUTPUT ENCODE					
GAMMA	CALLED BY CALLS (JMSL)	GETTIM GGAMA					

TABLE 8-2 PROGRAM UNIT REFERENCES (SUBROUTINES, FUNCTIONS, AND MAIN PROGRAM) - CONT'D

GETRNG	CALLED BY CALLS CALLS (SYS)	COPE SMPLCD FLOAT	STRACE					
GETTIM	CALLED BY CALLS	COPE ABRTTL	GAMMA	SMPLCD	URAN31			
GETVIS	CALLED BY CALLS CALLS (SYS)	COPE URAN31 STRACE						
HEADER	CALLED BY CALLS CALLS (SYS)	INPUT COMMNT DECODE	PEIDNT ENCODE	WRITE				
HITCHK	CALLED BY CALLS	COPE ABRTTL	URAN31					
IDCHAR	CALLED BY	NUMRIC	SEPREC					
INITLZ	CALLED BY CALLS CALLS (SYS)	COPE DEFAULT LOCF	USFT OPENMS	READMS	SYSTEMC			
INPUT	CALLED BY CALLS CALLS (SYS)	COPE CCPLOT PENAME ABS READMS	COMMNT PPLOT DECODE READ	CREAD PSMOKE ENCODE SYSTEMC	ECHO SEPREC EOF WRITE	FINDIT USFT FLOAT	HEADER LOCF	
LOSCHK	CALLED BY CALLS CALLS (SYS)	COPE ABRTTL AMAX1	URAN31					
MINTRN	CALLED BY CALLS CALLS (SYS)	COPE ABRTTL AMAX1	URAN31					
NOREC	CALLED BY CALLS (SYS)	STRACE	WRITE					
NUMRIC	CALLED BY CALLS CALLS (SYS)	FINDIT IDCHAR DECODE	ENCODE	WRITE				
OUTPUT	CALLED BY CALLS CALLS (SYS)	COPE FFRONT FLOAT	WRITE					
PECHK	CALLED BY CALLS CALLS (SYS)	COPE ABRTTL AMAX1	URAN31 FLOAT	MAX0	MIN0	MIN1		
PEIDNT	CALLED BY CALLS (SYS)	HEADER ABS	DECODE	FLOAT	MOD			
PENAME	CALLED BY CALLS (SYS)	INPUT ABS	ENCODE					

TABLE 8-2 PROGRAM UNIT REFERENCES (SUBROUTINES, FUNCTIONS, AND MAIN PROGRAM) - CONT'D

PKCHK	CALLED BY CALLS CALLS (SYS)	COPE ABRTTL MINØ	URAN31				
PPLDT	CALLED BY CALLS CALLS (SYS)	INPUT COMMNT ENCODI					
PSMOKE	CALLED BY CALLS (SYS)	INPUT AMIN1	FLOAT				
REINTZ	CALLED BY CALLS (SYS)	COPE MOD	WRITE				
RNDREL	CALLED BY CALLS	COPE ABRTTL	URAN31				
RNGCHK	CALLED BY CALLS	COPE ABRTTL					
SEPREC	CALLED BY CALLS CALLS (SYS)	CREAD COMMNT ENCODI	INPUT IDCHAR				
SMOKE	CALLED BY CALLS	COPE ABRTTL	URAN31				
SMPLOD	CALLED BY CALLS CALLS (SYS)	DETCTN URAN31 STRACE	GETRNG WRITE	GETTIM			
TIMCHK	CALLED BY CALLS	COPE ABRTTL					
TITLE	CALLED BY CALLS (SYS)	COPE DATE	TIME	WRITE			
URAN31	CALLED BY	ARTCHK LOSCHK SMPLOD	DECHK MINTRN WARNDØ	DUST PECHK	GETTIM PKCHK	GETVIS RNDREL	HITCHK SMOKE
USET	CALLED BY	INITLZ	INPUT				
VISCHK	CALLED BY CALLS	COPE ABRTTL					
WARNDØ	CALLED BY CALLS	COPE ABRTTL	URAN31				

TABLE 8-3 SYSTEM AND IMSL ROUTINE REFERENCES

ABS	CALLED BY	INPUT	PEIDNT	PENAME			
AMAX1	CALLED BY	PECHK	LDOSCHK	MINTRN	PECHK		
AMIN1	CALLED BY	PSMOKE					
DATE	CALLED BY	TITLE					
DECODE	CALLED BY	CREAD	HEADER	INPUT	NUMRIC	PEIDNT	
ENCODE	CALLED BY	CCPLOT	COMMNT	CREAD	PEIDNT	HEADER	INPUT
		NUMRIC	PENAME	PEIDNT	SEPRED		
EOF	CALLED BY	INPUT					
FLOAT	CALLED BY	GETENG	INPUT	OUTPUT	PECHK	PEIDNT	PSMOKE
GGAMA	CALLED BY	GAMMA					
IFIX	CALLED BY	FINDIT					
LOCFL	CALLED BY	FINDIT	INITLZ	INPUT			
MAX0	CALLED BY	PECHK					
MIN0	CALLED BY	PECHK	PECHK				
MIN1	CALLED BY	PECHK					
MOD	CALLED BY	PEIDNT	REINTZ				
OPENMS	CALLED BY	INITLZ					
READMS	CALLED BY	FINDIT	INITLZ	INPUT			
READ1	CALLED BY	CREAD	INPUT				
STRACE	CALLED BY	GETENG	GETVIS	NOREC	SMPLOC		
SYSTEMC	CALLED BY	FINDIT	INITLZ	INPUT			
TIME	CALLED BY	TITLE					
WRITE	CALLED BY	COMMNT	ECHO	FINDIT	HEADER	INPUT	NOREC
		NUMRIC	OUTPUT	REINTZ	SMPLOC	TITLE	

NOTE: ALL ROUTINES LISTED IN THE FIRST COLUMN OF THIS TABLE (EXCEPT GGAMA) ARE STANDARD SYSTEM ROUTINES OF CDC FORTRAN EXTENDED VERSION 4. GGAMA IS A SUBROUTINE FROM THE INTERNATIONAL MATHEMATICAL AND STATISTICAL LIBRARY (IMSL).

TABLE 8-4 COPE SUBROUTINE "TREE"

```

COPE----->ADDTBR
I----->ADDTBF
I----->ARTCHK----->ABRTTL
V          I----->URANS1
I----->BLOCHK----->ABRTTL
I----->DETCON----->ABRTTL
V          I----->SMPLCD----->URANS1
I----->DFCHK----->ABRTTL
V          I----->URANS1
I----->DUST----->ABRTTL
V          I----->URANS1
I----->GETRNG----->SMPLCD----->URANS1
I----->GETTIM----->ABRTTL
V          I----->CAMMA
V          I----->SMPLCD----->URANS1
V          I----->URANS1
I----->GETVIS----->URANS1
I----->DRITCHK----->ABRTTL
V          I----->URANS1
I----->INITLZ----->DEFAULT
V          I----->USET
I----->INPUT----->CCPLOT----->COMMNT
V          I----->COMMNT
V          I----->CREAD----->SEPREC----->COMMNT
V          V          I----->IDCHAR
V          I----->ECHO
V          I----->FINDIT----->NUMRID----->IDCHAR
V          I----->HEADER----->COMMNT
V          V          I----->SPIDNT
V          I----->PENAME
V          I----->PFLOT----->COMMNT
V          I----->PSMOKE
V          I----->SEPREC----->COMMNT
V          V          I----->IDCHAR
V          I----->USET
I----->BLOSCCHK----->ABRTTL
V          I----->URANS1
I----->SMINTRN----->ABRTTL
V          I----->URANS1
I----->OUTPUT----->FPACNT
I----->PECHK----->ABRTTL
V          I----->URANS1
I----->PKCHK----->ABRTTL
V          I----->URANS1
I----->REINTZ
I----->RNDREL----->ABRTTL
V          I----->URANS1
I----->RNGCHK----->ABRTTL
I----->SMOKE----->ABRTTL
V          I----->URANS1
I----->TIMCHK----->ABRTTL
I----->TITLE
I----->VISCHK----->ABRTTL
I----->WARNDG----->ABRTTL
V          I----->URANS1

```

NOREC

CHAPTER 9

9. SUBPROGRAMS

This chapter explains in moderate detail what each subprogram of COPE does. All four standard FORTRAN subprogram types (main program, block data, function, and subroutine) are present in COPE. In the following, the subprograms are discussed in this order: main program, block data, functions (in alphabetical order), and subroutines (in alphabetical order).

This chapter deals only with subprograms defined by COPE. System subroutines and functions are divided into three classes: standard FORTRAN IV routines, routines peculiar to CDC FORTRAN 4 Extended, and IMSL routines. The reader is assumed to be familiar with the first of these classes (or at least have access to a FORTRAN manual). System routines in the remaining two classes are discussed to some extent in Chapter 13.

9.1 COPE Main Program.

The COPE main program is well summarized by the flowchart in Figure 8-1.

The main program begins by calling routines that write the title page (TITLE) and initialize case independent variables (INITLZ).

Next the main program calls INPUT to read the first (or next) case's input. It then calls REINTZ which re-initializes certain case dependent counters and random number seeds. It then initializes the current replication number at one and the number of FO's (DO's) killed so far in the case at zero. The steps in this paragraph begin the "case loop."

The main program next begins the "replication loop" for the first (or next) replication of the current case. This loop begins by initializing the number of the current round to one and then setting various times, flags, and a counter.

The "replication loop" now continues by calling a sequence of subroutines which perform the tests described in Chapter 2 and sample weather, range-LOS, and time data.

<u>routines</u> (in order called)	<u>purpose</u>
ARTCHK	perform test 1
GETVIS	sample weather distribution
GETRNG	sample acquisition range and LOS segment distributions
VISCHK	perform test 2
RNGCHK	perform test 3
DETCTN	perform test 4
SMOKE	perform test 5
DUST	perform test 6
BLOCHK	perform test 7
GETTIM	sample time distribution(s) and perform tests 8 and 9
BLOCHK	perform test 10
TIMCHK	perform test 11 (for "shoot- ing gallery" only)
ADDTOP	calculate round arrival time

Each of these routines is explained in detail later in this chapter.

Note that if any routine that performs one (or more) of the tests is called and the test is failed, then the subsequent routines in the above list are not called; instead, control transfers to statement 140 (end of "replication loop") where a check is made to determine whether all of the replications for this case have been done. If they have, OUTPUT is called to print case results and then the end of the "case loop" is either hit and control returns to statement 100 which calls INPUT to read the input for the next case or, if the last case has been run, control passes to the STOP and the program terminates. If not all replications for the current case have been completed, then the number of the current replication is increased by one and control returns to statement 110 (which is the start of the "replication loop") to begin next replication.

If none of the tests are failed and the program reaches ADDTOP on the current replication, then the "round loop" begins (with statement 120). The "round loop" consists of a further sequence of subroutines which are called to perform those tests that may affect only one round.

<u>routines</u> (in order called)	<u>purpose</u>
LOSCHK	perform test 12
WARNFO	perform test 13
DFCHK	perform test 14
RNDREL	perform test 15
MINTRN	perform test 16
PECHK	perform test 17
HITCHK	perform test 18
PKCHK	perform test 19

Again, each of these routines is explained in detail later in this chapter.

If any subroutine that performs one of these tests (12 through 19) is called and the test is failed, then the subsequent routines in the above list are not called for the current round; instead, control transfers to statement 130 (end of "round loop") where a check is made to determine whether all of the rounds for this replication have been done. If they have, control transfers to statement 140 (end of "replication loop") where the check is made to determine whether more replications are to be done for the current case. If not all rounds for this replication have been completed, then the time the next round arrives is computed (ADDTBR), the round number is incremented by one, the end replication (or ENDREP) flag is reset and control transfers to statement 120 (start of "round loop") to begin the next round.

If no test is failed, then control reaches statement 130 after the entire sequence of subroutines in the "round loop" has been called. At this point a check is made to determine whether all rounds for this replication have been done. Control then proceeds just as in the case when statement 130 is reached due to a failed test as described in the paragraph above.

When all the rounds for a replication are completed, a check is made to determine whether more replications remain to be done (end of "replication loop"). If more replications remain, control transfers to 110 (start of "replication loop") to begin the next replication. If all replications for the current case are completed, then OUTPUT is called to print results for current case (end of "case loop").

When a case ends and its results are printed, the program checks whether the last case flag (ENDFLG) has been set. If it has, the program stops; if it has not been set, then control goes to statement 100 and INPUT is called to read the next cases input. If there is input for another case, it is read and the entire process from INPUT through end of "case loop" is repeated; if there is no input for further cases, the program terminates.

9.2 BDATA1 Block Data Subprogram.

BDATA1 consists entirely of data statements used to initialize variables in various COMMON blocks. These blocks are listed in section 4.7 and are their variables defined individually in Chapter 12 (Glossary of Variables).

9.3 FPRCNT Function.

This function formats percentages. It takes a numerical argument XIN and a logical argument PAREN and assigns a one word alphanumeric character string as the value of FPRCNT.

If XIN is negative, then FPRCNT is assigned:
the left justified character string UNDEF, when PAREN is true;
the right justified character string UNDEF, when PAREN is false.

If $XIN \geq 100.05$, a STOP occurs with an error message (see section 7.5.7).

If XIN is positive and less than 100.05, then $FPRCNT$ is assigned:

a left justified character string of the form $X.XX\%$, $XX.XX\%$, or 100.0% , when $PAREN$ is true;

a right justified character string of the form $X.XX\%$, $XX.XX\%$, or 100.0% , when $PAREN$ is false.

The character strings created by $FPRCNT$ are used by $OUTPUT$ to print percentages. $FPRCNT$ is called only from $OUTPUT$.

9.4 GAMMA Function.

This function takes two arguments, one real parameter $ALPHA$ and one integer IRN , and returns a real value as $GAMMA$.

$ALPHA$ is the α -parameter of the gamma distribution's probability density function:

$$g(x) = \frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x}$$

$GAMMA$ is assigned a value obtained by randomly sampling from the above distribution. This sampling is accomplished by calling the $IMSL$ subroutine $GGAMA$ with random number seed IRN .

The function $GAMMA$ itself then really does nothing more than dimension two arrays required by $GGAMA$, define an additional variable used by $GGAMA$, call $GGAMA$ with $ALPHA$, IRN , the two arrays and the additional variable as arguments, and then set $GAMMA$ equal to the random sample obtained by $GGAMA$ from the gamma distribution.

9.5 IDCHAR Function.

The $IDCHAR$ function takes three arguments, one real array A and two integer variables N and I . It returns an integer value.

The array A of dimension N should be filled with left justified characters, one character per word. (Most often A is dimensioned with $N=80$ and consists of a card image read under 80A1 Format in $INPUT$). The function $IDCHAR$ then examines the I th element of the array, that is $A(I)$.

The function IDCHAR is then assigned a value as follows:

```
IDCHAR = 0  if A(I) is a number or minus sign (hyphen);  
IDCHAR = 1  if A(I) is a letter or an apostrophe;  
IDCHAR = 2  if A(I) is a decimal point (period);  
IDCHAR = 3  if A(I) is a blank;  
IDCHAR = 4  if A(I) is a separator (i.e., a comma, a left or  
             right parenthesis, a slash, or a question mark);  
IDCHAR = 5  if A(I) is a dollar sign; and  
IDCHAR = 6  if A(I) is anything else.
```

9.6 NUMRIC Function (Logical Valued).

The NUMRIC function takes one real array argument, B, and three integer variable arguments N1, N2, and IB.

The array B of dimension N1 by N2 (i.e., DIMENSION B(N1, N2)) is checked to determine whether its IBth line (that is, entries B(J, IB) for J = 1, 2, 3, ..., N2) is numeric data (i.e., whether it consists only of numerals, minus sign, and decimal point possibly with trailing blanks).

If the line is numeric data, NUMRIC is assigned the value TRUE. If the line contains some non-numeric characters (letters, commas, etc.), then NUMRIC is assigned the value FALSE.

The IBth line of the B array is treated as a left justified character string (10 characters per word) which is examined character by character. Only the first 80 characters of the line are examined.

When the line being checked meets certain special conditions, other actions may be taken: If more than one decimal point is encountered, an error stop with message occurs. If no decimal point is encountered but the characters are otherwise numeric data, then a decimal point is added to B at the end of the non-blank characters, NUMRIC is assigned the value TRUE, and a return occurs. If the data are numeric but consist of more than ten non-blank characters (including the added decimal point, if any), then an error stop with message occurs. If the data contain imbedded blanks, then checking stops at the first imbedded (i.e., non-trailing) blank and the program behaves as though every character following were a blank. (This should have no effect in practice since the only imbedded blanks occur in PEDATA record names entered with the \$ preceding them. Those names will already have failed the numeric test before ever reaching their first imbedded blanks).

9.7 URAN31 Function.

This function takes one integer argument I and returns a real value between 0 and 1 (exclusive).

This is the "random" number generator used to sample most COPE data and to generate "random" numbers to be compared against probabilities to determine whether various tests are passed.

The best results are obtained when I is an odd integer in the range from 0 to 67108864. URAN31 generates a sequence of "random" numbers by replacing I by $3125 * I \pmod{67108864}$ and then setting URAN31 equal to I divided by 67108864. (That is, a linear congruential type random number generator $A(N+1) = A(N) * B \pmod{C}$ where $B = 3125$ and $C = 2^{26}$). Then repeating the process with the new value of I gives the next "random" number.

The function is written so that overflow will not occur even on machines with a word size as small as 32 bits.

Normally, one would not define the random number generator explicitly in the program code since one is available on nearly every computer as a standard system function. However, because users on other machines may wish to run the sample cases (see appendices) to check for correct implementation, the URAN31 function has been built into COPE as its standard (system independent) random number generator.

9.8 ABRTTL Subroutine.

This subroutine takes the integer argument IABRT which is the number of the test whose failure caused ABRTTL to be called.

ABRTTL updates the test abort total array NABORT by increasing the proper entry of the NABORT array by one. For mission aborts, the elements corresponding to all rounds of the IABRTth test are increased; for round aborts, only the element corresponding to the IABRTth test and current (KRFth) round is increased.

In addition, the flag ENDREP is set to TRUE to indicate that a test has been failed and the current replication (or round) need not continue through the subsequent tests.

9.9 ADDTBR Subroutine.

This subroutine adds TBR (time between rounds) to TIMRA (time round arrived) to obtain the new value for TIMRA (time next round arrives).

9.10 ADDTOF Subroutine.

This subroutine adds TOF (time of flight) to TIMNOW (time gun is ready to fire Copperhead) to obtain the TIMRA (time first round arrives).

9.11 ARTCHK Subroutine.

This subroutine performs the test to determine whether the FO (or DO, designator operator) has been killed or suppressed by RED preparatory artillery fires on this replication.

9.12 BLOCHK Subroutine.

This subroutine performs the bail out checks (tests 7 and 10).

9.13 CCPLLOT Subroutine.

This subroutine would produce a Cal-Comp plot of case results, if implemented. At the present time it merely produces a comment that no plot was produced for the case.

9.14 COMMNT Subroutine.

This subroutine takes four arguments: one real array A containing the character string that is the comment, one integer N1 which is the dimension of A, and two logical flags that indicate whether the comment is a user comment and whether the comment is a program generated "\$\$\$" comment.

The subroutine writes the comment immediately (along with a "\$\$\$" comment page heading, if needed) if the comment is a "\$\$\$" comment (either user or program generated). If the comment is a "\$\$" comment, then it is encoded into the CMEN array for later printing with the current case. If the comment is a "\$" comment, it is ignored and a return occurs.

Note that user comments are in format 80A1 with DIMENSION A (80) whereas program generated comments are in format 8A10 with DIMENSION A(10). This difference need not concern the casual user, but is required knowledge if one is to modify the program in so far as comments are concerned.

9.15 CREAD Subroutine.

This subroutine is called by INPUT whenever the user exercises the temporary option. The second field on the TEMPORARY card will be used by subroutine INPUT to set the value of IT2 equal to the keyword number for which the temporary option is being used.

CREAD then sets ITYPE equal to IT2-20 and executes a computed GO TO on ITYPE. As a result of this computed GO TO, control transfers to one of eight sections where the formatted temporary data is read. Each of these sections corresponds to one of the eight data blocks and each consists mainly of READ statements that fill in that data block's variables. When the reading is completed, control branches to statement number 410 where a return occurs.

9.16 DETCTN Subroutine.

This subroutine samples the detection time distribution to obtain the detection delay time for this replication (i.e., time from unmask until D.O. initiates call for fire. It includes detection, acquisition, and decision time).

If "random occurrence" is being played, control returns. If "shooting gallery" is being played, the detection delay time is compared with the line of sight duration (test 4). If the test is passed, a return occurs; if the test is failed the abort totaller is called (ABRTTL) prior to the return.

9.17 DFAULT Subroutine.

This routine merely sets the default values for many of the case descriptor parameters, data block record names, and flags. (It could really be replaced by some data statements to fill in these same variables if one so desires).

9.18 DFCHK Subroutine.

This subroutine performs test 14. If the FO (DO) has been killed on a previous call for the current replication, control transfers to 120 where the round abort is totalled (call to ABRTTL) followed by a return.

Otherwise, the routine calculates the current designator-to-target range. Then it uses that range value to interpolate in the DFFOKL array to obtain probability that direct fire kills the DO and probability that direct fire obscures but does not kill the DO. When these probabilities are obtained, test 14 is performed to determine whether the DO is unaffected, obscured but not killed, or killed. If the DO is unaffected, the routine returns; if the DO is obscured but not killed, the routine calls ABRTTL to increment the round abort counter and then returns; and, if the DO is killed, the routine increments the DO killed counter, sets the DO killed flag, then calls ABRTTL and returns.

9.19 DUST Subroutine.

This subroutine performs test 6 and calls ABRTTL if the test is failed.

9.20 ECHO Subroutine.

This subroutine takes the integer argument IUNIT. IUNIT is the logical I/O unit to which ECHO will write. (IUNIT = 8 currently)

Subroutine ECHO does nothing more than write out the current values of the parameters, flags, arrays, etc. that describe the conditions of the case. (see section 7.6) This printout is provided primarily as a debugging aid.

9.21 FINDIT Subroutine.

This subroutine takes a real array B of character strings dimensioned N1 by N2 and returns an integer valued array IT dimensioned N3. Normally N1 = 8, N2 = 10, and N3 = 10 so we have

DIMENSION B(8, 10), IT (10). The Kth line of B (i.e., B(J,K) for J = 1,2,3,...,8 with K fixed) consists of the character string found in the Kth field of the input card just read. This character string is compressed by eliminating blanks (the exception is that blanks occurring in PEDATA record names preceded by \$ are preserved when moved into the B array) and packing the results 10 characters per word into B.

Subroutine FINDIT compares the first line of the B array (B(1,1) through B(8,1)) with the keywords in the CHAR array in an attempt to find a match. If not successful, an error message and error stop occur; if a match is found, then IT(1) is set to the keyword number (contained in CHAR array also).

Next, if the match occurred with the TEMPORARY, OVERRIDE, or RESET keywords, then a second pass through the keyword list (CHAR array) is made to find a match with the second line of the B array. If this search is successful, IT(2) is set to the value of the matched keyword and a return occurs. If unsuccessful, an error message and stop occurs. An error message and stop also occur if a match is found for a keyword that is not allowed with TEMPORARY, OVERRIDE, or RESET.

If the first match was not with TEMPORARY, OVERRIDE, or RESET, then the routines checks CHAR (2, IMATCH) to see how many option word character strings are allowed with the IT(1)th keyword. If this number is zero, that is the flag to the program that it may have one of the keywords that takes only numerical values and so control transfers to statement 210. At statement 210, B(1,2) is checked. If B(1,2) is blank, a return occurs; if B(1,2) is numeric data, IT(2) is set to zero; if B(1,2) is neither blank nor numeric an error print occurs. Note that the blank occurs either when the keyword has no optionwords allowed (as with FIRST CASE, END, etc.) or when the keyword is omitted (i.e., the default valued is desired).

If CHAR(2, IMATCH) is non-zero, it gives the number of different option word character strings allowed with the current card's keyword. The TAPE 11 record name is then set to the abbreviated form of the keyword and the record of option word character strings allowed with the keyword is read into the CHARA array. The program then proceeds to match B(2,1) through B(2,8) with allowable option word choices for option word 1, B(3,1) through B(3,8) with allowable choices for option word 2, etc. If a match is not found for any line of B, then it is tested to see whether it is numeric; if not, an error message and stop occur, if it is numeric, the next line of B is checked. If line K of B is numeric, IT(K) is set to zero; if line K of B is blank, IT(K) is left at one (the value indicating default) to which it was set near the beginning of FINDIT; if line K of B matches one of the character strings allowed for option word K-1, then IT(K) is set equal to the number of the option word choice.

It is through setting the IT array that FINDIT signals subroutine INPUT as to which keyword and optionwords were found on the input card. Of course, in the case of numeric data or OVERRIDE record names, the values in the B array will have further use even after IT has been determined.

9.22 GETRNG Subroutine.

This subroutine samples the cumulative distribution of acquisition ranges (actually designator-to-target ranges at time of unmask) to obtain the initial designator-to-target range for the current replication. The cumulative distribution of line-of-sight segment lengths corresponding to the range bracket in which the designator-to-target range lies is then sampled to obtain the length of the segment of the target's path that is visible to the D0. This LOS segment length is then added to the target column's length and the sum is divided by the target velocity to obtain the duration of line of sight (i.e., the length of the time interval from target unmask until LOS to last vehicle in column is lost). Next a return is executed.

If the target velocity is zero (or negative) an error stop with message occurs.

9.23 GETTIM Subroutine.

This subroutine serves three basic functions: (1) perform test 8, (2) perform test 9, and (3) calculate the response time (time from D0 call for fire until gun is ready to fire Copperhead) for those missions that pass tests 8 and 9.

As mentioned in Chapter 2, there are three different ways of playing communication and processing delay times.

(1) If parameterized response time (PRSPTM) is non-zero, then this routine sets the time the gun is ready to fire (TIMNOW) equal to PRSPTM. It then performs test 8 and 9. If either test fails, ABRTTL is called and a return occurs; otherwise, ABRTTL is not called and the routine executes a return with TIMNOW = PRSPTM.

(2) If parameterized response time is zero, then the routine is to sample to obtain response time. If the variable FTSILL is TRUE, the routine samples the single response time distribution (FSRT array) to obtain the response time which is then added to the detection time to obtain the time at which the gun is ready to fire. It then performs tests 8 and 9. If either test fails, ABRTTL is called and a return occurs; otherwise, ABRTTL is not called and the routine executes a return with TIMNOW set at the value to be used for this replication.

(3) Finally, if parameterized response time is zero and the variable FTSILL is FALSE, then the routine uses the sum of the separate response time components as the response time. If digital communication is modeled, then the time to enter digital data into DMD is sampled and added to current time (TIMNOW which was previously set to DETTIM).

Then an unnumbered test is made to determine whether digital commo is successful. If it is, a transmission time is added to TIMNOW and control transfers to statement 160 (see below). If digital commo fails, then a test is performed to determine whether voice commo is successful. If voice is successful, then a time to switch from digital to voice and the voice commo transmission time are added to TIMNOW before control transfers to statement 160. If voice is also unsuccessful, then test 8 is considered to have been failed, ABRTTL is called, and a return executed.

If digital commo is not selected, then voice commo only is played. In this case, the test for successful voice commo is made (test 8) and if failed, an abort (call to ABRTTL) occurs followed by a return. If the test is passed, a voice commo time is added to TIMNOW and control reaches statement 160.

At statement 160, the battery computer system processing and battery response time are added to TIMNOW. Test 9 is then performed. If it fails, ABRTTL is called and a return executed; if it is passed, then TIMNOW is returned to be used for the current replication.

9.24 GETVIS Subroutine.

This subroutine samples the weather data used in the current case to obtain the cloud ceiling altitude and the visibility range limit.

The routine first randomly determines whether there is a cloud ceiling. Next it randomly determines whether there is a cloud free line-of-sight for the COPPERHEAD trajectory.

If there is a cloud-free line-of-sight, then the cloud ceiling altitude is set to its highest value and a visibility limit is determined by random sampling using the PRGCFL array. Control then branches to statement 160 (see below).

If there is no cloud-free line-of-sight, then the cloud ceiling altitude and visibility limit are both determined by a random sampling from the W array.

Note that the items of importance that are defined by this routine are the subscript of the cloud ceiling altitude (ICC), the subscript of the visibility range limit (IVL), and the visibility range limit itself (VISLIM). Also, note that ICC is initially a function of increasing altitude (that is, higher altitudes correspond to larger values of ICC); however, because the ordering convention for cloud ceilings was reversed between the weather data and the PE data, ICC is redefined after statement

160 by setting the new value of ICC equal to seven minus the old ICC value (so that higher altitudes correspond to lower values of ICC).

9.25 HEADER Subroutine.

This subroutine produces the heading page for each case.

The heading page is produced by encoding the lines for the first column of boxes into the AOUT array; those for the second column into the BOUT array; and those for the third column into the COUT array.

Next the routine generates some comments regarding time inconsistencies, record name overrides, and data read by the TEMPORARY option as applicable.

Finally, following the lines giving case number and number of replications (which are written before HEADER does any encoding) HEADER writes the AOUT, BOUT, and COUT arrays to produce the nine boxes of case description information that occur on the case header page. Then if any "\$\$" comments are to be printed for this case, the CMENT array is written (this includes both user comments and program generated comments).

9.26 HITCHK Subroutine.

This subroutine performs test 18.

First, it interpolates with respect to designator-to-target range (RNGNOW) in the proper row of the ITF array as determined by designator type and target posture (exposure) to obtain the probability of hit. This probability of hit is really the probability of hit given that the round engages (that is, that the seeker locks onto the laser spot on the target and does so within the maneuver footprint of the COPPERHEAD round).

Next the routine compares a random number (between 0 and 1) to the probability of hit to determine whether the current round hits. If the round hits, a return occurs; if not, ABRTTL is called to record the failure (miss) and a return occurs.

9.27 INITLZ Subroutine.

The subroutine initializes certain case independent values.

After setting IRAY and calling SYSTEMC to set the non-stand- and error recovery process, the routine proceeds to set the values of CHAR (2,I) for I = 1,2,...NREC from the values in the record named OPTNUMS on Tape 11. (This record is automatically updated by the PREPMS program each time new optionword character strings are added for a keyword).

The program next calls DFAULT to set the default values for the various case descriptors, record names, etc. Then USET is called to set the values to be used as the base to which the first case input card values are to be applied. Finally, MSAGLM is set and a return is executed.

9.28 INPUT Subroutine.

This subroutine reads the input lines for a given case and sets the variables in the program to the values required to run the case.

This routine is the longest and one of the most complex in the COPE model. It is most easily followed by means of a flowchart (see Chapter 11); however, an attempt to sketch its main sections is given here.

(1) First, the routine checks to determine whether there is another case to read. If not, it stops. If there is more to read, the routine first sets IRAY and calls SYSTEMC to establish non-standard error recovery. Then it clears various arrays and counters and increments the case number.

(2) Next it reads input cards until it encounters an END, ENDF, or end-of-file. For each card that it reads, it first checks whether that card is a comment card; if it is, subroutine COMMNT is called and the next card is read. If the card is not a comment, the routine calls SEPREC which separates the fields on the input card and moves each of them into a line of the B array. Then subroutine FINDIT is called to match each line in the B array against the keywords and option words allowed in that field. This matching determines the values of the IT array.

The program next checks to see whether the current card supersedes some of the previous cards for the current case. If it does, a comment is generated; in either case, the routine next branches to one of five statements depending on whether the card is a FIRST CASE, NEXT CASE, END, ENDF, or any other card, respectively. If the card is an END or ENDF, control transfers to statement 490 (see below). If the card is a NEXT CASE, RD (or equivalent), USET is called to establish the default values as the base to which changes caused by the input cards are to be applied; if the card is NEXT CASE, DR, then USET is not called and the changes caused by the input cards are applied to the conditions in effect for the previous case. Either type of NEXT CASE (or FIRST CASE) causes control to then read the next card. If the card is not an END, ENDF, NEXTCASE, or FIRST CASE, then control reaches statement 200.

At statement 200, a many branched computed GO TO is executed; the branch followed depends on the keyword number (IT1 or IT(1)). The actions taken for each case of the computed GO TO are varied but the main activities are to fill the UVALUE array with either numerical values, record names, or flag values. If the keyword was TEMPORARY, CREAD is called; if OVERWRITE, the record name is set; and if RESET, the default for that keyword is reactivated. Each case of the computed GO TO ends with a GO TO returning control to the line that reads the next input card.

(3) At line 490 the routine starts filling in the values of the various parameters that describe the current case (flags, arrays, single variables, etc.). These are generally filled in by calls to READMS to read records to fill in data blocks (record names as established when filling in UVALUE array), by equating variables to UVALUE numerical values, by looking up values in pre-defined arrays according to a subscript value in UVALUE array, by calling a special subroutine (in the case of PEDATA) to generate a record name and then reading that record from TAPE 11 into the data block, or by calling special routines (such as PSMOKE) that use inputs to calculate a parameter value. In any event, by the end of INPUT all variables required to run the case have been set and the replications can begin.

(4) INPUT finishes by calling HEADER and ECHO and then returning.

9.29 LOSCHK Subroutine.

This subroutine is used to determine whether the DO still has (macro-terrain) LOS to the target when the round arrives (test 12).

It performs the test in two different ways depending on the LOS model used.

For "shooting gallery", it merely compares the time the round arrives with the duration of the line-of-sight. If the round arrives before the line-of-sight ends, a return is executed; if the LOS ends before the round arrives, ABRTTL is called prior to the return.

For the "random occurrence" LOS model, the probability P of LOS to a single vehicle in the current designator-to-target range bracket is found. The probability that at least one vehicle is in view is then taken to be equal to $1-(1-P)^{XNV}$ where XNV is the number of target vehicles currently unkilld. Once this probability is obtained, it is compared to a random number and the result settles test 12. If test 12 is passed, a return occurs; if it is false, ABRTTL is called, then a return occurs.

9.30 MINTRN Subroutine.

This subroutine performs test 16 and sets target posture (exposure) with respect to mini-terrain.

First the current designator-to-target range is computed. Then this range is used to interpolate in each of the three rows of the PSTTBL array. As a result of this interpolation the PSTVAL array is filled with the probabilities of completely obscured target posture, fully exposed target posture, and hull defilade (turret exposed) target posture respectively.

A random number is then drawn and compared to these probabilities to determine which of the three target postures occurs. If the target posture is "completely obscured", then test 16 is failed, ABRTTL is called, and a return occurs; otherwise, ITGTPS is set to indicate the target posture (1 for fully exposed; 2 for hull defilade) to be used in hit and kill probability tests and a return occurs.

9.31 NOREC Subroutine.

This routine is called indirectly through the CDC non-standard error recovery facility (SYSTEMC routine explained in CDC Fortran 4 Extended) It is called whenever READMS attempts to read a record from TAPE 11 using a record name that does not correspond to any of the records on TAPE 11.

The routine writes a message giving the record name that could not be found, executes a subroutine call trace and then executes a stop with message.

9.32 OUTPUT Subroutine.

This subroutine writes the results page for each case.

The routine begins by clearing several arrays.

Next, it calculates the percentage of total missions that have a failure (abort) at each test (for "round" aborts, this percentage is calculated for each round). This is the ABRPCT array.

Then it calculates what percentage of the missions that reached a given test passed it. Again this is calculated for each round for the "round" aborts. This is the PASPCT array.

Next the program sets the number of occasions, attempted engagements, and shots.

Then it compares the number of kills as calculated by two different counters and, if there is a disagreement in value between them, it prints a message (see Section 7.3.1).

Next, it calculates the probability of kill given an occasion, the probability of kill given an attempted engagement, and the probability of kill given a shot. These are calculated separately for each round of the mission.

Then it calculates the probability of a shot given an occasion, probability of a shot given an attempted engagement, and probability of attempted engagement given a shot. It also calculates the probability of having line-of-sight on each round.

Next it compares the number of missions (or rounds, as appropriate) tested for each abort condition as computed in two different ways. If the results differ an error message is generated (see section 7.3.2).

Then the program calculates the FO (DO) survivability measures; that is, probability DO is killed and probability DO is not killed given that the mission reached that test.

Now that all of the values to be printed on the results page have been calculated the routine proceeds to print the results in the order they appear on the case results page (see Chapter 6).

Note that prior to printing percentages, function FPRCNT is applied to format the percentages for output. Also note that any percentage valued variables with the value -1.0 are printed as "UNDEF" indicating undefined value (this occurs when the percentage could not be calculated due to a zero denominator. It is for this reason that all percentage valued variables in OUTPUT are initialized at -1.0 and then only assigned values if the ratio used in computing the percentage has non-zero denominator).

9.33 PECHK Subroutine.

This subroutine performs test 17 (i.e., does COPPERHEAD round engage the target?)

First it calculates the delay time (DELTAT) for the current round. By delay time is meant the time the round arrives minus the expected arrival time (which is based on nominal response time). Next the value of IMUTS (the "MUTS" factor -- see Section 16.1) is determined and the range value indices to be used for interpolation are set.

If IMUTS is 2, it means that not all vehicles in the target column have passed the point of closest approach to the centroid of the COPPERHEAD footprint. That is, delay has no effect and so PE is computed by interpolating with respect to designator-target range alone in the PETBL array (using delay time equal to zero (IDELTT=1) and holding cloud ceiling and visibility limit constant at their previously sampled values (obtained in GETVIS)).

If IMUTS is 1, it means that all vehicles in the target column have passed the point of closest approach to the centroid of the COPPERHEAD footprint. That is, delay time has an effect and so PE is computed by interpolating with respect to both designator-target range and delay time while holding cloud ceiling and visibility limit constant.

In either case, once PE (probability that seeker engages target and target is within Copperhead maneuver footprint) is computed, it is compared to a random number sampled from a uniform distribution (from 0 to 1). If the random number is less than PE, a return occurs; otherwise, ABRTTL is called before the return occurs.

9.34 PENAME Subroutine and PEIDNT Entry.

This subroutine takes ten arguments. When the call is to PENAME the first nine are "input" and the last one is "result"; whereas a call to PEIDNT uses the current value of the tenth argument as "input" and returns the values for the other nine.

This subroutine is used to create PE data record names. The values of the nine parameters (see section 5.4) are taken as arguments (in the same order they are described in section 5.4). They are loaded into the AVALUE array and are searched for in the XVALUE array. When matches are found, the corresponding XVALUE entries (for XVALUE (I,J,2)) are used to construct the record name which is returned as IDCODE.

When PEIDNT is called, the record name is decomposed into the separate XVALUE (I,J,2) values (and TR value) used to create it. The corresponding XVALUE (I,J,1) values are found and equated to the nine parameters (note slightly different handling of TR, nominal response time). These values are then returned.

Note that while every choice of the nine parameters can give at most one record name, there may be more than one choice of parameters that can give that record name. For example, if all parameters are held constant except target velocity (VEL), then VEL = 2 m/s and VEL = 3 m/s will give the same record name (using the current XVALUE array). Hence, when that record name is decoded by PEIDNT to obtain the parameters to which it corresponds, it will have no way of choosing between 2 m/s and 3 m/s for VEL; instead it will merely take the first one that matches (2 m/s in this case).

9.35 PKCHK Subroutine.

This subroutine performs test 19 (i.e., did the COPPERHEAD kill the target?)

The routine looks up the PK (probability of kill given a hit) in an array as a function of target type and target posture (exposure). This value is then compared to a random number sampled from

a uniform distribution (from 0 to 1). If the random number is less than PK, then a kill occurs, the number of vehicles killed on the KRFth round for this case (RNDREC) is incremented, the number of vehicles killed (NVEHKL) on this replication is incremented and a return occurs. If not, ABRTTL is called and then the return occurs.

9.36 PLOT Subroutine.

This subroutine is currently just a "stub" that produces a "\$\$" program generated comment saying that no printer plot was produced for the current case.

If implemented, this routine would produce printer plots (bar graphs) giving a graphical presentation of case results.

9.37 PSMOKE Subroutine.

This routine calculates the probability that smoke aborts the potential COPPERHEAD mission. This probability is a function of the number of each of two types of smoke rounds fired (NSMK2 and NSMK5) and the distribution of Pasquill atmospheric stability categories, distribution of wind speed velocities, and distribution of relative humidity levels corresponding to the current case's weather (i.e., month and time of day).

The entire routine reduces to evaluating PSMKKL (probability smoke kills mission) according to the formula:

$$PSMKLL = \frac{1}{FRONT} \sum_{I=1}^3 \sum_{J=1}^3 \sum_{K=1}^2 \left\{ \min \left(\frac{NSMK2}{SMK2(I,J,K)} + \frac{NSMK5}{SMK5(I,J,K)}, FRONT \right) * \right. \\ \left. WNDSPD(I,J) * HUMID(K) * [PASQL(2*I-1) + PASQL(2*I)] \right\}$$

where all of the variables are as defined in the Glossary of Variables (Chapter 12).

Note that because certain combinations of windspeed and Pasquill category never occur, the SMK2(I,J,K) and SMK5(I,J,K) values corresponding to them have been left blank (i.e., equated to zero). To avoid the division by zero that would otherwise result, the range of J in the summation is sometimes limited in the routine to avoid certain SMK2 and SMK5 values which should have no weight in the sum anyway.

9.38 REINTZ Subroutine.

This subroutine resets certain counter arrays to zero. It also resets the random number seeds for each case and on the first case it writes the seeds on Tape 8.

Note that the seeds are set according to the formula:

$$IR(I) = 2*(MOD(8**I, 29) * 2**25/29) + 1$$

for $I = 1, 2, \dots, 28$.

This assures that each IR is odd and that the IR's are approximately uniformly spaced between 0 and 2^{26} .

9.39 RNDREL Subroutine.

This subroutine performs the test of round reliability (test 15).

9.40 RNGCHK Subroutine.

This subroutine performs test 3.

It compares designator-to-target range at unmask to the maximum designator range. If the maximum designator range is less than the designator-to-target range, ABRTTL is called and a return occurs; otherwise, the test is passed and a return occurs.

9.41 SEPREC Subroutine.

This subroutine takes a real array A of dimension N1 and breaks it up into fields (i.e., separates records) and compresses them one per line into the B array which is of dimension N2 by N3.

The routine first clears several arrays to be used in separating the fields on A. Then it calls IDCHAR once for each element of A to determine what kind of character it is. A is assumed to be an array (usually DIMENSION A(80)) of characters filled by reading one input card under 80A1 format.

If the routine encounters sequence numbers in columns 73-78 of the first card (i.e., elements 73-78 of array A) and blanks in 63-72 and 79-80, then it issues a "\$\$\$" comment (see section 7.1) and removes the sequence numbers on the first card and all subsequent cards (A arrays) of the current run. If it encounters something other than the blanks and sequence numbers as described above in columns 63-80 of the first card, then it leaves columns 73-78 unchanged on all cards of the current run.

Next the program separates the fields on the card (i.e., in A) and compresses them into B. It does this by copying one character at a time from A to PB omitting blanks (except when part of a ten

character string preceded by a "\$" sign) and starting a new line of PB every time a separator (comma, left or right parenthesis, question mark, or slash) is encountered. It then encodes each line of PB into a line of B so that each element of B contains up to ten characters.

This means that B(I,J) for $I = 1, 2, \dots, 8$ contains the characters of the Jth field of the input card compressed by removing blanks (with the exception noted) and packed ten characters per word (until the characters run out).

A return now occurs with the B array filled with the character strings to be compared to the keywords and option words allowed.

Note that an input card cannot have more than ten fields nor fewer than one. A field on an input card (or in A) is defined as a string of characters lying between two successive separators where for this definition we include the beginning of the card and the end of the card as separators and do not count characters that would otherwise be separators if they lie in a ten character string following a dollar sign. (The normal separators are mentioned above in this section).

9.42 SMOKE Subroutine.

It compares the value of PSMKKL calculated in PSMOKE with a random number sampled from a uniform distribution to determine whether smoke was sufficient to abort the potential Copperhead mission. As usual, if an abort occurs, ABRTTL is called prior to returning; otherwise, an immediate return occurs.

9.43 SMPLCD Subroutine.

This subroutine randomly samples from a cumulative distribution. As arguments, it takes an array A, its dimensions ND1, ND2, the row number NRS containing the values to be sampled, the number of entries NDR used in the row ($NDR \leq ND1$), and the random number seed IRNDM. It returns the value sampled XSMPL and the class ICLASS in which it falls.

This routine requires an array A with $A(I,1)$ for $I = 1, 2, \dots, NDR \leq ND1$ consisting of probabilities $A(1,1) = 0.0 \leq A(2,1) \leq \dots \leq A(NDR,1) = 1.0$. The routine samples a random number RN from a uniform distribution (from 0 to 1) and then brackets this number with two entries from $A(I,1)$ such that $A(I,1) \leq RN \leq A(I+1,1)$.

The corresponding entries $A(I,NRS)$ and $A(I+1,NRS)$ from the NRSth row of A are obtained. The value sampled XSMPL is then defined by straight line interpolation, (i.e., XSMPL is defined so that the point (RN, XSMPL) lies on the line segment joining $(A(I,1), A(I,NRS))$ to $(A(I+1,1), A(I+1,NRS))$).

Put in terms of a formula, that means:

$$XSMPL = A(I,NRS) + (RN - A(I,1)) * (A(I+1,NRS) - A(I,NRS)) / (A(I+1,1) - A(I,1)).$$

ICLASS is then defined to be equal to I and a return occurs.

Note that A defines a cumulative distribution. That is, when sampling.

$$\text{Pr}(\text{XSMPLE} \leq A(I, \text{NRS})) = A(I, 1) \text{ for } I = 1, 2, \dots, \text{NDR}.$$

Hence, the user must take care to use only cumulative distributions for the input data that will eventually be sampled.

9.44 TIMCHK Subroutine.

This subroutine performs test 11 (i.e., does the D0 still have line-of-sight to the target when the battery is ready to fire?)

It is called only for the "shooting gallery" LOS model.

9.45 TITLE Subroutine.

This routine merely writes the Title page of the output (see Chapter 6). It also sets the date and time that the current run began.

9.46 USET Subroutine.

This subroutine is called when the next case to be run is to use the baseline (default) values of the case parameters as the base to which its own input cards are to be applied.

The routine sets the UVALUE common block and a few other variables to their default values. Whatever inputs the next case then contains are applied to changing these variables from their defaults rather than from the previous cases values.

9.47 VISCHK Subroutine.

This subroutine performs test 2.

It compares designator-to-target range at unmask to the visibility range limit. If the visibility range limit is less than the designator-to-target range, ABRTTL is called and a return occurs; otherwise, the test is passed and a return occurs.

9.48 WARND0 Subroutine.

This subroutine performs test 13.

If the current round is the first of this COPPERHEAD mission, then a random number is sampled from a uniform distribution and compared to the probability that D0 is warned to lase to determine whether the D0 received the message to begin lasing. If he did receive the message, the test is passed and a return occurs; if not, ABRTTL is called prior to returning.

If the current round is not the first round of this COPPERHEAD fire mission, then D0 is assumed to have been warned either by communication or by sighting the impact of the first round and, in either case, turns on his laser so test 13 is passed and a return occurs.

CHAPTER 10

10. MEMORY ORGANIZATION: COMMON BLOCKS AND EQUIVALENCE CLASSES

10.1 Common Blocks.

This chapter describes the common blocks and equivalence classes in COPE. This organization of the values stored in memory is a matter of some importance to anyone attempting to modify the program.

Tables 10-1 and 10-2 give, respectively, a list of the common blocks in each subprogram of COPE and a list of the subroutines in which each common block occurs.

10.2 Equivalence Classes.

Table 10-3 gives the equivalence classes. There are twelve equivalence classes used in COPE (some are used in more than one subroutine). The equivalence classes are described below in alphabetical order of the names of their first items. Eight of the twelve equivalence classes are the eight data blocks referred in Section 4.5.1.:

ACQDAT is the Acquisition Range data block,
AINVDA is the Invariant data block,
DFS DAT is the Direct Fire Suppression data block,
PEDATA is the Probability of Engagement data block,
STDAT is the Target Posture Distribution data block,
RODATA is the Random Occurrence data block,
RSPDAT is the Response Time data block, and
WDATA is the Weather data block.

These data block equivalence classes (along with a MAX equivalence class which has nothing to do with COPE's use of PREPMS) are also present in the PREPMS program.

Each equivalence class is formed by an equivalence statement establishing that one array or non-subscripted variable occupies the same memory locations as a set of one or more other arrays or non-subscripted variables. In each case the one array or non-subscripted variable (after which the equivalence class is named in Table 10-3) occupies all of the computer memory used by that equivalence class and so is dimensioned with a value equal to the entire length of the equivalence class (if the length is 1, then the equivalence class occupies but one word of computer memory and the class is named after a non-subscripted variable).

In Table 10-3, the information about each equivalence class is separated by dashed lines and is read as follows:

The first entry on the left gives the name of the equivalence class.

The next entry (length) gives the length of the equivalence class in terms of the words of computer memory occupied by the members of the equivalence class. This length is also the dimension of the array (or non-subscripted variable if length is one) in the first entry.

The entries under "MEMBERS" give bias, name, and length of each member. That is, for each member (variable) of the equivalence class (other than the member in the first entry above) there are three items given: first the bias or number of words preceding the variable in that equivalence class in computer memory space, second the variable (or array) name, and third the length of the array (or non-subscripted variable) in terms of computer words (one for non-subscripted variables).

Note that for each equivalence class, the memory space occupied is part (or all) of some common block. This correspondence is as follows:

ACQDAT is in the RNGLOS common block,
AINVDA is in the HIT common block,
D is in the DVALUE common block,
DFSDAT is in the DODF common block,
ICODE is in the ICODE common block,
PEDATA is in the HIT common block,
PSTDAT is in the HIT common block,
RECODE is in the RECNAM common block,
RODATA is in the RANDOC common block,
RSPDAT is in the RSPTIM common block,
U is in the UVALUE common block, and
WDATA is in the WEATHR common block.

Also, note that the variables in an equivalence class may be used (under their names in the COMMON statements) in subprograms other than those in which the equivalence class is defined.

TABLE 10-1 COMMON BLOCKS IN EACH SUBPROGRAM OF THE COPE PROGRAM

SUBPROGRAM	COMMON BLOCKS							
ABRTTL	ABORT	FLAG	RPLCTN					
ADDTER	TIME							
ADDTDF	TIME							
ARTCHK	KTEST	MISC	RANDOM	RPLCTN				
BDATA1	ABRLBL LOGFLG	ACHAR POINT	DESRNG SMOKED	DISPLY STITLE	FLTTIM SYMBOL	FSRESP XVALUE	HEADNG	
BLOCHK	BAIL	KTEST	RANGE	RPLCTN	TARGET	TIME		
CCPLOT	RUNDAT							
COMMNT	COMENT							
COPE	ABRLBL FLTTIM SMOKED	ACHAR FSRESP STITLE	BAIL HEADNG SYMBOL	DESRNG LOGFLG TIME	DISPLY POINT XVALUE	DODF RPLCTN	FLAG RUNDAT	
CREAD	DODF	HIT	RANDOC	RNGLOS	RSPTIM	WEATHR		
DETCN	FLAG	KTEST	RANDOM	RPLCTN	RSPTIM	TIME		
DFAULT	DVALUE	RUNDAT						
DFCHK	DODF WEATHR	KTEST	RANDOM	RANGE	RPLCTN	TARGET	TIME	
DUST	KTEST	MISC	RANDOM	RPLCTN				
ECHO	BAIL PEDESC RUNDAT	DODF RANDOC SMOKED	FLAG RANGE TARGET	HIT RECNAM TIME	LOGFLG RNGLOS WEATHR	MISC RPLCTN	OVER RSPTIM	
FINDIT	ACHAR	ICODE	INDEX1	RUNDAT	SYMBOL			
GETRNG	RANDOM	RANGE	RNGLOS	RPLCTN	TARGET	TIME	WEATHR	
GETTIM	FLAG TIME	FSRESP	KTEST	LOGFLG	RANDOM	RPLCTN	RSPTIM	
GETVIS	RANDOM	RANGE	RPLCTN	WEATHR				
HEADER	BAIL OVER SMOKED	COMENT PEDESC SYMBOL	DISPLY RANGE TARGET	FLAG RECNAM TIME	HEADNG RPLCTN UVALUE	LOGFLG RSPTIM	MISC RUNDAT	
HITCHK	HIT TIME	KTEST	RANDOM	RANGE	RPLCTN	RSPTIM	TARGET	
IDCHAR	SYMBOL							

TABLE 10-1 COMMON BLOCKS IN EACH SUBPROGRAM OF THE COPE PROGRAM - CONTINUED

SUBPROGRAM	COMMON BLOCKS						
INITLZ	ABORT UVALUE	ACHAR	DVALUE	FLAG	ICODE	INDEX1	RUNDAT
INPUT	ABORT FLAG OVER RNGLOS TIME	BAIL FLTTIM PEDESC RPLOTN UVALUE	COMMENT HIT POINT RSPTIM WEATHR	DESRNG ICODE RANDOM RUNDAT	DISPLY INDEX1 RANDOM SMOKED	DODF LOGFLG RANGE SYMBOL	DVALUE MISC RECNAM TARGET
LOSCHK	FLAG TIME	KTEST	RANDOC	RANDOM	RANGE	RPLOTN	TARGET
MINTRN	HIT	KTEST	RANDOM	RANGE	RPLOTN	TARGET	TIME
NOREC	ICODE						
NUMRIC	SYMBOL						
OUTPUT	ABORT SCCSS	ABRLBL	DODF	FLAG	KTEST	RPLOTN	RUNDAT
PECHK	FLAG TARGET	HIT TIME	KTEST WEATHR	RANDOM	RANGE	RPLOTN	RSPTIM
PENAME	XVALUE						
PKCHK	HIT	KTEST	RANDOM	RPLOTN	SCCSS	TARGET	
RPLOT	RUNDAT						
PSMOKE	SMOKED	WEATHR					
REINTZ	ABORT	KTEST	RANDOM	RPLOTN	SCCSS		
RNDREL	KTEST	MISC	RANDOM	RPLOTN			
RNGCHK	KTEST	RANGE	RPLOTN				
SEPREC	COMMENT	LOGFLG	SYMBOL				
SNOKE	KTEST	RANDOM	RPLOTN	SMOKED	WEATHR		
TIMCHK	FLAG	KTEST	RPLOTN	TIME			
TITLE	RUNDAT	STITLE					
USFT	DVALUE	LOGFLG	OVER	UVALUE			
VISCHK	KTEST	RANGE	RPLOTN				
WARNDG	FLAG	KTEST	RANDOM	RPLOTN	RSPTIM	TIME	

TABLE 10-2 SUBPROGRAMS OF COPE IN WHICH EACH COMMON BLOCK OCCURS

COMMON	SUBPROGRAMS							
ABORT	ABRTTL	INITLZ	INPUT	OUTPUT	REINTZ			
ABRLEB	BDATA1	OUTPUT						
ACHAP	BDATA1	FINDIT	INITLZ					
BAIL	BLOCKK	COPE	ECHO	HEADER	INPUT			
COMENT	COMMNT	HEADER	INPUT	SEPPEC				
DESRNG	BDATA1	INPUT						
DISPLY	BDATA1	HEADER	INPUT					
DOOF	COPE	CREAD	DFCHK	ECHO	INPUT	OUTPUT		
DVALUE	DEFAULT	INITLZ	INPUT	USET				
FLAG	ABRTTL	COPE	DETCTN	ECHO	GETTIM	HEADER	INITLZ	
	INPUT	LOSCHK	OUTPUT	PECHK	TIMCHK	WARNDQ		
FLTTIM	BDATA1	INPUT						
FSRESP	BDATA1	GETTIM						
HEADNG	BDATA1	HEADER						
HIT	CREAD	ECHO	HITCHK	INPUT	MINTRN	PECHK	PKCHK	
ICODE	FINDIT	INITLZ	INPUT	NOREC				
INDEX1	FINDIT	INITLZ	INPUT					
KTEST	ARTCHK	BLOCKK	DETCTN	DFCHK	DUST	GETTIM	HITCHK	
	LOSCHK	MINTRN	OUTPUT	PECHK	PKCHK	REINTZ	RNDREL	
	RNGCHK	SMOKE	TIMCHK	VISCHK	WARNDQ			
LOGFLG	BDATA1	ECHO	GETTIM	HEADER	INPUT	SEPPEC	USET	
MISC	ARTCHK	DUST	ECHO	HEADER	INPUT	RNDREL		
OVER	ECHO	HEADER	INPUT	USET				
PEDESC	ECHO	HEADER	INPUT					
POINT	BDATA1	INPUT						
RANDOC	CREAD	ECHO	INPUT	LOSCHK				
RANDOM	ARTCHK	DETCTN	DFCHK	DUST	GETRNG	GETTIM	GETVIS	
	HITCHK	INPUT	LOSCHK	MINTRN	PECHK	PKCHK	REINTZ	
	RNDREL	SMOKE	WARNDQ					
RANGE	BLOCKK	DFCHK	ECHO	GETRNG	GETVIS	HEADER	HITCHK	
	INPUT	LOSCHK	MINTRN	PECHK	RNGCHK	VISCHK		

TABLE 10-2 SUBPROGRAMS OF COPE IN WHICH EACH COMMON BLOCK OCCURS - CONT'D

COMMON	SUBPROGRAMS							
RECNAM	ECHO	HEADER	INPUT					
RNGLOS	CREAD	ECHO	GETRNG	INPUT				
RPLCTN	ABRTTL ECHO LOSCHK RNGCHK	ARTCHK GETRNG MINTRN SMOKE	BLOCHK GETTIM OUTPUT TIMCHK	COPE GETVIS PECHK VISCHK	DETCN HEADER PKCHK WARND0	DFCHK HITCHK REINTZ	DUST INPUT RNDREL	
RSPTIM	CREAD PECHK	DETCN WARND0	ECHO	GETTIM	HEADER	HITCHK	INPUT	
RUNDAT	CCPLOT INPUT	COPE OUTPUT	DEFAULT PLOT	ECHO TITLE	FINDIT	HEADER	INITLZ	
SCCSS	OUTPUT	PKCHK	REINTZ					
SMOKED	BDATA1	ECHO	HEADER	INPUT	PSMOKE	SMOKE		
STITLE	BDATA1	TITLE						
SYMBOL	BDATA1	FINDIT	HEADER	IDCHAR	INPUT	NUMRIC	REP'REC	
TARGET	BLOCHK LOSCHK	DFCHK MINTRN	ECHO PECHK	GETRNG PKCHK	HEADER	HITCHK	INPUT	
TIME	ADDTBR GETRNG PECHK	ADDTOF GETTIM TIMCHK	BLOCHK HEADER WARND0	COPE HITCHK	DETCN INPUT	DFCHK LOSCHK	ECHO MINTRN	
UVALUE	HEADER	INITLZ	INPUT	USET				
WEATHR	CREAD PSMOKE	DFCHK SMOKE	ECHO	GETRNG	GETVIS	INPUT	PECHK	
XVALUE	BDATA1	PENAME						

TABLE 10-3 COPE EQUIVALENCE CLASSES

NAME (Name of First Item)	LENGTH	MEMBERS		
		Bias	Name	Length
ACQDAT	161	0	NRP	1
		1	NPP	1
		2	NRNGCL	1
		3	CRNGD	22
		25	SEGLOC	121
		146	RNGCLB	11
		157	VELTBL	4

(Occurs in subroutines CREAD and INPUT)

AINVDA	167	0	DLTT	6
		6	PKTBL	20
		26	NRNGTT	1
		27	RNGTTF	20
		47	TTF	120

(Occurs in subroutines CREAD and INPUT)

D	70	0	DVALUE	70

(Occurs in subroutine INPUT)

DFSDAT	31	0	DFDOKL	30
		30	NDFSP	1

(Occurs in subroutine CREAD and INPUT)

ICODE	1	0	CODE	1

(Occurs in subroutines FINDIT, INITLZ, and INPUT)

PEDATA	4260	0	INDEX	60
		60	PETBL	4200

(Occurs in subroutines CREAD and INPUT)

TABLE 10-3 COPE EQUIVALENCE CLASSES - CONT'D

NAME (Name of First Item)	LENGTH	MEMBERS		
		Bias	Name	Length
PSTDAT	41	0	NRNGPS	1
		1	RNGPST	10
		11	PSTTBL	30

(Occurs in subroutines CREAD and INPUT)

RECODE	8	0	WCODE	1
		1	ACCODE	1
		2	RSCODE	1
		3	DFCODE	1
		4	ROCODE	1
		5	PECODE	1
		6	PSCODE	1
		7	AICODE	1

(Occurs in subroutine (HEADER))

RODATA	22	0	TCRIT	1
		1	NPLOSR	1
		2	RNGPLS	10
		12	PRBLOS	10

(Occurs in subroutines CREAD and INPUT)

RSPDAT	49	0	BATRTM	3
		3	XMVADF	2
		5	DUMI	1
		6	BCSPTM	4
		10	XMTTIM	4
		14	TMEAN	4
		18	TSIGMA	4
		22	DETTMA	20
		42	NDT	1
		43	TRARRY	6

(Occurs in subroutines CREAD and INPUT)

TABLE 10-3 COPE EQUIVALENCE CLASSES - CONT'D

NAME (Name of First Item)	LENGTH	Bias	MEMBERS	
			Name	Length
U (Occurs in subroutines HEADER and INPUT)	70	0	UVALUE	70

WDATA	193	0	PRCLCG	1
		1	PRCFLS	2
		3	PRGCFL	22
		25	W	132
		157	VIS	11
		168	CLOUD	6
		174	NCC	1
		175	NVL	1
		176	PASQL	6
		182	WNDSPD	9
		191	HUMID	2
(Occurs in subroutines CREAD and INPUT)				

CHAPTER 11

11. FLOWCHARTS

This chapter includes flowcharts of the COPE main program and subroutine INPUT. Flowcharts of the other subroutines are not included since they are generally simple enough to understand by reading the descriptions in chapter 9 and studying the actual program code with the aid of the glossary of variables in chapter 12.

Several conventions are observed in the flowcharts. First, connectors on-page are indicated by circles whereas connectors off-page are indicated by pentagons. Numbered connectors correspond to actual statement numbers in the program code whereas letter connectors are merely flowcharting conveniences. Each STOP block in the flowchart includes the section number of this report that explains that stop.

A list of abbreviations used in the flow charts is included on the next page.

ABBREVIATIONS USED IN FLOW CHARTS

acq	Acquisition	OW	Option word
CLGP	COPPERHEAD	param	parameterized
C.O.	Completely obscured	pr	priority
comt	comment	prob	probability
desig	designator	rd(s)	round(s)
DF	direct fire	resp	response
dgt	digital	R.N.	random number(s)
D.O.	designator operator	rng	range
F.E.	fully exposed	SG	shooting gallery
fn	function	smk	smoke
H.D.	hull defilade	supp	suppression
hdng	heading	temp	temporary
KW	keyword	tgt	target
max	maximum	vel	velocity
mo	month	vis	visibility
msg	message	w/	with
msn	mission	#	number
no	number		
nt	night		
opt	option		

Flowchart of Main Program of COPE

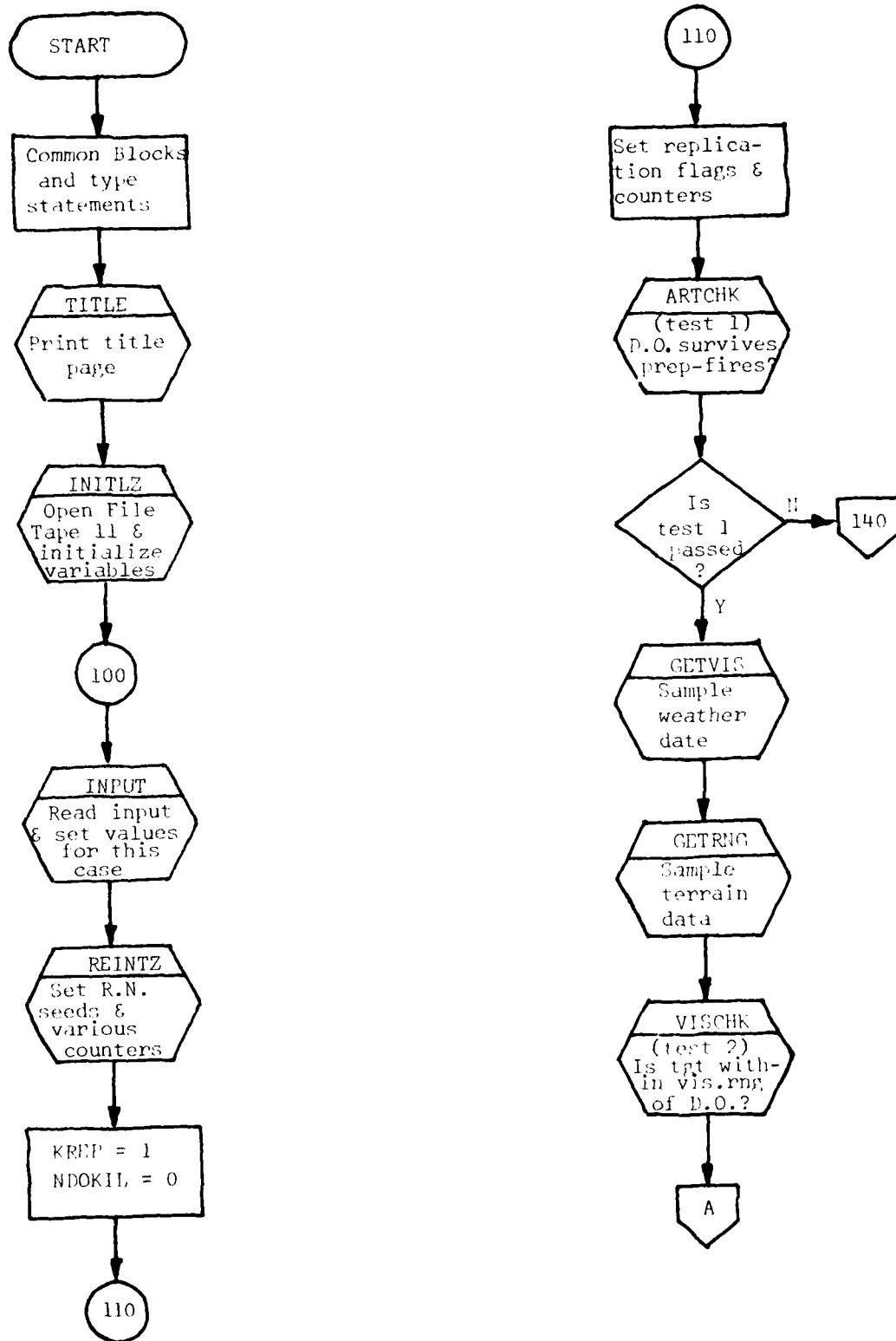


FIGURE 11-1 Flowchart of Main Program of COPE (Page 1 of 5)

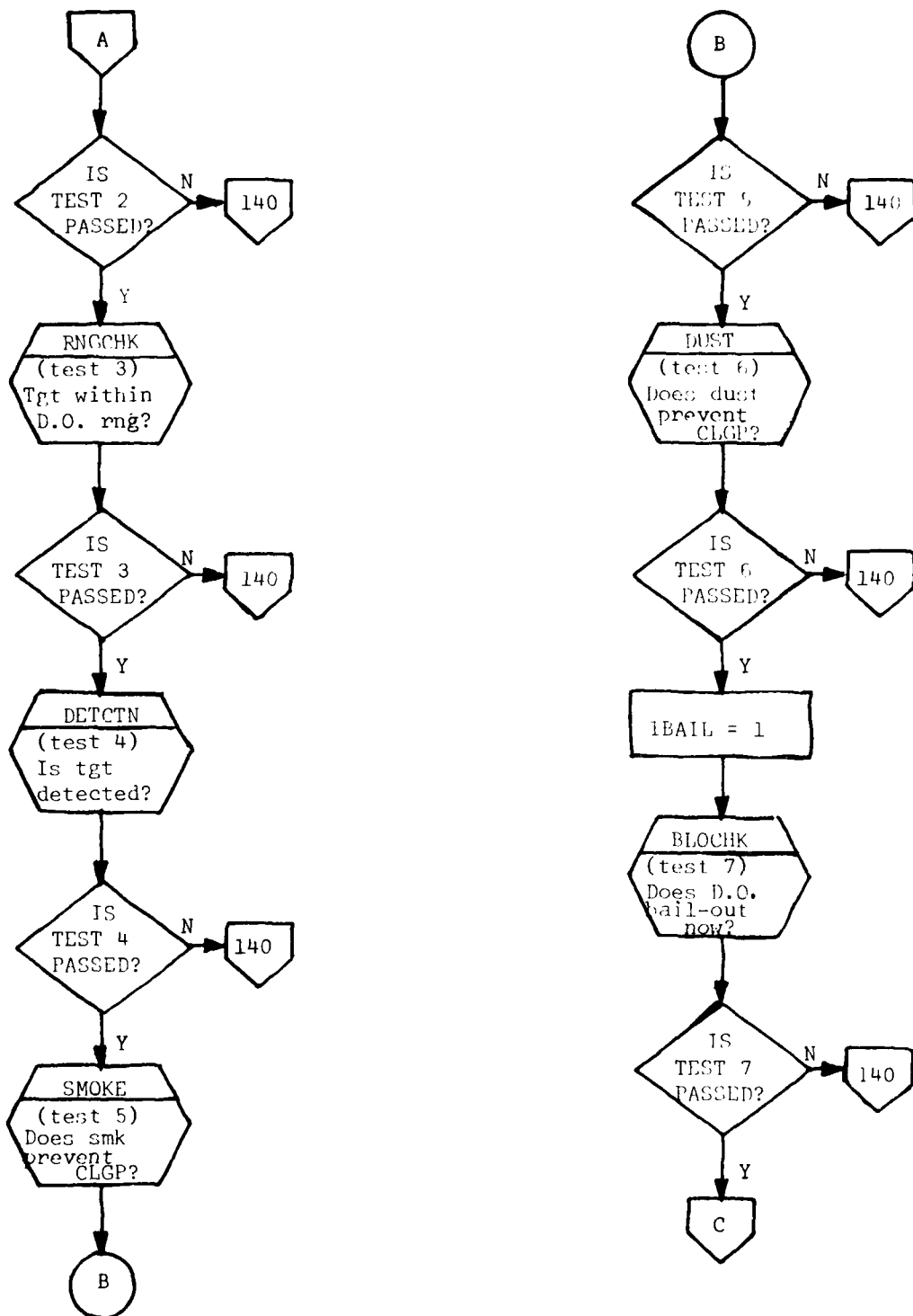


FIGURE 11-1 Flowchart of Main Program of COPE (Page 2 of 5)

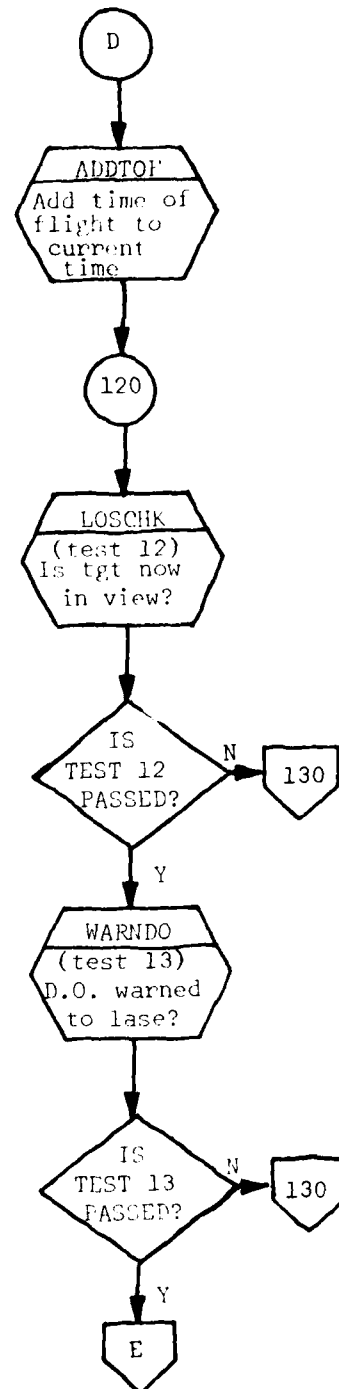
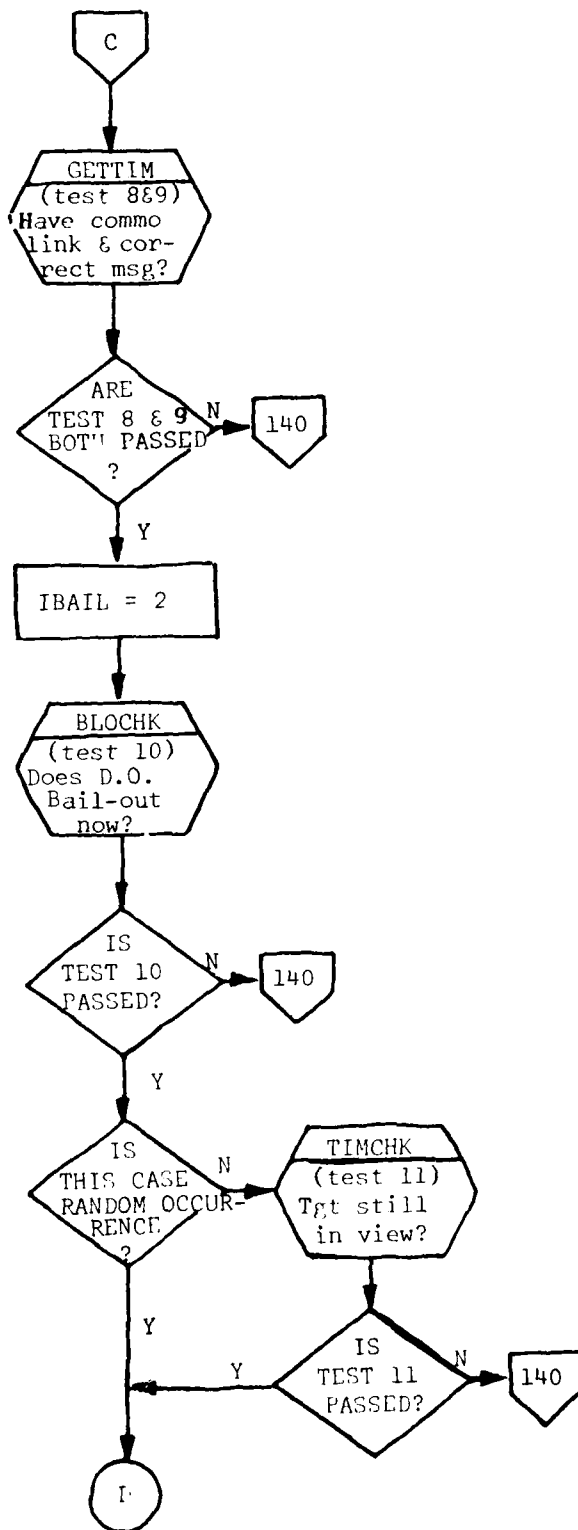


FIGURE 11-1 Flowchart of Main Program of COPE (Page 3 of 5)

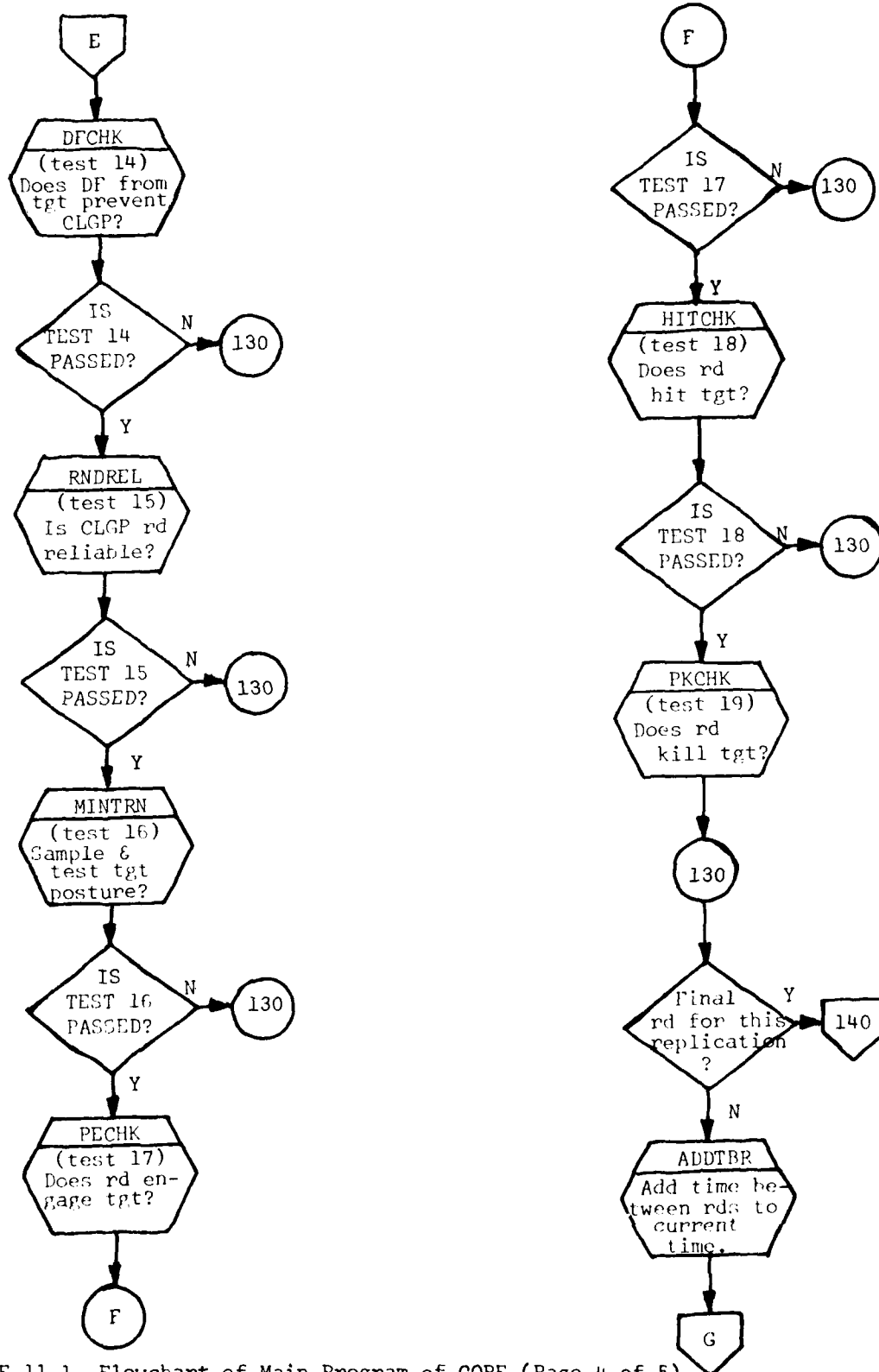


FIGURE 11-1 Flowchart of Main Program of COPE (Page 4 of 5)

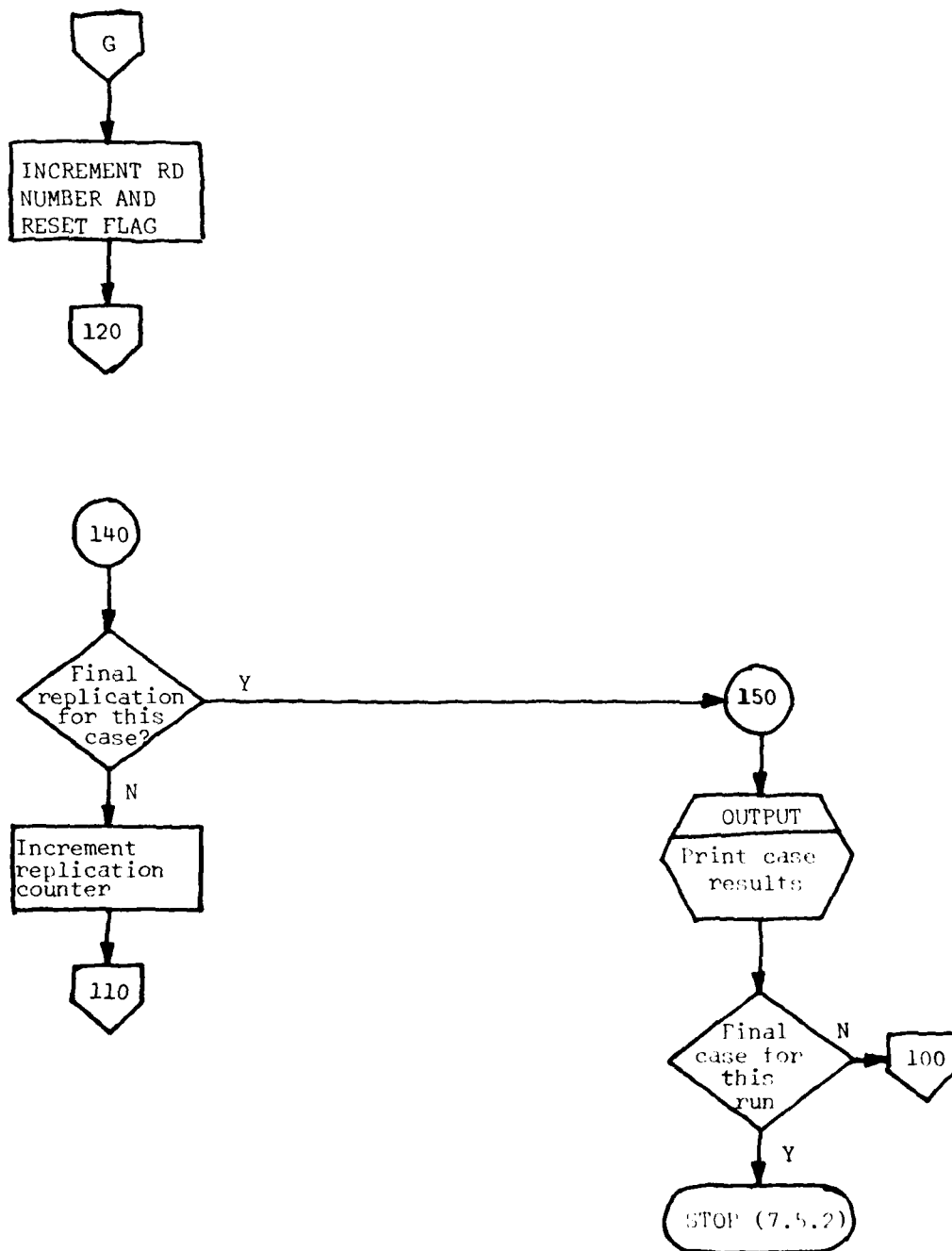


FIGURE 11-1 Flowchart of Main Program of COPE (Page 5 of 5)

Flowchart of Subroutine INPUT of COPE

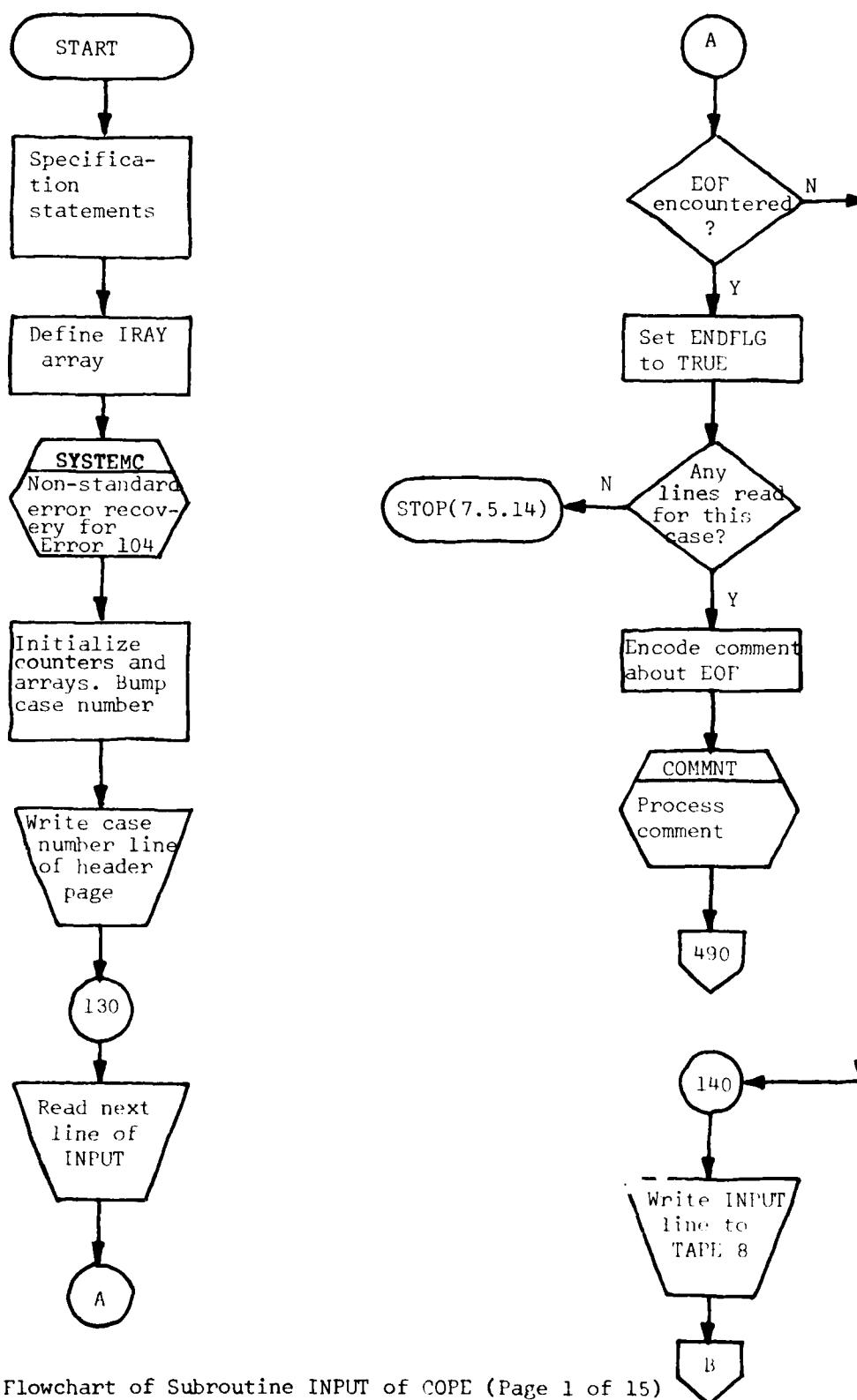


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 1 of 15)

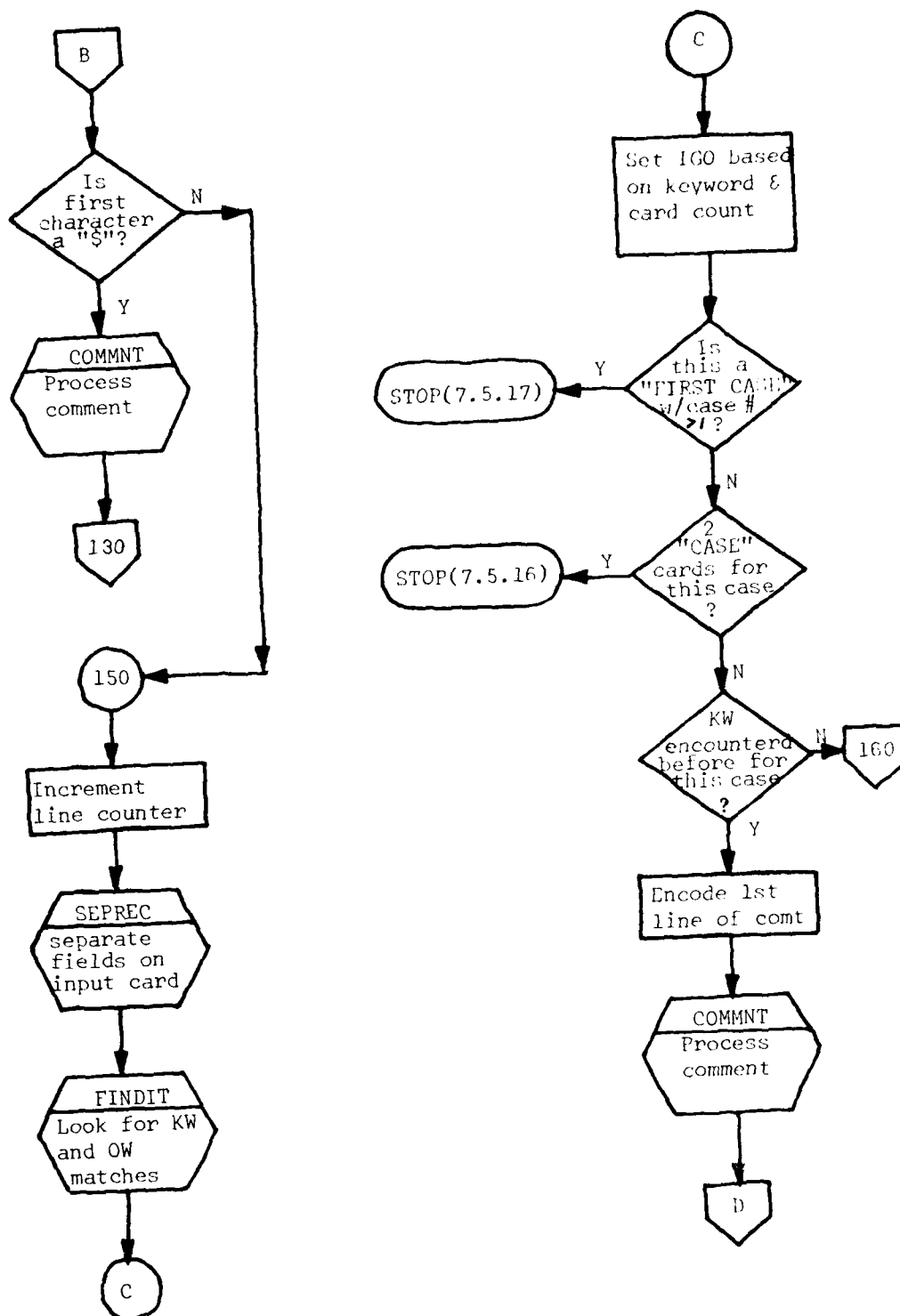


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 2 of 15)

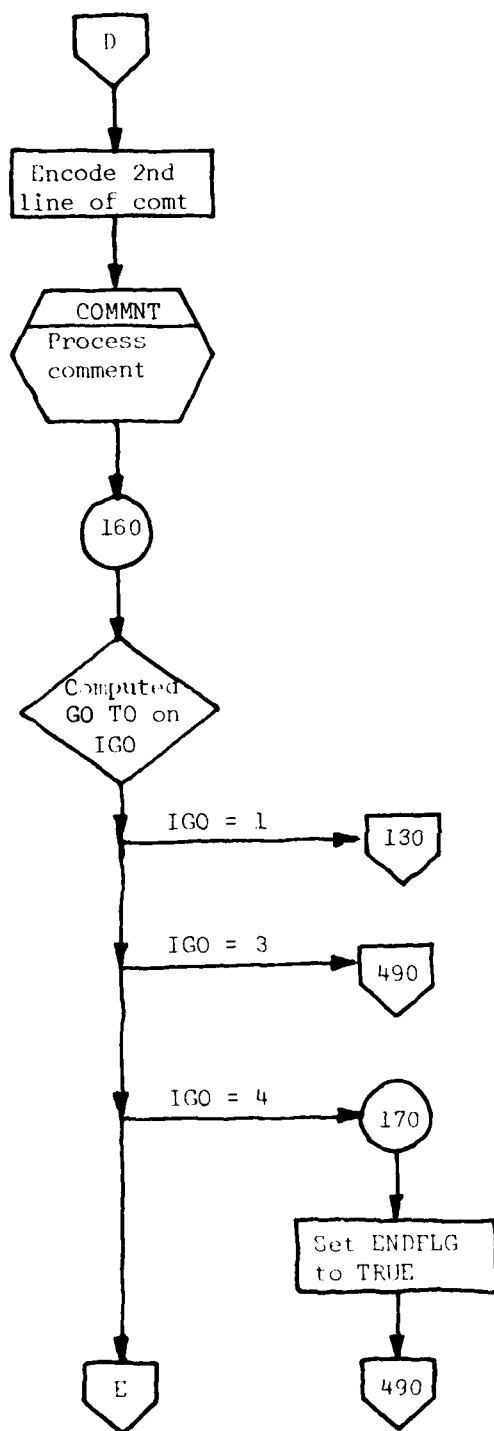


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 3 of 15)

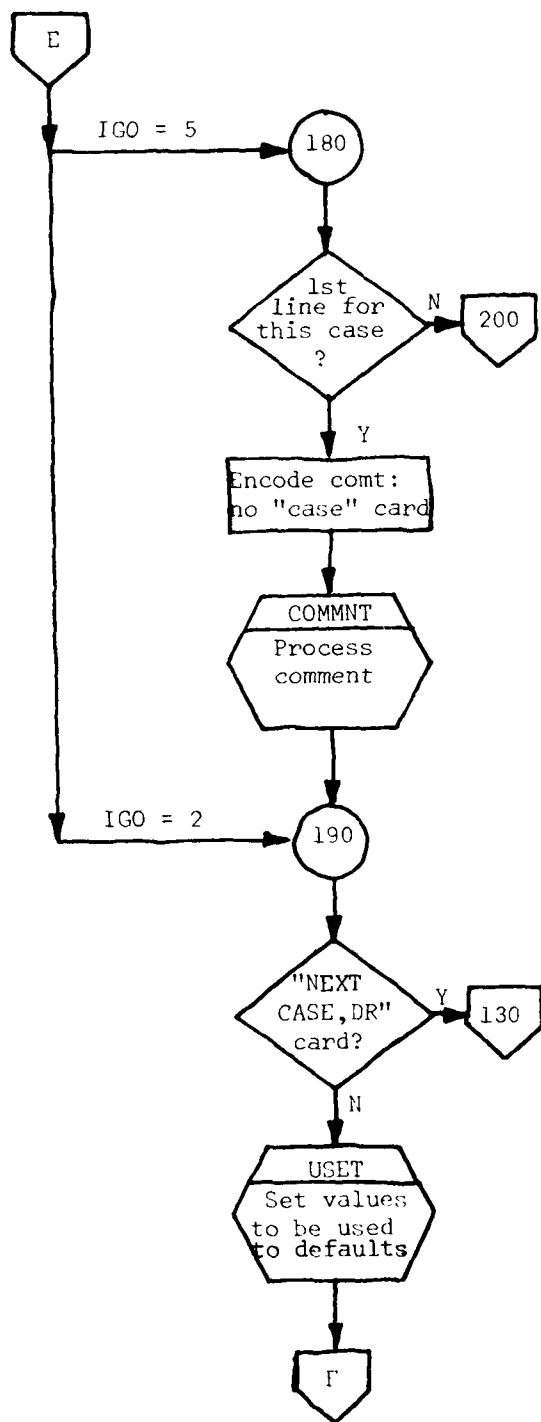


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 4 of 15)

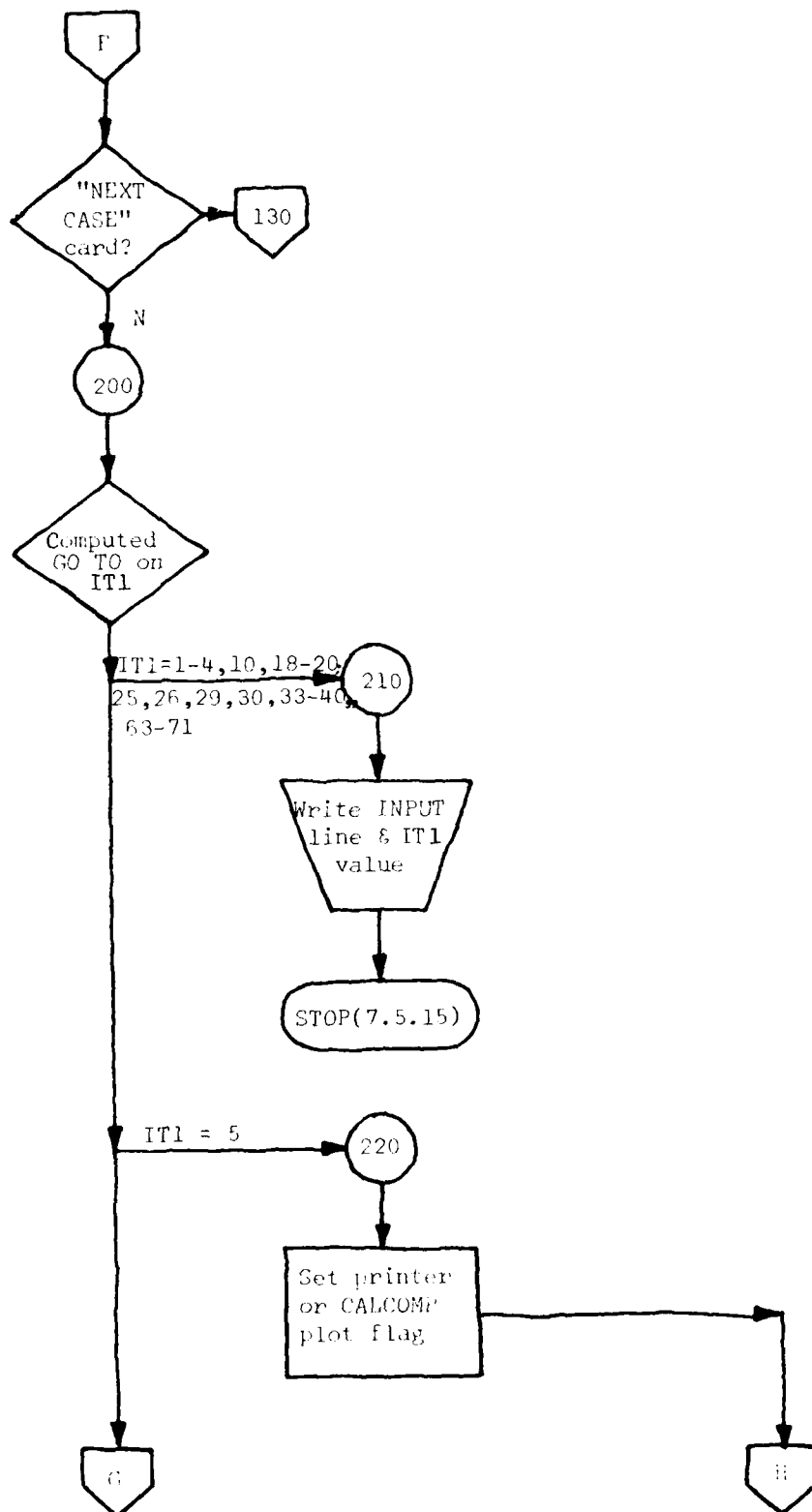


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 5 of 15)

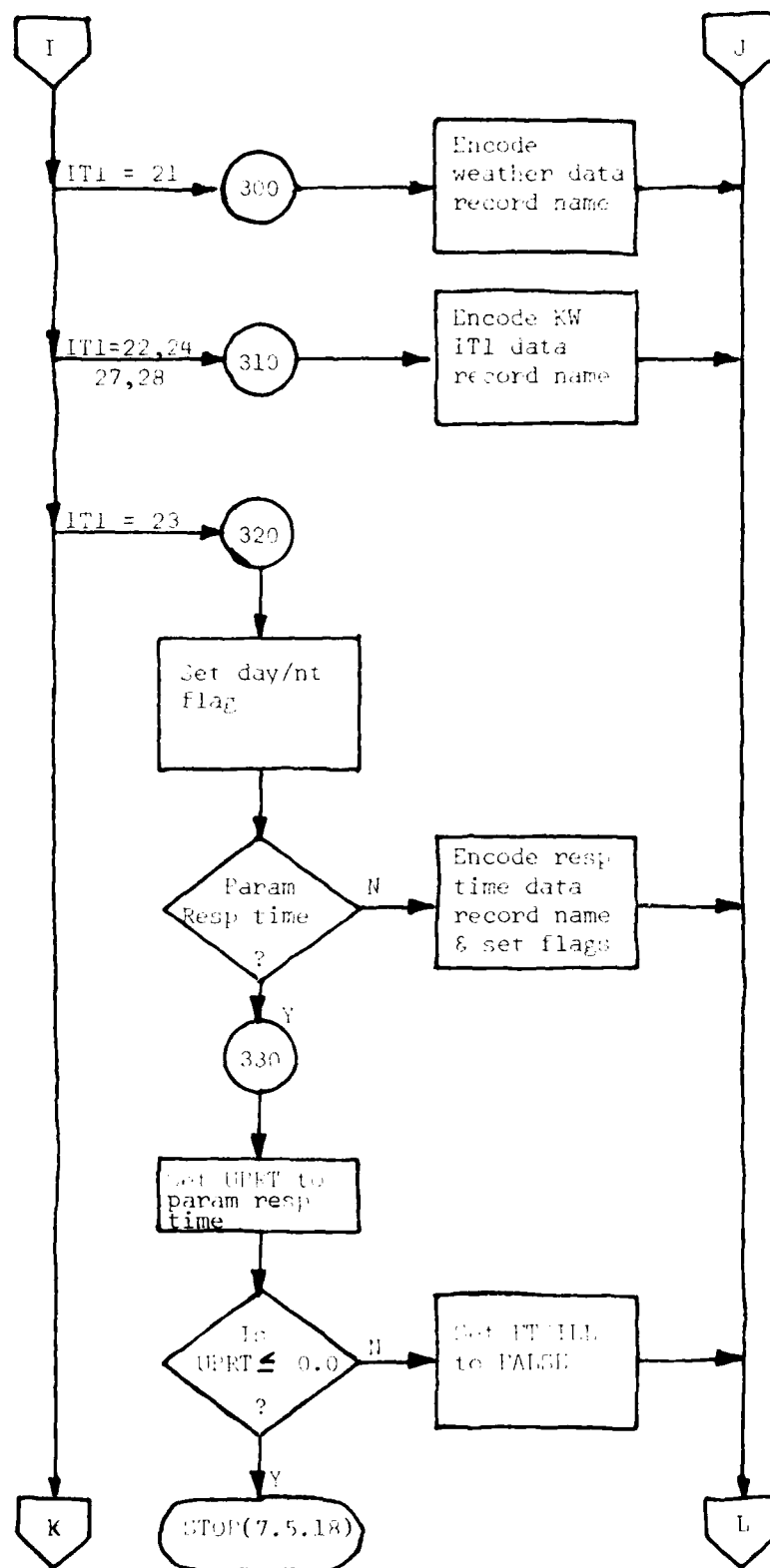


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 7 of 15)

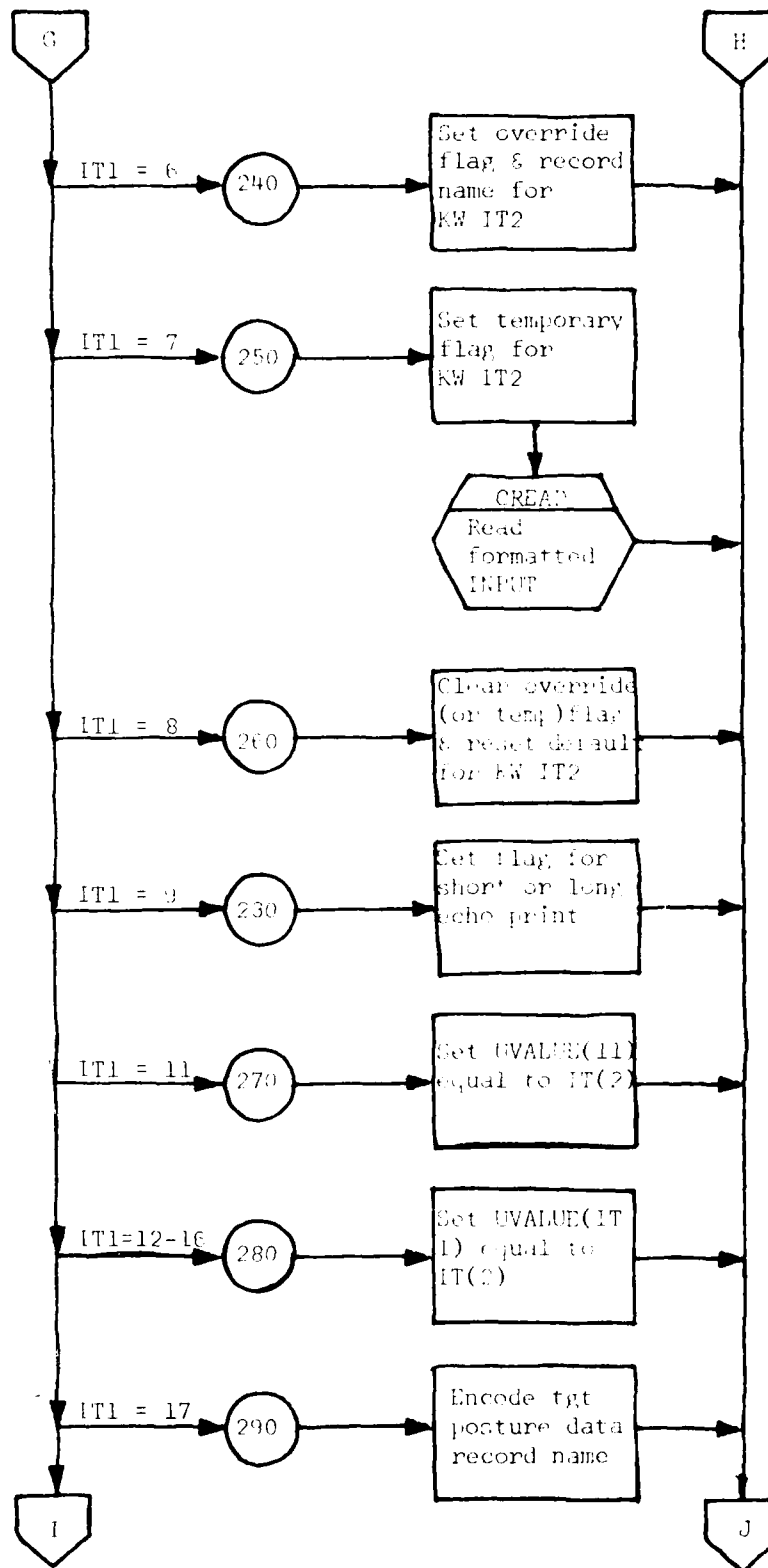


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 6 of 15)

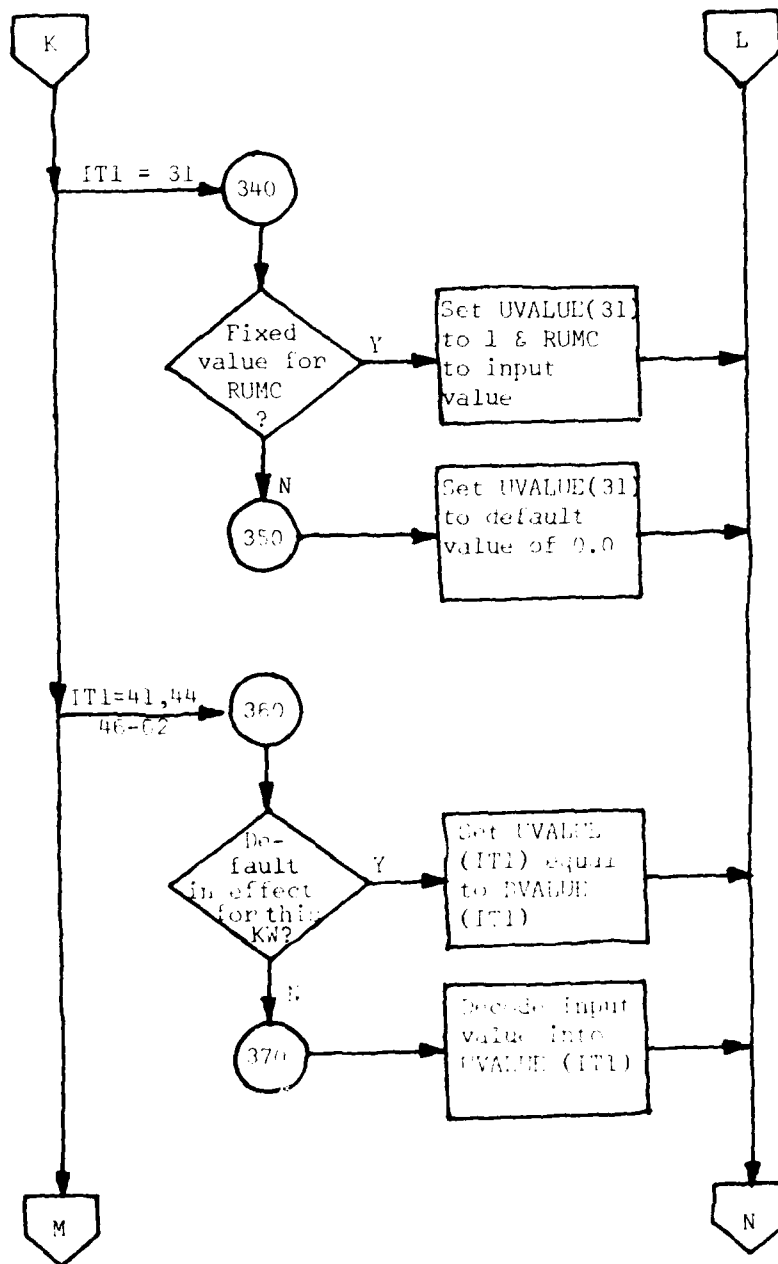


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 8 of 15)

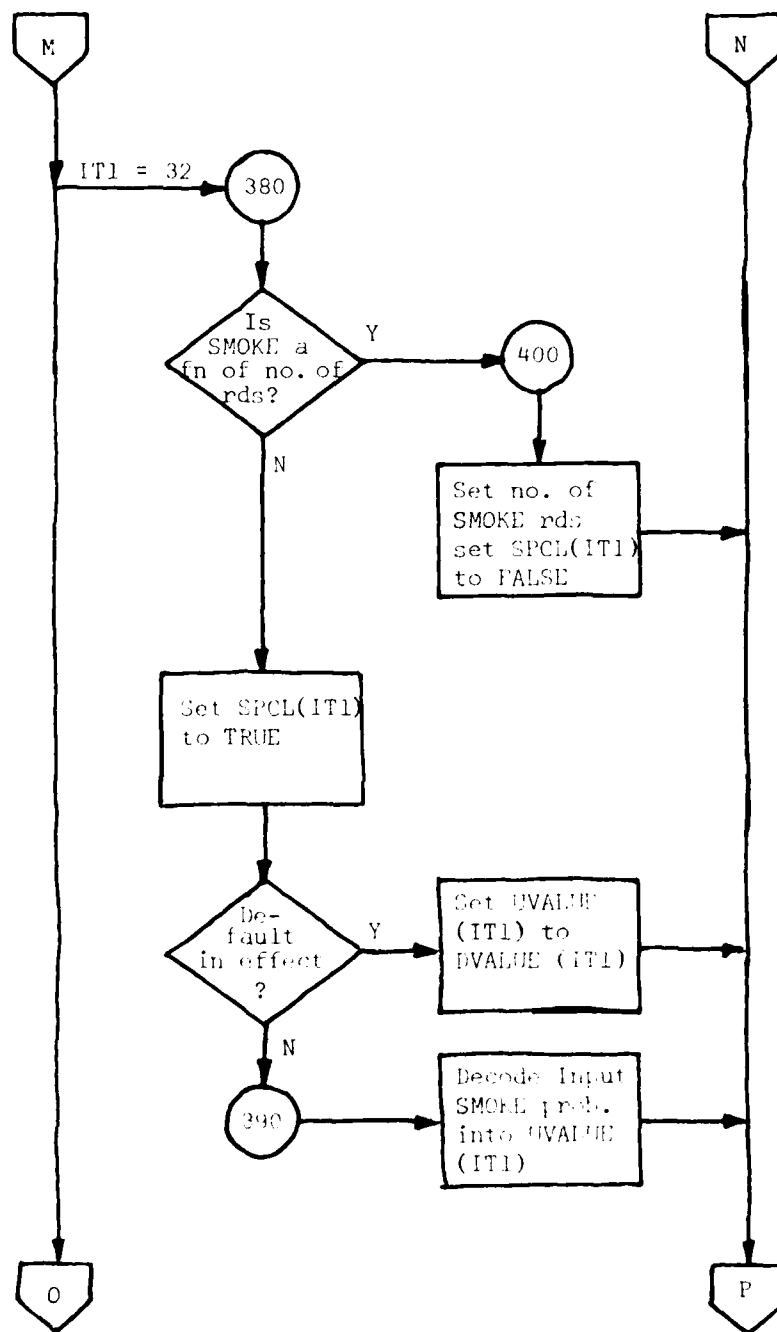


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 9 of 15)

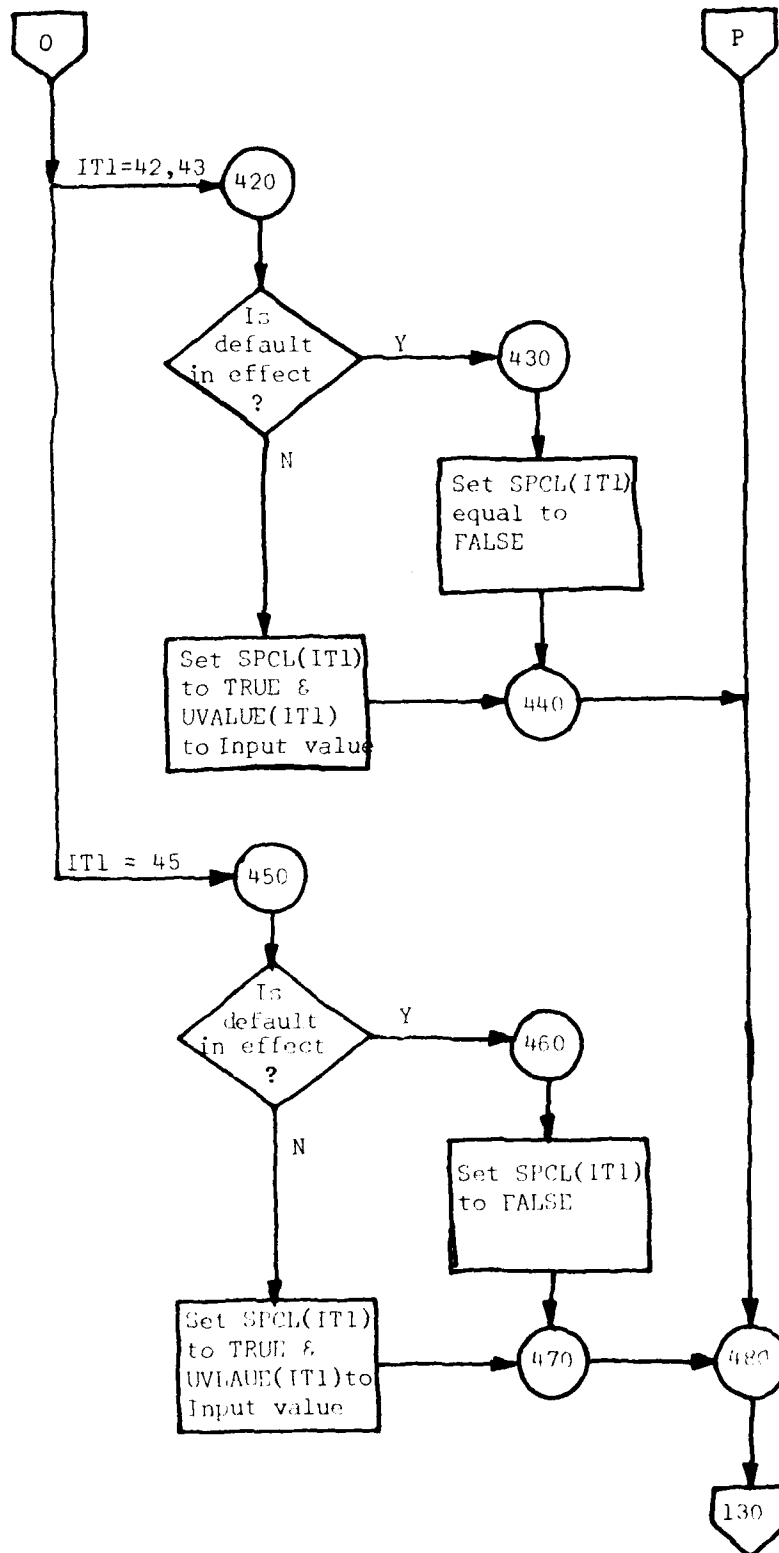


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 10 of 15)

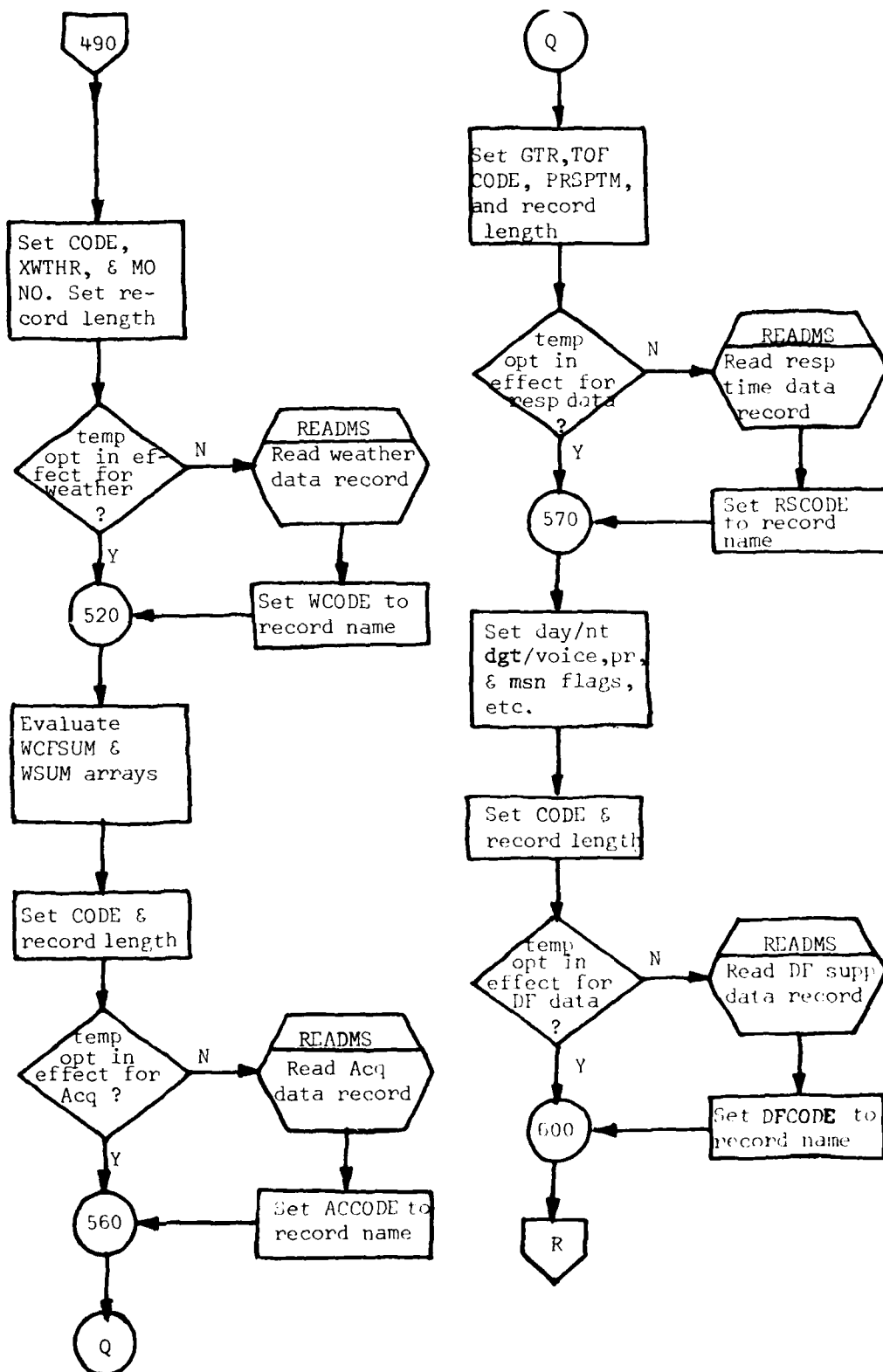


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 11 of 15)

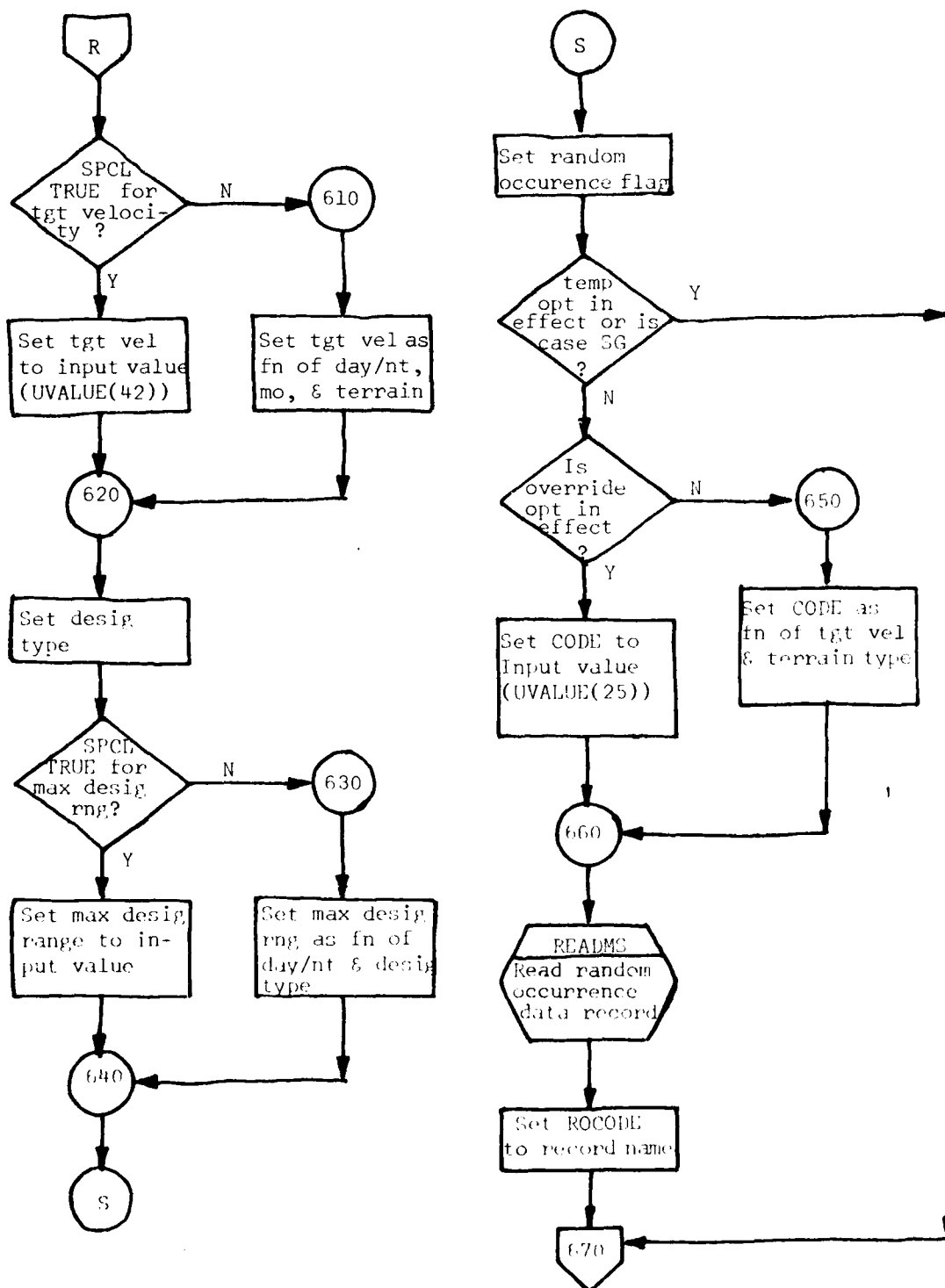


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 12 of 15)

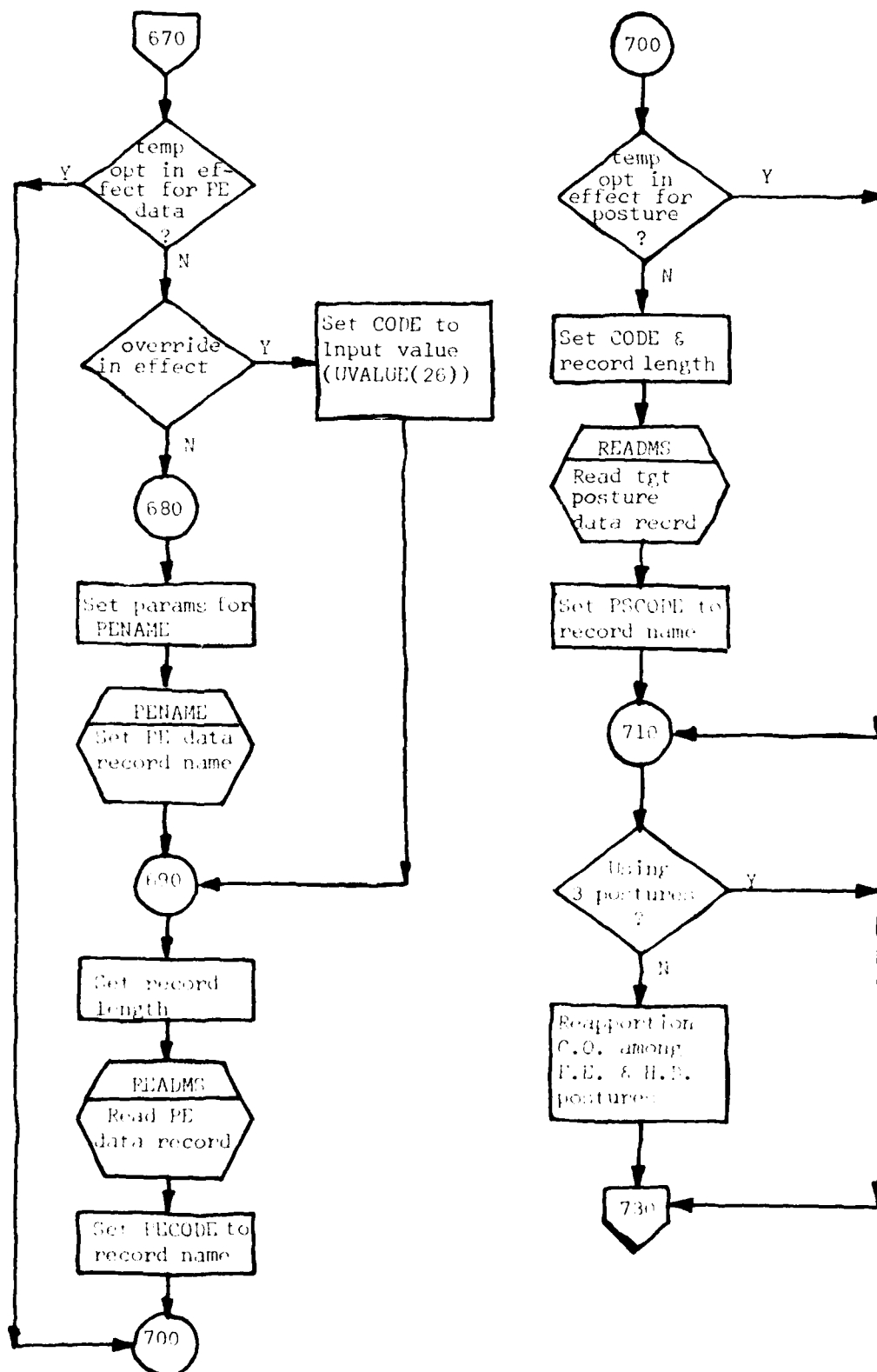


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 13 of 15)

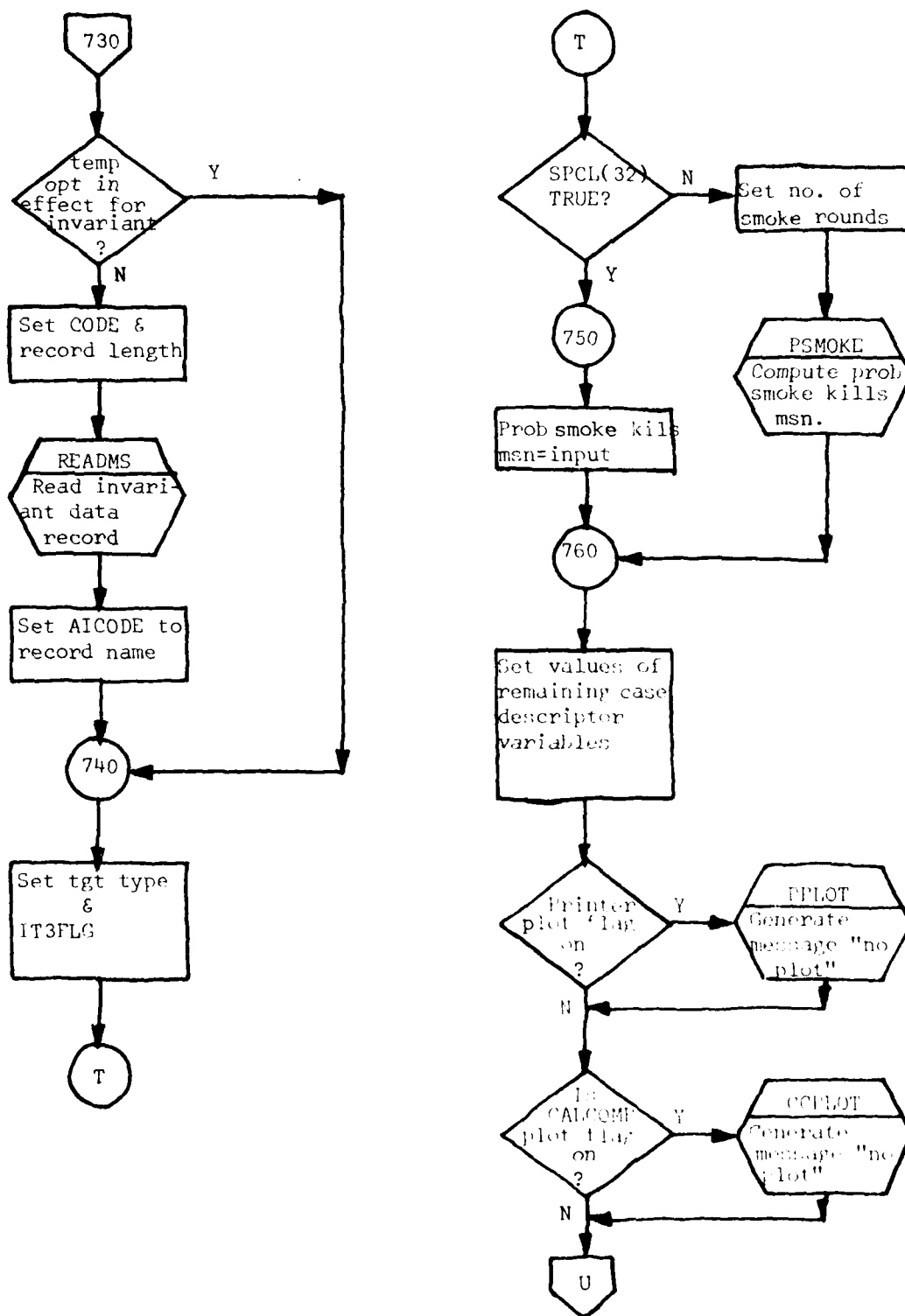


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 14 of 15)

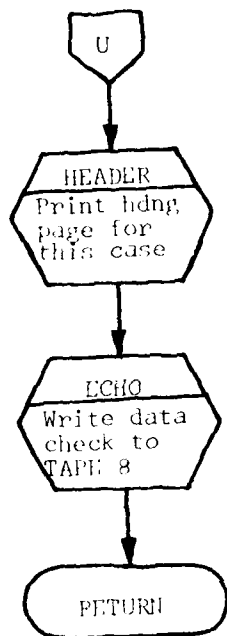


FIGURE 11-2 Flowchart of Subroutine INPUT of COPE (Page 15 of 15)

CHAPTER 12

12. GLOSSARY OF MAIN COPE PROGRAM VARIABLES

This chapter consists of a glossary of variables for the main COPE program. (Glossaries for the PREPMS and PRBLOS preprocessor programs are respectively, included in sections 14.3 and 15.3. No glossary is provided in this report for the PAM preprocessor program.)

The glossary consists of an alphabetical list of the variable names. For each variable name, there is an entry giving VARIABLE, TYPE, COMMON BLOCK, UNITS, and DEFINITION.

Under the VARIABLE heading is the variable name and, if the variable is an array, the array dimensions are given in parentheses after the name.

Under the TYPE column, the variable type is given; if the type is omitted, it means the FORTRAN type default is in effect (i.e., a variable beginning with letter I, J, K, L, M, or N is integer type whereas one beginning with any other letter is single precision real type).

Under the COMMON BLOCK column is the name of the common block (if any) to which the variable belongs. If this column contains "(LOCAL)", it means the variable is a local variable; if it contains "(F.P.)", it means the variable is a formal parameter in a FUNCTION or SUBROUTINE.

Under the UNITS column is the name of the units of measure in which the variable is to be given. If a blank occurs in this column, it means that the variable is unitless.

Finally, under the DEFINITION heading is a brief description of the variable.

GLOSSARY OF MAIN COPE PROGRAM VARIABLES

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
A(30) A(N)		(LOCAL) (F.P.)		ARRAY INTO WHICH INPUT LINE IS READ BY SUBROUTINE INPUT. A(1) IS THE ITH CHARACTER IN THE INPUT LINE. ALSO USED AS A FORMAT PARAMETER TO RECEIVE COMMENTS IN SUBROUTINE COMMENT OR TO PASS INPUT LINE TO SUBROUTINE SEPREC TO UNSTRING KEYWORDS AND OPTION WORDS.
ABRPCT(20, 6)		(LOCAL)		ABRPCT(I, J) IS THE PERCENT OF THE TOTAL NUMBER OF REPLICATIONS THAT ABORTED FOR ITH ABORT REASON ON ROUND J. IF ABRPCT(I, J) IS UNDEFINED, IT IS SET EQUAL TO -1.
ACCODE		RECHAM		RECORD NAME KEY FOR THE ACQUISITION DATA RECORD LAST READ FROM THE WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
ACQDAT(161)		(LOCAL)		ARRAY TO WHICH ELEMENTS OF ACQUISITION RANGE DATA BLOCK ARE EQUIVALENCED.
ACQLBL(2, 4)		HEADHG		ALPHANUMERIC DESCRIPTOR OF THE TERRAIN TYPE USED FOR CURRENT CASE. IT IS USED TO LABEL THE "TERRAIN AND LOS" BLOCK OF THE OUTPUT HEADING.
ALCODE		PECNAM		RECORD NAME KEY FOR THE "INVARIANT" DATA RECORD LAST READ FROM THE WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
AINVDA(167)		(LOCAL)		ARRAY TO WHICH ELEMENTS OF INVARIANT DATA BLOCK ARE EQUIVALENCED.
ALF		(LOCAL)		TEMPORARY NAME FOR THE ALPHA PARAMETER OF THE GAMMA DISTRIBUTION USED TO SAMPLE DIGITAL MESSAGE DEVICE DELAY TIME.
ALFBET(27)		SYMBOL		ALFBET(I) IS THE ITH LETTER OF THE ALPHABET FOR I=1,2,...,26 ALFBET(27) IS AN APOSTROPH. THESE SYMBOLS ARE ALL STORED IN AT FORMAT.
ALPHA		(F.P.)		ALPHA IS THE 'SHAPING FACTOR' OF THE GAMMA DISTRIBUTION.
ALPHA(2, 2)		RSPTM		ALPHA(I, J) IS THE ALPHA PARAMETER TO BE USED WITH THE GAMMA DISTRIBUTION (I=1 FOR DAY, I=2 FOR NIGHT, J=1 FOR A PRE- PLANNED TARGET, AND J=2 FOR A TARGET OF OPPORTUNITY).
ANGLET		PLDESC (F.P.)	DEGREES	THE ANGLE BETWEEN THE GUN-TARGET LINE AND THE DESIGNATOR- TARGET LINE.
ANUMBR(11)		SYMBOL		ANUMBR(1)=1-1 FOR 1<10. ANUMBR(11) IS A MINUS SIGN. THESE ARE STORED IN AT FORMAT.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
AOUT(4, 40)		(LOCAL)		ARRAY INTO WHICH THE FIRST COLUMN OF CASE HEADING OUTPUT BLOCKS ARE ENCODED PRIOR TO BEING PRINTED OUT IN SUBROUTINE HEADER.
AVALUE(9)		(LOCAL)		ARRAY USED TO TEMPORARILY STORE VALUES USED IN CREATING AND DECIPHERING PROBABILITY OF ENGAGEMENT RECORD NAME KEYS USED TO READ TAPE 11.
A1		(LOCAL)		USED IN FUNCTION URAN31 TO COMPUTE PSEUDO-RANDOM NUMBER.
B(8, 10) B(M, N)		(LOCAL) (F.P.)		B(I, J) CONTAINS CHARACTERS 10*(I-1)+1 THROUGH 10*I OF THE JTH KEYWORD OR OPTION WORD OF THE MOST RECENTLY READ NON-COMMENT INPUT LINE STORED IN A10 FORMAT WITH MOST BLANKS REMOVED.
BATRTM(3)		RSPTM	SECONDS	BATRTM(1) IS THE TIME USED BY THE FIRING BATTERY FROM THE TIME IT RECEIVES A COPPERHEAD FIRE MISSION ORDER FROM THE FDC UNTIL IT IS READY TO FIRE THE FIRST COPPERHEAD ROUND OF THAT MISSION. I=1 FOR PRE-PLANNED TARGETS, I=2 FOR TARGETS OF OPPORTUNITY, I=3 FOR PRIORITY PRE-PLANNED TARGETS.
BCSPTM(2, 2)		RSPTM	SECONDS	DCSPTM(I, J) IS THE TIME REQUIRED FOR THE BATTERY COMPUTER SYSTEM TO PROCESS THE COPPERHEAD FIRE MISSION REQUEST. I=1 FOR DIGITAL COMMUNICATIONS, I=2 FOR VOICE COMMUNICATIONS; J=1 FOR PRE-PLANNED TARGETS, J=2 FOR TARGETS OF OPPORTUNITY.
BETA(2, 2)		RSPTM	SECONDS	BETA(I, J) IS THE SCALING FACTOR TO BE USED WITH THE GAMMA DISTRIBUTION (I=1 FOR DAY, I=2 FOR NIGHT; J=1 FOR PRE-PLANNED TARGETS, J=2 FOR TARGETS OF OPPORTUNITY).
BLANK		SYMBOL		BLANK IS A BLANK SYMBOL (ONE EMPTY PRINT SPACE) STORED IN A1 FORMAT.
BLORNG		BAIL	METERS	BLORNG IS THE DESIGNATOR BAIL-OUT RANGE.
BOUT(4, 40)		(LOCAL)		ARRAY INTO WHICH SECOND COLUMN OF CASE HEADING OUTPUT BLOCKS ARE ENCODED PRIOR TO BEING PRINTED OUT IN SUBROUTINE HEADER.
C(80)		(LOCAL)		C(I) IS THE ITH CHARACTER OF AN OPTION WORD THAT IS BEING CHECKED TO DETERMINE WHETHER IT IS NUMERIC. (STORED IN A1 FORMAT.)

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
CHAR(10, 120)		ACCHAR		CHAR(1, J) FOR I>2 CONTAINS CHARACTERS 10*(I-3)+1 THROUGH 10*(I-2) OF THE JTH KEYWORD CHARACTER STRING IN AIO FORMAT. CHAR(1, J) IS THE KEYWORD NUMBER AND CHAR(2, J) IS THE NUMBER OF OPTION WORD CHARACTER STRINGS ALLOWED WITH JTH KEYWORD.
CHARA(10, 30)		(LOCAL)		CHAR(1, J) FOR I>2 CONTAINS CHARACTERS 10*(I-3)+1 THROUGH 10*(I-2) OF THE JTH OPTION WORD CHARACTER STRING IN AIO FORMAT. CHARA(1, J) IS THE OPTION WORD CHOICE NUMBER AND CHARA(2, J) IS THE NUMBER INDICATING THE LEVEL (POSITION NUMBER) OF THE OPTION WORD.
CLOUDS		(LOCAL)	METERS	CLOUD CEILING ALTITUDE. (NOT ACTUALLY USED AFTER IT IS SET). THE INDEX OF THE CLOUD CEILING ALTITUDE (ICC) IS THE NUMBER USED TO ACCESS DATA THAT DEPENDS ON CLOUD CEILING ALTITUDE.)
CLOUDS(6)		WEATHR	METERS	ARRAY OF CLOUD CEILING ALTITUDE VALUES.
CM(10)		(LOCAL)		CONTAINS COMMENT LINE GENERATED BY PROGRAM AND READY TO BE PROCESSED BY SUBROUTINE COMMENT.
CMNT(13, 20)		COMMENT		CMNT(1, J) CONTAINS CHARACTERS 10*(I-1)+1 THROUGH 10*I OF THE JTH LINE OF COMMENTS TO BE PRINTED WITH THE CURRENT CASE
CODE		(LOCAL)		CODE AND ICODE ARE EQUIVALENT. (SEE ICODE)
COU(4, 40)		(LOCAL)		ARRAY INTO WHICH THIRD COLUMN OF CASE HEADING OUTPUT BLOCKS ARE ENCODED PRIOR TO BEING PRINTED OUT IN SUBROUTINE HEADER.
COMBUF(13)		CONTENT		COMMENT BUFFER. IF FOR ANY CASE THE NUMBER OF \$\$ COMMENTS EXCEEDS THE SECOND DIMENSION OF THE CMNT ARRAY (ORIGINALLY SET AT 20), THEN THE COMMENT THAT CAUSES THE OVERFLOW IS STORED IN COMBUF AND AN ERROR PRINT OCCURS FOLLOWED BY A STOP WITH MESSAGE.
CRNG(11, 2)		PRGLOS	NONE, METERS	CUMULATIVE ACQUISITION RANGE DISTRIBUTION. CRNG(1, 1) IS THE PROBABILITY THAT THE ACQUISITION RANGE IS LESS THAN OR EQUAL TO CRNG(1, 2). (CRNG(1,1) IS UNITLESS; CRNG(1, 2) IS IN METERS).
CR7(70)		DVALUE		DEFAULT VALUES FOR PARAMETERS ASSOCIATED WITH KEYWORDS.
CRAT		RUNDAT		DATE OF RUN FOR CURRENT COMPUTER JOB. (OBTAINED FROM COMPUTER'S INTERNAL CALENDAR.)

AD-A100 285

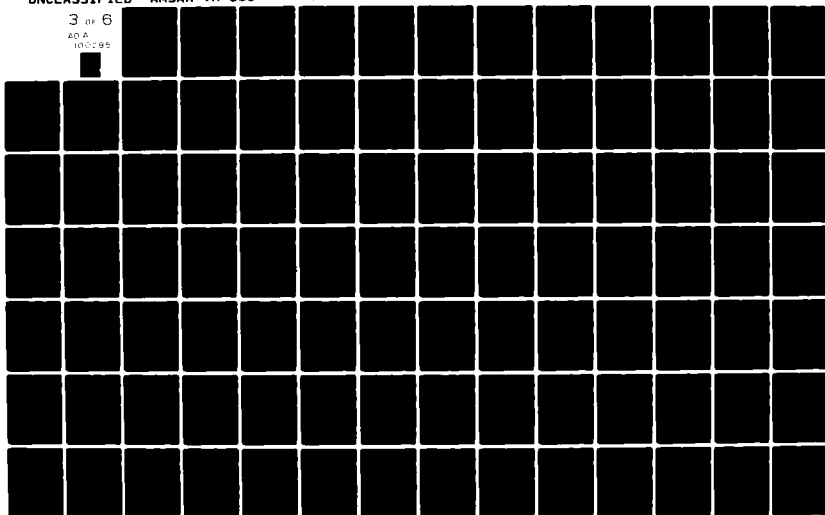
ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY ABERDEEN PROV--ETC F/6 19/1
COPPERHEAD OPERATIONAL PERFORMANCE EVALUATION (COPE): COMPUTER --ETC(U)
MAR 81 R S SANDMEYER
AMSAA-TR-318

NL

UNCLASSIFIED

3 OF 6

40 A
100:95



GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
DAYLBL (2)		HEADING		ALPHANUMERIC LABELS USED IN LABELING OUTPUT HEADINGS AS DAY OR NIGHT.
DDT		DVALUE		NOT USED.
DECFLG		(LOCAL)		IF DECFLG=1.0, A DECIMAL POINT HAS BEEN ENCOUNTERED IN C ARRAY; IF DECFLG=0.0, NO DECIMAL POINT HAS BEEN ENCOUNTERED IN C ARRAY.
DECPNT		SYMBOL		ALPHANUMERIC SYMBOL FOR DECIMAL POINT STORED IN A1 FORMAT.
DEFB		PEDESC(LOCAL) IN PENAME)	METERS	DEFLECTION BIAS (DISTANCE FROM FOOTPRINT CENTROID TO POINT OF CLOSEST APPROACH OF TARGET).
DELTAT		(LOCAL)	SECONDS	TIME ROUND ARRIVES ON TARGET MINUS TIME LAST TARGET VEHICLE IN COLUMN PASSES POINT OF CLOSEST APPROACH TO FOOTPRINT CENTROID. IF DELTAT AS COMPUTED ABOVE IS NEGATIVE, THEN A VALUE OF ZERO IS USED.
DESRNG (3, 2)		DESRNG	METERS	DESRNG(I,J) IS THE MAXIMUM DESIGNATOR RANGE FOR THE ITH DESIGNATOR TYPE AND THE JTH DAY OR NIGHT CONDITION (J=1 FOR DAY, J=2 FOR NIGHT).
DETTM		RSPTIM	SECONDS	TIME FROM TARGET UNMASK TO OBSERVER'S CALL FOR COPPERHEAD.
DETTMA (10, 2)		RSPTIM	NONE, SEC	DETECTION TIME CUMULATIVE DISTRIBUTION ARRAY. DETTMA(1,1) IS THE PROBABILITY THAT DETECTION TIME IS LESS THAN OR EQUAL TO DETTMA(1,2).
DFCODE		RECNAME		RECORD NAME KEY FOR THE DIRECT FIRE DATA RECORD LAST READ FROM THE WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
DFDOKL (10, 3)		DODF	METERS, NONE	DFDOKL(1,1) IS THE 1TH RANGE CLASS UPPER LIMIT: DFFOKL(1,2) IS THE PROBABILITY THAT DIRECT FIRE OBSCURES DESIGNATOR'S VIEW OF TARGET AT RANGE DFDOKL(1,1) BUT DOES NOT KILL DESIGNATOR: DFDOKL(1,3) IS THE PROBABILITY THAT DIRECT FIRE KILLS THE DESIGNATOR AT RANGE DFDOKL(1,1).
DFS0AT		(LOCAL)	METERS	ARRAY TO WHICH ELEMENTS OF DIRECT FIRE SUPPRESSION DATA CLOCK ARE EQUIVALENCED.
DFSLBL (2, 2)		HEADING		ARRAY CONTAINING LABELS FOR DIRECT FIRE SUPPRESSION LEVEL PRINT OUT.
DGTLBL (2, 2)		HEADING		ARRAY CONTAINING LABELS FOR COMMUNICATIONS PRINT OUT.

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
DIDIGT		DVALUE		DEFAULT COMMUNICATIONS TYPE (DIGITAL).
DIDN		DVALUE		DEFAULT DAY OR NIGHT TIME (DAY).
DISBVH		TARGET	METERS	MEAN DISTANCE BETWEEN CONSECUTIVE VEHICLES IN TARGET COLUMN.
DISPM(12)		DISPLY		DISPM(1) IS THE FIRST THREE LETTERS OF THE ITH MONTH NAME USED IN PROGRAM. (A3 FORMAT)
DISPT(24)		DISPLY		DISPT(1) IS THE FOUR DIGIT TIME OF DAY FOR THE ITH TIME OF DAY PLAYED. (A4 FORMAT)
DLTT(6)		HIT	SECONDS	ARRAY OF DELAY TIME VALUES USED FOR INTERPOLATING IN PROBABILITY OF ENGAGEMENT ARRAY.
DOARPK		MISC		PROBABILITY DESIGNATOR HAS BEEN KILLED BY PREPARATORY ARTILLERY FIRES.
DOKILD	LOGICAL	DODF		FLAG USED TO INDICATE THAT DESIGNATOR HAS BEEN KILLED BY DIRECT FIRE. (TRUE = KILLED; FALSE = NOT KILLED)
DOLBL(3,2)		HEADING		ARRAY OF DESIGNATOR LOCATION LABELS USED FOR PRINT OUT.
DOLLAR		SYMBOL		DOLLAR SIGN SYMBOL IN A1 FORMAT.
DPRT		DVALUE	SECONDS	DEFAULT VALUE FOR PARAMETERIZED RESPONSE TIME.
DSMK(2)		DVALUE		DEFAULT NUMBER OF SMOKE ROUNDS FIRED BY RED. DSMK(1) FOR TYPE 1 SMOKE ROUNDS; DSMK(2) FOR TYPE 2 SMOKE ROUNDS.
DTRNG		RANGE	METERS	DESIGNATOR-TO-TARGET RANGE.
DUMT		RSPTIM		DUMMY VARIABLE NO LONGER USED (BUT RETAINED TO AVOID CHANGING COMMON BLOCK SIZE).
DURLOS		TIME	SECONDS	DURATION OF LINE-OF-SIGHT (TIME FROM WHEN FIRST VEHICLE IN COLUMN UNMASKS UNTIL LAST VEHICLE IN COLUMN LEAVES LINE-OF-SIGHT SEGMENT).
DVALUE(70)		(LOCAL)		DVALUE(1) IS EQUIVALENT TO D(1).
EDDTIM		(LOCAL)	SECONDS	TIME TO ENTER FIRE REQUEST ON DIGITAL MESSAGE DEVICE.
ENDFLG	LOGICAL	FLAG		FLAG INDICATING NO FURTHER CASES TO BE READ.
ENDREP	LOGICAL	FLAG		FLAG INDICATING REPLICATION (OR FIRING OF ROUND) IS TO BE ENDED.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
FDT		(LOCAL)		USED TO CALCULATE FDT2.
FDT1 FDT2		(LOCAL)		USED FOR INTERPOLATING WITH RESPECT TO DELAY TIME IN THE PROBABILITY OF ENGAGEMENT TABLES.
FIRSTL	LOGICAL	LOGFLG		FLAG USED TO INDICATE FIRST INPUT LINE IN CHECKING FOR SEQUENCE NUMBERS.
FPRCNT		FUNCTION		FUNCTION VALUE, SPECIAL FORMATTING OF PERCENTAGES USED FOR PRINT OUT.
FRAC		(LOCAL)		USED TO INTERPOLATE WITH RESPECT TO RANGE IN THE UFDOKL ARRAY.
FRONT		SMOKED	KM	WIDTH OF FRONT WHICH RED WANTS TO SMOKE.
FSRT(14,2)		FSRESP	NONE, SECONDS	FT SILL'S CUMULATIVE RESPONSE TIME DISTRIBUTION. FSRT(1,1) IS THE PROBABILITY THAT THE TOTAL DESIGNATOR-FDC-BATTERY RESPONSE TIME IS LESS THAN OR EQUAL TO FSRT(1,2).
FSRTIM		(LOCAL)	SECONDS	TOTAL DESIGNATOR-FDC-BATTERY RESPONSE TIME AS SAMPLED FROM FSRT ARRAY WHEN USING FT SILL RESPONSE TIME DISTRIBUTION.
FTSILL	LOGICAL	LOGFLG		FLAG TO INDICATE WHETHER FT SILL RESPONSE TIME DISTRIBUTION IS BEING USED.
GAMMA		FUNCTION		FUNCTION VALUE. GAMMA DISTRIBUTED RANDOM VARIABLE.
GTR		(F.P.)	KM	GUN-TO-TARGET RANGE AS USED IN SUBROUTINE PENNAME.
GTRNG		PEDESC	KM	GUN-TO-TARGET RANGE.
HUMID(2)		WEATHR		HUMID(1) IS THE PROBABILITY THAT THE RELATIVE HUMIDITY FALLS IN CLASS 1. (IF 1=1, RELATIVE HUMIDITY <65%; IF 1=2, RELATIVE HUMIDITY >65%).
I		(LOCAL) (F.P.)		USED AS CONTROL VARIABLE IN VARIOUS DO-LOOPS AND IMPLIED DO-LOOPS. ALSO USED AS A FORMAL PARAMETER (RANDOM NUMBER SEED AND SUBSCRIPT) AND A COUNTER.
IAGRT		(LOCAL) (F.P.)		INDEX NUMBER OF CAUSE OF MISSION (OR ROUND) ABORT.
IACQL		(LOCAL)		LENGTH OF ACQUISITION DATA RECORD ON WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
ICENG		(LOCAL)		NUMBER OF ATTEMPTED ENGAGEMENTS FOR THE CURRENT CASE.
IB		(LOCAL & F.P.)		USED AS SECOND SUBSCRIPT OF B ARRAY IN NUMRIC AND FINUIT.
IBAIL		BAIL		FLAG INDICATING WHICH BAIL-OUT CHECK IS BEING MADE. IBAIL=1 INDICATES PRE-COMMUNICATIONS CHECK; IBAIL=2 INDICATES POST-COMMUNICATIONS CHECK.
IC		(LOCAL)		CONTROL VARIABLE FOR IMPLIED DO-LOOP IN DECODE STATEMENT. ALSO AN UNUSED RETURNED PARAMETER FROM A CALL TO SMPLOC.
ICC		WEATHR		ICC INDICATES WHICH OF THE SIX CLOUD CEILING ALTITUDES IS BEING USED ON THIS REPLICATION. IF CLOUD FREE LINE-OF-SIGHT EXISTS, ICC=1 (HIGHEST CEILING) IS USED.
ICCPFG		RUNDAT		CALCOMP PLOT FLAG. ICCPFG=0 INDICATES NO CALCOMP PLOT HAS BEEN REQUESTED FOR THE CURRENT CASE; ICCPFG=1 INDICATES THAT CALCOMP PLOT OUTPUT HAS BEEN REQUESTED.
ICFLOS		(LOCAL)		CLOUD FREE LINE-OF-SIGHT FLAG. ICFLOS=1 INDICATES A CLOUD FREE LINE-OF-SIGHT; ICFLOS=2 INDICATES NO CLOUD FREE LINE-OF-SIGHT.
ICHNL		(LOCAL)		NUMBER OF OPTION WORDS IN CHARA ARRAY TO BE CHECKED IN SEARCH FOR MATCH WITH OPTION WORDS OF CURRENT KEYWORD.
ICHRLL		(LOCAL)		NUMBER OF WORDS IN OPTION WORD DATA RECORD ON WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
ICLASS		(LOCAL) (F.P.)		WHEN A CUMULATIVE DISTRIBUTION IS SAMPLED USING THE SMPLOC SUBROUTINE, THE VALUE RETURNED IN ICLASS IS THE NUMBER OF THE INTERPOLATION INTERVAL FROM WHICH THE SAMPLE WAS OBTAINED.
ICLNGF		(LOCAL)		CLOUD CEILING FLAG. ICLNGF=1 INDICATES A CLOUD CEILING; ICLNGF=2 INDICATES NO CLOUD CEILING (I.E., SCATTERED CLOUDS)
ICMFLG		(LOCAL)		COMMENT FLAG. ICMFLG=0 INDICATES NO TIME OF DAY/WEATHER INCONSISTENCY COMMENT WILL BE PRINTED FOR THIS CASE; ICMFLG=1 INDICATES SUCH A COMMENT WILL BE PRINTED.
ICMNT		COMENT		NUMBER OF \$\$ COMMENT LINES TO BE PRINTED FOR THIS CASE.
ICODE		ICODE		RECORD NAME KEY WHEN READING FROM WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11). ICODE IS EQUIVALENT TO CODE.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
ICRD				INDEX NUMBER USED TO CHECK WHETHER MORE THAN ONE INPUT LINE HAS BEEN READ WITH THE SAME KEYWORD FOR EACH CASE.
IDABRT(3,19)		ABRLBL		ALPHANUMERIC DESCRIPTORS OF CAUSES OF MISSION (OR ROUND) ABORTS. IDABRT(I,J) CONTAINS CHARACTERS 10*(I-1)+1 THROUGH 10*I OF ABORT DESCRIPTOR NUMBER J.
IDCHAR		FUNCTION		FUNCTION VALUE. A NUMERICAL VALUE USED TO INDICATE TO WHICH CLASS OF SYMBOLS A GIVEN CHARACTER BELONGS.
IDCHR		(LOCAL)		A NUMERICAL VALUE INDICATING TO WHICH CLASS OF SYMBOLS A PARTICULAR CHARACTER BELONGS.
IDCODE		(F.P.)		RECORD NAME KEY FOR A PROBABILITY OF ENGAGEMENT DATA RECORD TO BE READ OR INTERPRETED.
IDELTT		(LOCAL)		SUBSCRIPT OF INDEX ARRAY USED WHEN DELTAT IS ZERO.
IDELT1		(LOCAL)		SUBSCRIPT OF INDEX ARRAY USED FOR INTERPOLATION WHEN DELTAT IS POSITIVE.
IDELT2		(LOCAL)		SUBSCRIPT OF INDEX ARRAY USED FOR INTERPOLATION WHEN DELTAT IS POSITIVE (IDELT2=IDELT1+1).
IDFL		(LOCAL)		IDFL=1 INDICATES HIGH LEVEL OF DIRECT FIRE SUPPRESSION; IDFL=2 INDICATES NO DIRECT FIRE SUPPRESSION. (IDFL IS USED TO PRINT CASE HEADING LABELS.)
IDFSRL		(LOCAL)		DIRECT FIRE SUPPRESSION DATA RECORD LENGTH ON WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
IDIGTL		FLAG		FLAG TO INDICATE WHICH METHOD OF COMMUNICATION IS TO BE FIRST CHOICE WHEN DESIGNATOR CALLS REQUEST FOR FIRE TO FDC. (IDIGTL=1 INDICATES DIGITAL COMMUNICATION; IDIGTL=2 INDICATES VOICE COMMUNICATION.)
IDN		FLAG		FLAG TO INDICATE DAY OR NIGHT. (IDN=1 INDICATES DAYTIME; IDN=2 INDICATES NIGHTTIME.)
IDOLOC		(LOCAL)		FLAG TO INDICATE DESIGNATOR OPERATOR LOCATION. (IDOLOC=1 INDICATES VANTAGE POINT LOCATION; IDOLOC=2 INDICATES MANEUVER UNIT LOCATION).
IDSGTP		RSPTIM		FLAG TO INDICATE DESIGNATOR TYPE. (IDSGTP=1 INDICATES GLLD, IDSGTP=2 INDICATES MULE, AND IDSGTP=3 INDICATES LTD).

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
IDT		(F.P.)		FORMAL PARAMETER INDICATING DESIGNATOR TYPE. VALUES OF 1,2, AND 3 HAVE SAME MEANING AS FOR THE VARIABLE IDSGTP.
IGO		(LOCAL)		COMPUTED GO TO PARAMETER USED TO SEND CONTROL TO SECTION OF SUBROUTINE INPUT THAT HANDLES A PARTICULAR CLASS OF INPUT LINES.
IGTR		(LOCAL)	KM	GUN-TO-TARGET RANGE ROUNDED TO NEAREST INTEGER.
II		(LOCAL)		USED AS SUBSCRIPT FOR INDEXING PETBL ARRAY. ALSO USED AS DO-LOOP CONTROL VARIABLE.
III		(LOCAL)		USED AS SUBSCRIPT FOR INDEXING PETBL ARRAY.
II2		(LOCAL)		USED AS SUBSCRIPT FOR INDEXING PETBL ARRAY.
IJ		(LOCAL)		CONTROL VARIABLE IN VARIOUS DO-LOOPS.
IK		(LOCAL)		CONTROL VARIABLE IN DO-LOOP.
IKTEST		(LOCAL)		DO-LOOP CONTROL VARIABLE IN REINITIALIZING KTEST ARRAY.
IL		(LOCAL)		CONTROL VARIABLE IN IMPLICIT DO-LOOP.
IMATCH		(LOCAL)		SUBSCRIPT OF KEYWORD CHARACTER STRING FOUND TO MATCH A CURRENT INPUT LINE CHARACTER STRING.
IMATCH1		(LOCAL)		STORES SUBSCRIPT OF KEYWORD CHARACTER STRING MATCHING FIRST CURRENT INPUT LINE KEYWORD WHILE SECOND KEYWORD IS BEING CHECKED FOR A MATCH (APPLIES ONLY TO "TEMPORARY", "OVERRIDE" AND "RESET" KEYWORDS).
IMNTH		(LOCAL)		INDEX NUMBER OF MONTH OF CURRENT CASE'S WEATHER DATA. (E.G., IF JUNE AND DECEMBER ARE THE ONLY MONTHS PLAYED, IMNTH=1 FOR JUNE AND IMNTH=2 FOR DECEMBER).
IMUTS		TARGET		FLAG AND SUBSCRIPT INDICATING WHETHER TARGET IS PAST THE POINT OF CLOSEST APPROACH TO THE CENTROID OF THE FOOTPRINT. (IMUTS=1 IF TARGET IS PAST POINT OF CLOSEST APPROACH; IMUTS=2 IF NOT).
IMXNR		(LOCAL)		NUMBER OF OPTION WORDS TO BE CHECKED FOR MATCH ON CURRENT INPUT LINE.
IN		(LOCAL)		NUMBER OF GAMMA DISTRIBUTION RANDOM DEVIATES TO BE RETURNED BY CALL OF IMSL SUBROUTINE GGAMA.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
INDEX(6,5,2)		HIT		INDEXING ARRAY FOR REFERENCING PETBL ARRAY. INDEX(I,J,K) IS THE NUMBER OF THE PETBL TO BE USED FOR CLOUD CEILING I, VISIBILITY RANGE J, AND IMUTS VALUE K.
INDEX11(2001)		INDEX1		ARRAY CONTAINING MASTER INDEX FOR WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
INVDRL		(LOCAL)		INVARIANT DATA RECORD LENGTH ON WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
IOCSN		(LOCAL)		NUMBER OF OCCASIONS FOR THE CURRENT CASE (EQUAL TO NREP).
IOPTNM(70)		(LOCAL)		IOPTNM(1) IS THE NUMBER OF DIFFERENT OPTION WORD CHARACTER STRINGS ALLOWED FOR KEYWORD I.
IOVER(70)		OVER		IOVER(1)=1 INDICATES "OVERRIDE" OPTION IN EFFECT FOR KEYWORD I; IOVER(1)=2 INDICATES "TEMPORARY" OPTION IN EFFECT FOR KEYWORD I.
IPEPL		(LOCAL)		PROBABILITY OF ENGAGEMENT DATA RECORD LENGTH ON WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
IPOINT(70)		POINT		IPOINT(1) IS THE VALUE OF THE COMPUTED GO TO PARAMETER THAT WILL SEND CONTROL THE PROPER PART OF SUBROUTINE INPUT FOR HANDLING KEYWORD I INPUTS.
IPPFPG		RUNDAT		PRINTER PLOT FLAG. IPPFG=1 INDICATES PRINTER PLOT OUTPUT REQUESTED FOR THIS CASE; IPPFG=0 INDICATES PRINTER PLOT OUTPUT NOT REQUESTED FOR THIS CASE.
IPREPL		FLAG		PRE-PLANNED TARGET FLAG. IPREPL=1 INDICATES PRE-PLANNED TARGET; IPREPL=2 INDICATES TARGET OF OPPORTUNITY.
IPRINT		(LOCAL)		FLAG TO INDICATE WHETHER CURRENT RANDOM NUMBER SEEDS HAVE YET BEEN WRITTEN TO TAPE 8. IPRINT=0 INDICATES THEY HAVE NOT; IPRINT=1 INDICATES THEY HAVE.
IPRTY		FLAG		PRIORITY FLAG. IPRTY=2 INDICATES PRIORITY PRE-PLANNED TARGET; IPRTY=0 INDICATES NO SPECIAL PRIORITY.
IRAY(6)		(LOCAL)		ARRAY CONTAINING ERROR PROCESSING SPECIFICATIONS FOR CALL OF SYSTEMC SUBROUTINE FOR NON-STANDARD RECOVERY FOR CDC FORTRAN ERROR 104.
IRC		(LOCAL)		UNUSED VALUE RETURNED BY A CALL OF SUBROUTINE SMPLOC.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
IRF		(LOCAL)		INITIAL PARAMETER OF A DO-LOOP (IRF=1).
IRM		(LOCAL)		SUBSCRIPT USED IN INTERPOLATING WITH RESPECT TO RANGE IN RNTTF ARRAY TO OBTAIN PROBABILITY OF HIT GIVEN AN ENGAGEMENT.
IRN		(LOCAL) (F.P.)		SUBSCRIPT USED IN INTERPOLATING WITH RESPECT TO RANGE IN PSTTBL ARRAY TO OBTAIN PROBABILITIES OF THE VARIOUS TARGET POSTURES. ALSO FORMAL PARAMETER FOR RANDOM NUMBER SEED IN GAMMA.
IRNDM		(F.P.)		RANDOM NUMBER SEED TO BE USED WHEN CALLING URAN31 FROM SMPLCD.
IRNDOC		FLAG		RANDOM OCCURRENCE FLAG. IRNDOC=1 INDICATES RANDOM OCCURRENCE LINE-OF-SIGHT METHODOLOGY USED FOR THIS CASE; IRNDOC=0 INDICATES SHOOTING GALLERY LOS METHODOLOGY USED FOR THIS CASE.
IRNG1		(LOCAL)	KM	LOWER LIMIT OF RANGE BRACKET USED TO INTERPOLATE IN PETBL ARRAY.
IRNG2		(LOCAL)	KM	UPPER LIMIT OF RANGE BRACKET USED TO INTERPOLATE IN PETBL ARRAY.
IRORL		(LOCAL)		RANDOM OCCURRENCE DATA RECORD LENGTH ON WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
IRON		(LOCAL)		NUMBER OF THE ROW OF THE SEGLOS ARRAY TO BE USED IN SAMPLING FOR LINE-OF-SIGHT SEGMENT LENGTH ON THIS REPLICATION.
IRSPRL		(LOCAL)		RESPONSE TIME DATA RECORD LENGTH ON WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
IR(28)		(LOCAL)		ARRAY OF RANDOM NUMBER SEEDS. IR(1) IS EQUIVALENT TO IR1, IR(2) TO IR2, . . . , IR(28) TO IR28.
IR1, IR2, . . . , IR28		RANDOM		RANDOM NUMBER SEEDS. IR1 IS SEED FOR FIRST RANDOM NUMBER STREAM, IR2 FOR SECOND, ETC.
ISHOT		(LOCAL)		NUMBER OF REPLICATIONS OF CURRENT CASE THAT REACHED THE STAGE IN THE MISSION WHERE A ROUND WAS FIRED.
ISTART		(LOCAL)		DO-LOOP INITIAL PARAMETER. ISTART IS THE SUBSCRIPT FOR THE FIRST LINE TO BE READ FROM CARDS AS RESULT OF A "TEMPORARY" OPTION FOR PEDATA.

GLOSSARY- CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
IT(10)		(LOCAL)		IT(1) = NUMBER OF CURRENT INPUT LINE'S KEYWORD. IT(J) IS THE NUMBER OF THE (J-1)ST OPTION WORD CHOICE OF THE CURRENT LINE FOR J>1.
ITDCTL		(LOCAL)		FLAG FOR CURRENT COMMUNICATIONS MODE. ITDCTL=1 FOR DIGITAL; ITDCTL=2 FOR VOICE.
ITGTPS		TARGET		TARGET POSTURE FLAG. ITGTPS=1 FOR FULLY EXPOSED TARGET; ITGTPS=2 FOR HULL DEFILADE TARGET.
ITGTFP		TARGET		TARGET TYPE NUMBER. ITGTFP=1 FOR TARGET TYPE 1, ITGTFP=2 FOR TARGET TYPE 2, ETC.
ITIMDL		(LOCAL)		ITIMDL=1 INDICATES FIRST METHOD OF MODELING RESPONSE TIME IS PLAYED (SINGLE DISTRIBUTION SAMPLED); ITIMDL=2 INDICATES SECOND METHOD OF MODELING RESPONSE TIME IS PLAYED (SINGLE VALUE USED); ITIMDL=3 INDICATES THIRD METHOD OF MODELING RESPONSE TIME IS PLAYED (ADDING CONTRIBUTIONS OF INDIVIDUAL DELAY COMPONENTS).
ITPSRL		(LOCAL)		TARGET POSTURE DATA RECORD LENGTH FOR WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
ITPRH		(LOCAL)		SUBSCRIPT INDICATING TERRAIN TYPE USED FOR CURRENT CASE. USED TO REFERENCE ACQBL ARRAY FOR OUTPUT HEADING.
ITRY		(LOCAL)		FLAG USED TO INDICATE WHETHER CURRENT ATTEMPT TO MATCH KEYWORD IS FOR FIRST OR SECOND KEYWORD OF CURRENT INPUT LINE. APPLIES ONLY TO "TEMPORARY", "OVERRIDE", AND "RESET" FIRST KEYWORDS.
ITVEL		(LOCAL)	METERS/SEC	TARGET VELOCITY.
ITYPE		(LOCAL)		COMPUTED GO TO PARAMETER USED TO SEND CONTROL TO PROPER READ STATEMENTS WHEN "TEMPORARY" OPTION IS USED.
IT1		(LOCAL)		NUMBER OF CURRENT INPUT LINE'S FIRST KEYWORD.
IT2		(LOCAL)		NUMBER OF CURRENT INPUT LINE'S SECOND KEYWORD OR FIRST OPTION WORD.
IT2		(LOCAL)		NUMBER OF CURRENT INPUT LINE'S SECOND OPTION WORD.
IT3FLG		FLAG		FLAG USED TO INDICATE WHETHER RUMC IS TO BE CONSTANT. IF IT3FLG=0, RUMC VARIES FROM REPLICATION TO REPLICATION, IF IT3FLG=1, RUMC IS CONSTANT.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
IUNIT		(F.P.) (LOCAL)		LOGICAL UNIT NUMBER TO WHICH OUTPUT OF SUBROUTINE ECHO IS TO BE WRITTEN. ALSO UNIT FROM WHICH "TEMPORARY" OPTION PEDATA IS TO BE READ IN SUBROUTINE CREAD.
IVL		WEATHER	KM	VISIBILITY RANGE LIMIT FOR THIS REPLICATION.
IVLA		(LOCAL)		VISIBILITY RANGE SUBSCRIPT VALUE FOR REFERENCING PETBL ARRAY (IVLA=IVL EXCEPT THAT IF IVL>10, THEN IVLA=10).
IW		(LOCAL)		COUNTER INDICATING POSITION NUMBER OF THE KEYWORD OR OPTION WORD CURRENTLY BEING ANALYZED IN THE SEPREC SUBROUTINE.
IWTHRL		(LOCAL)		WEATHER DATA RECORD LENGTH ON WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
IX		(LOCAL)		COUNTER USED IN SEPREC SUBROUTINE SEPARATION ALGORITHM.
IXXX		(LOCAL)		IXXX=1 INDICATES GOOD WEATHER (CURRENTLY JUNE AND SEPTEMBER) IXXX=2 INDICATES BAD WEATHER (CURRENTLY MARCH AND DECEMBER) IXXX GOVERNS THE VALUE OF THE VELTBL ARRAY TO BE USED IN THE CURRENT CASE. (APPLICABLE ONLY IF TARGET VELOCITY DEFAULT IS IN EFFECT.)
I1		(LOCAL)		CONTROL PARAMETER FOR IMPLIED DO-LOOPS.
I2		(LOCAL)		CONTROL PARAMETER FOR IMPLIED DO-LOOPS. ALSO RECEIVING ARGUMENT FOR DESIGNATOR TYPE IN CALL OF PEIDNT.
J		(LOCAL)		CONTROL VARIABLE IN VARIOUS DO-LOOPS. ALSO USED IN CALCULATION OF RANDOM NUMBER IN URAN31 AND AS A COUNTER IN SEPREC.
JEND		(LOCAL)		DO-LOOP MAXIMUM LIMIT (NUMBER OF DIFFERENT WIND SPEEDS USED IN SMOKE CALCULATION FOR THIS PASQUILL CATEGORY).
JJ		(LOCAL)		DO-LOOP CONTROL VARIABLE.
JK		(LOCAL)		CONTROL VARIABLE FOR DO-LOOPS AND IMPLIED DO-LOOPS.
JKTEST		(LOCAL)		CONTROL VARIABLE FOR DO-LOOP.
JL		(LOCAL)		CONTROL VARIABLE FOR IMPLICIT DO-LOOP.
K		(LOCAL)		CONTROL VARIABLE FOR VARIOUS DO-LOOPS AND IMPLICIT DO-LOOPS. ALSO USED IN CALCULATING INITIAL RANDOM NUMBER SEEDS.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
KCHAR		(LOCAL)		NUMBER OF CHARACTERS IN KEYWORD OR OPTION WORD ABOUT TO BE ENCODED.
KI		(LOCAL)		POSITION NUMBER OF CHARACTER OF OPTION WORD CURRENTLY BEING CHECKED TO DETERMINE WHETHER OPTION WORD IS A NUMERIC VALUE.
KREP		RPLCTN		NUMBER OF CURRENT REPLICATION.
KRF		RPLCTN		NUMBER OF THE ROUND (SHOT) CURRENTLY BEING SIMULATED.
KTEST(20,6)		KTEST		KTEST(I,J) IS THE NUMBER OF TIMES ABORT CONDITION I IS TESTED FOR ROUND (SHOT) NUMBER J. IF $I < MSABL M$, THEN $KTEST(I,J)=0$ FOR $J>1$; IN SUCH CASES $KTEST(I,1)$ IS THE NUMBER OF TIMES MISSION ABORT CONDITION I IS TESTED.
LIMU		(LOCAL)		MAXIMUM LIMIT ON DO-LOOP COMPARING KTEST AND NTEST ARRAYS. (LIMU=1 IF COMPARING MISSION ABORT TESTS; LIMU=6 IF COMPARING ROUND ABORT TESTS).
LSTCMA		(LOCAL)		GIVES POSITION OF LAST COMMA ENCOUNTERED THUS FAR IN SEPARATING KEYWORDS AND OPTION WORDS OF CURRENT INPUT LINE.
M		(LOCAL)		USED IN CALCULATION OF INITIAL RANDOM NUMBER SEEDS.
MSABL M		ABORT		MISSION ABORT LIMIT. CAUSE OF ABORT NUMBERS LESS THAN OR EQUAL TO MSABL M ARE MISSION ABORTS WHEREAS ABORT CAUSES NUMBERED GREATER THAN MSABL M ARE ROUND ABORTS. A MISSION ABORT CANCELS ALL FURTHER STEPS IN THE POTENTIAL COPPERHEAD FIRE MISSION (REPLICATION) WHEREAS A ROUND ABORT MEANS ONLY THAT THE CURRENT ROUND IS UNSUCCESSFUL.
MYDRNG	REAL	RANGE	METERS	MAXIMUM DESIGNATOR-TO-TARGET RANGE AT WHICH DESIGNATOR OPERATOR WILL CALL FOR COPPERHEAD FIRE.
N		(F.P.)		NUMBER OF CHARACTERS IN INPUT LINE WHOSE ITH CHARACTER IS CURRENTLY BEING CHECKED FOR SYMBOL CLASS.
NABORT(20,6)		ABORT		NABORT(I,J) IS THE NUMBER OF TIMES TEST I WAS FAILED ON ROUND (SHOT) J. IF $I < MSABL M$, THEN $NABORT(I,J)=0$ FOR $J>1$; IN SUCH CASES $NABORT(I,1)$ IS THE NUMBER OF TIMES MISSION ABORT TEST I WAS FAILED.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
NC(80)		(LOCAL)		NC(1) IS THE NUMBER INDICATING WHICH TYPE OF SYMBOL THE ITH CHARACTER OF THE CURRENT INPUT LINE IS. (0 INDICATES NUMERICAL OR MINUS SIGN, 1 INDICATES LETTER OR APOSTROPHE, 2 INDICATES DECIMAL POINT, 3 INDICATES BLANK, 4 INDICATES SEPARATOR, 5 INDICATES A DOLLAR SIGN, AND 6 INDICATES ANY OTHER CHARACTER.)
NCASE		RUNDAT		SEQUENCE NUMBER OF CASE CURRENTLY BEING EXECUTED IN THIS PROGRAM RUN.
NCC		WEATHR		NUMBER OF DIFFERENT CLOUD CEILING ALTITUDE VALUES USED IN WEATHER DATA.
NCODE		(LOCAL)		USED IN CREATING AND DECIPHERING PEDATA RECORD NAME.
NCPR(10)		(LOCAL)		NCPR(1) IS THE NUMBER OF CHARACTERS IN THE ITH LINE OF THE B ARRAY (ITH FIELD ON INPUT LINE AFTER COMPRESSING MOST BLANKS AND POSSIBLY ADDING DECIMAL POINTS).
NCRD(70)		(LOCAL)		NCRD(1) IS THE NUMBER OF INPUT LINES READ FOR THE CURRENT CASE THAT DETERMINE THE OPTIONS USED WITH KEYWORD NUMBER 1.
NDFSP		DODF		NUMBER OF RANGE POINTS USED IN DIRECT FIRE SUPPRESSION DISTRIBUTION (DFOOKL ARRAY).
NDOKIL		ROOF		NUMBER OF REPLICATIONS IN WHICH DESIGNATOR OPERATOR HAS BEEN KILLED FOR CURRENT CASE.
NDR		(F.P.)		INDICATES NUMBER OF POINTS USED IN DEFINITION OF CUMULATIVE DISTRIBUTION FUNCTION NOW BEING SAMPLED.
NDRMI		(LOCAL)		MAXIMUM LIMIT ON DO-LOOP. (EQUALS NDR-1).
NDT		RSPTIM		NUMBER OF POINTS USED IN DETECTION TIME DISTRIBUTION.
ND1		(F.P.)		FIRST DIMENSION OF ARRAY WHOSE ELEMENTS DEFINE CUMULATIVE DISTRIBUTION BEING SAMPLED.
ND2		(F.P.)		SECOND DIMENSION OF ARRAY WHOSE ELEMENTS DEFINE CUMULATIVE DISTRIBUTION BEING SAMPLED.
NKILL(6)		(LOCAL)		NKILL(1) IS THE NUMBER OF REPLICATIONS IN WHICH THE ITH ROUND OF THE FIRE MISSION KILLED A VEHICLE FOR THE CURRENT CASE.
NLRTC		(LOCAL)		NUMBER OF INPUT LINES READ SO FAR FOR THE CURRENT CASE.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
NP(9)		(LOCAL)		USED TO CONSTRUCT THE PEDATA RECORD NAME. NP(1) IS THE INDEX NUMBER (IN THE XVALUE ARRAY) OF THE CURRENT VALUE OF THE ITH PARAMETER OF THE PEDATA RECORD NAME.
NPLOS		RANDOC		NUMBER OF RANGE POINTS USED IN THE PROBABILITY OF LINE-OF-SIGHT FOR RANDOM OCCURRENCE (RNGPLS ARRAY).
NPP		RNGLOS		NUMBER OF POINTS USED IN DEFINING CUMULATIVE DISTRIBUTIONS FOR LINE-OF-SIGHT SEGMENT LENGTHS.
NRE		(LOCAL)		NUMBER OF KEYWORDS AND OPTION WORDS ENCOUNTERED ON CURRENT INPUT LINE.
NREC		ACHAR		NUMBER OF DIFFERENT KEYWORD CHARACTER STRINGS ALLOWED AS INPUTS.
NREP		RPLCTN		NUMBER OF REPLICATIONS TO BE DONE FOR THE CURRENT CASE.
NRF		RPLCTN		NUMBER OF ROUNDS (SHOTS) TO BE FIRED FOR EACH REPLICATION (POTENTIAL FIRE MISSION SAMPLE) FOR THIS CASE.
NRNGCB		(LOCAL)		NUMBER OF ROWS TO BE READ INTO SEGLOS DISTRIBUTION.
NRNGCL		RNGLOS		NUMBER OF RANGE CLASSES INTO WHICH LINE-OF-SIGHT SEGMENT LENGTH DATA IS DIVIDED.
NRNGPI		(LOCAL)		MAXIMUM LIMIT ON DO-LOOP. (EQUALS NRNGPS)
NRNGPS		HIT		NUMBER OF RANGE POINTS FOR WHICH TARGET POSTURE DATA IS ENTERED IN RNGPST ARRAY.
NRNGTT		HIT		NUMBER OF RANGE POINTS FOR WHICH "TRUE TARGET FACTOR" DATA IS ENTERED IN RNGTTF ARRAY.
NRP		RNGLOS		NUMBER OF POINTS USED IN CUMULATIVE ACQUISITION RANGE DISTRIBUTION TABLE (CRNGD ARRAY).
NRS		(F.P.)		ROW OF A ARRAY IN WHICH INTERPOLATION IS TO BE PERFORMED TO OBTAIN RANDOM VARIABLE VALUE IN SUBROUTINE SMPLOC.
NSMK2		SMOKED	TYPE 1 SMOKE ROUNDS	NUMBER OF TYPE 1 SMOKE ROUNDS FIRED BY RED TO CREATE SMOKE SCREEN.
NSMK5		SMOKED	TYPE 2 SMOKE ROUNDS	NUMBER OF TYPE 2 SMOKE ROUNDS FIRED BY RED TO CREATE SMOKE SCREEN.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
NTEST(20,6)		(LOCAL)		NTEST(I,J) IS THE NUMBER OF TIMES ABORT CONDITION I HAS BEEN TESTED FOR ROUND J FOR THIS CASE. FOR I<MSABLM, NTEST(I,J)=NTEST(J,1) FOR J>1.
NTRC		COMENT		NUMBER OF "TRIPLE \$" COMMENTS READ AND WRITTEN WITH CURRENT CASE.
NUMRIC	LOGICAL	FUNCTION VALUE		TRUE, IF THE ITEM TESTED IS NUMERIC (CONTAINS ONLY NUMERALS, "-", AND "." AS SYMBOLS); FALSE, IF ITEM IS NOT NUMERIC.
NVEHCL		TARGET		NUMBER OF VEHICLES IN TARGET UNIT.
NVEHKL		RPLCTN		NUMBER OF VEHICLES KILLED BY COPPERHEAD THUS FAR IN CURRENT REPLICATION. (APPLIES ONLY TO RANDOM OCCURRENCE LINE-OF-SIGHT OPTION).
NVL		WEATHR		NUMBER OF DIFFERENT METEOROLOGICAL VISIBILITY RANGE LIMITS USED WITH WEATHER DATA.
NXTLIN		(LOCAL)		COUNTER USED IN KEEPING TRACK OF LINE NUMBERS IN PRINTING OF CASE HEADING OUTPUT.
N1		(F.P.)		VARIABLE ARRAY DIMENSION.
N2		(F.P.)		VARIABLE ARRAY DIMENSION.
N3		(F.P.)		VARIABLE ARRAY DIMENSION.
OUT(6)		(LOCAL)		ARRAY USED TO STORE ITEMS ABOUT TO BE WRITTEN ON THE PRINTOUT.
PAREN	LOGICAL	(F.P.)		FLAG TO INDICATE WHETHER A RIGHT PARENTHESIS IS TO BE INCLUDED IN FORMATTING A PERCENTAGE. (TRUE INDICATES PARENTHESIS IS TO BE INCLUDED; FALSE INDICATES IT IS NOT).
PASPCT(20,6)		(LOCAL)		PASPCT(I,J) IS THE PERCENTAGE OF REPLICATIONS TESTED AT ABORT CONDITION I THAT PASSED (DID NOT ABORT) FOR ROUND J. (IF I<MSABLM, PASPCT(I,J) = PASPCT(I,1) FOR J>1; IF NTEST(I,J)=0, PASPCT(I,J) IS SET TO -1).
PASQL(6)		WEATHR		PASQL(1) IS THE PROBABILITY OF OCCURRENCE OF PASQUILL ATMOSPHERIC STABILITY CATEGORY I FOR THE CURRENT CASE'S WEATHER CONDITIONS.
PASQT		(LOCAL)		THE PASQL VALUES ARE GROUPED IN PAIRS. PASQT IS THE SUM OF PASQL(2*1-1) AND PASQL(2*1) FOR THE CURRENT VALUE OF I.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
PB(80,10)		(LOCAL)		PB(I,J) IS THE ITH CHARACTER OF THE JTH KEYWORD OR OPTION WORD OF THE CURRENT INPUT LINE.
PCRMMSG		RSPTIM		PROBABILITY THAT THE DESIGNATOR-TO-FDC CALL-FOR-FIRE MESSAGE CONTAINS THE CORRECT FIRE MISSION INFORMATION AND IS CORRECTLY INTERPRETED.
PDOOKIL		(LOCAL)		PROBABILITY THAT DESIGNATOR OPERATOR IS KILLED BY DIRECT FIRE FROM TARGET VEHICLE.
PDOSUP		(LOCAL)		PROBABILITY THAT DESIGNATOR OPERATOR IS SUPPRESSED DUE TO OBSCURANTS CREATED BY NEAR MISSES CAUSED BY DIRECT FIRE FROM TARGET VEHICLE.
POSTKL		MISC		PROBABILITY THAT DUST FROM HE ARTILLERY FIRE IS SUFFICIENT TO OBSCURE TARGET AND PREVENT USE OF COPPERHEAD.
PE		(LOCAL)		PROBABILITY THAT ROUND ENGAGES TARGET (I.E., SUFFICIENT REFLECTED LASER ENERGY IS PICKED UP BY THE SEEKER AND THE TARGET IS WITHIN THE MANEUVER FOOTPRINT OF THE COPPERHEAD ROUND).
PECHK		(LOCAL)		USED TO DECIPHER PEDATA RECORD NAME FOR CASE HEADING OUTPUT.
PECODE		RECNAME		RECORD NAME KEY FOR PROBABILITY OF ENGAGEMENT DATA RECORD ON WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
PEDATA(4260)		(LOCAL)		ARRAY TO WHICH THE ELEMENTS OF THE PROBABILITY OF ENGAGEMENT (PE) DATA BLOCK ARE EQUIVALENCED.
PETBL(60,10,7)		HIT		PETBL(I,J,K) IS THE PROBABILITY THAT THE COPPERHEAD ROUND ENGAGES THE TARGET GIVEN INDEX VALUE I, VISIBILITY RANGE LIMIT J, AND DESIGNATOR TARGET RANGE K.
PETIR1		(LOCAL)		PROBABILITY OF ENGAGEMENT VALUE FOR TIME 1 AND RANGE 1 WHEN INTERPOLATING IN PROBABILITY OF ENGAGEMENT DATA WITH IMUTS=1
PETIR2		(LOCAL)		SAME AS PETIR1 BUT FOR TIME 1 AND RANGE 2.
PET2R1		(LOCAL)		SAME AS PETIR1 BUT FOR TIME 2 AND RANGE 1.
PET2R2		(LOCAL)		SAME AS PETIR1 BUT FOR TIME 2 AND RANGE 2.
PEI		(LOCAL)		LOWER RANGE PROBABILITY OF ENGAGEMENT VALUE WHEN INTERPOLATING WITH RESPECT TO RANGE IN PROBABILITY OF ENGAGEMENT DATA WITH IMUTS=2.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
PE2		(LOCAL)		UPPER RANGE PROBABILITY OF ENGAGEMENT VALUE WHEN INTERPOLATING WITH RESPECT TO RANGE IN PROBABILITY OF ENGAGEMENT DATA WITH IMUTS=2.
PH		(LOCAL)		PROBABILITY ROUND HITS TARGET GIVEN THAT ROUND ENGAGES.
PK		(LOCAL)		PROBABILITY ROUND KILLS TARGET VEHICLE GIVEN THAT ROUND HITS TARGET VEHICLE.
PKIBL(10,2)		HIT		PKIBL(I,J) IS THE PROBABILITY OF KILL GIVEN A HIT FOR COUNTERHEAD AGAINST TARGET TYPE I IN TARGET POSTURE J.
PR		(LOCAL)		USED TO BRACKET UNIFORM RANDOM NUMBER IN A WEATHER CLASS PROBABILITY TABLE.
PRBAEO		(LOCAL)		PROBABILITY OF ATTEMPTED ENGAGEMENT GIVEN AN OCCASION.
PRBKAE		(LOCAL)		PROBABILITY OF KILL GIVEN AN ATTEMPTED ENGAGEMENT.
PRBKO		(LOCAL)		PROBABILITY OF KILL GIVEN AN OCCASION.
PRBKS		(LOCAL)		PROBABILITY OF KILL GIVEN A SHOT.
PRBLOS(10)		RANDOC		PRBLOS(I) IS THE PROBABILITY THAT A SINGLE VEHICLE LOCATED AT RANDOM ALONG AN APPROACH PATH IN RANGE BRACKET I WILL BE WITHIN THE DESIGNATOR OPERATOR'S LINE-OF-SIGHT FOR AT LEAST THE CRITICAL TIME (TCRIT SECONDS).
PRBSAE		(LOCAL)		PROBABILITY OF SHOT GIVEN AN ATTEMPTED ENGAGEMENT.
PRBSO		(LOCAL)		PROBABILITY OF SHOT GIVEN AN OCCASION.
PRCFLS(2)		WEATHR		PRCFLS(I) IS THE PROBABILITY OF CLOUD FREE LINE-OF-SIGHT GIVEN CONDITION I. (I=1 INDICATES A CLOUD CEILING; I=2 INDICATES SCATTERED CLOUDS ONLY).
PRCLCG		WEATHR		PROBABILITY THAT THERE IS A CLOUD CEILING.
PRDOKC		(LOCAL)		PROBABILITY DESIGNATOR KILLED BY DIRECT FIRE GIVEN THAT DIRECT FIRE OCCURRED.
PROOKL		(LOCAL)		PROBABILITY DESIGNATOR KILLED GIVEN AN OCCASION
PROOW		RSPTIM		PROBABILITY THAT THE DESIGNATOR-OPERATOR IS WARNED TO BEGIN LASING IN TIME TO GUIDE THE FIRST ROUND OF THE FIRE MISSION TO THE TARGET.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
PRGCFL(11,2)		WEATHR		PRGCFL(I,J) IS THE PROBABILITY OF METEOROLOGICAL VISIBILITY RANGE LIMIT I GIVEN CLOUD FREE LINE-OF-SIGHT AND CONDITION J (J=1 INDICATES CLOUD CEILING; J=2 INDICATES SCATTERED CLOUDS)
PRHLOS(6)		(LOCAL)		PRHLOS(I) IS THE PROBABILITY THE LINE-OF-SIGHT EXISTS FROM DESIGNATOR TO TARGET FOR ROUND I. (I.E., LOS EXISTS BOTH BEFORE FIRING AND DURING TERMINAL PHASE OF TRAJECTORY).
PRIV		(LOCAL)		PROBABILITY THAT AT LEAST ONE TARGET IS IN VIEW FOR THE CRITICAL TIME (TCRIT SECONDS) DURING THE TERMINAL PHASE OF THE COPPERHEAD TRAJECTORY.
PRLS		(LOCAL)		PROBABILITY OF LINE-OF-SIGHT TO A SINGLE TARGET FOR THE CRITICAL TIME DURATION (TCRIT SECONDS) DURING THE TERMINAL PHASE OF THE COPPERHEAD TRAJECTORY.
PRSDT		RSPTIM		PROBABILITY THAT DESIGNATOR TO FDC DIGITAL COMMUNICATION LINK IS OPERATING SUCCESSFULLY.
PRSPM		RSPTIM	SECONDS	PARAMETERIZED RESPONSE TIME.
PRSVT		RSPTIM		PROBABILITY THAT DESIGNATOR TO FDC VOICE COMMUNICATION IS SUCCESSFUL.
PR1		(LOCAL)		USED TO BRACKET RANDOM NUMBER IN A WEATHER CLASS PROBABILITY TABLE.
PSCODE		RECNAME		NAME RECORD KEY FOR TARGET POSTURE DATA RECORD ON WORD ADDRESSABLE RANDOM ACCESS MASS STORAGE FILE (TAPE 11).
PSMKKL		SMOKED		PROBABILITY THAT SMOKE ABORTS POTENTIAL COPPERHEAD FIRE MISSION.
PSTDAT(41)		(LOCAL)		ARRAY TO WHICH THE ELEMENTS OF THE TARGET POSTURE DISTRIBUTION DATA BLOCK ARE EQUIVALENCED.
PSTTBL(10,3)		HIT		PSTTBL(I,J) IS THE PROBABILITY THAT A MOVING TARGET AT RANGE I IS IN POSTURE J (WHERE J=1 INDICATES A COMPLETELY OBSCURED TARGET, J=2 INDICATES A FULLY EXPOSED TARGET, AND J=3 INDICATES A HULL DEFILADE TARGET).
PSTVAL(3)		(LOCAL)		PSTVAL(I) IS THE PROBABILITY THAT THE TARGET IS IN POSTURE I WHERE I=1,2 AND 3 RESPECTIVELY INDICATE THAT THE TARGET IS COMPLETELY OBSCURED, FULLY EXPOSED, AND HULL DEFILADE.
R(1)		(LOCAL)		GAMMA RANDOM DEViate.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
RECLBL(8)		DISPLY		RECLBL(1) IS AN ALPHANUMERIC CODE THAT FORMS PART OF THE RECORD NAME KEY FOR WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11) RECORDS OF TYPE I. (WHICH CORRESPOND TO KEYWORD 20+1).
REFL		PEDESC (F.P.)		REFLECTIVITY OF TARGET TO LASER ENERGY IN DESIGNATOR WAVE LENGTH.
RELIF		MISC		ROUND IN FLIGHT RELIABILITY. PROBABILITY THAT ROUND FUNCTIONS CORRECTLY.
RN		(LOCAL)		UNIFORMLY DISTRIBUTED RANDOM NUMBER.
RNDREC(6)		SCCSS		RNDREC(1) IS THE NUMBER OF TARGETS KILLED FOR THE CURRENT CASE BY THE ITH ROUND. (RNDREC(1) = NKILL(1).)
RNGCLB(11)		RNGLOS	METERS	RNGCLB(1) IS THE UPPER BOUNDARY OF THE ITH RANGE CLASS FOR LINE-OF-SIGHT SEGMENT LENGTHS.
RNGK		(LOCAL)	KM	RNGK IS THE CURRENT DESIGNATOR-TO-TARGET RANGE IN KILOMETERS
RNGNOW		(LOCAL)	METERS	CURRENT DESIGNATOR-TO-TARGET RANGE IN METERS.
RNGPLS(10)		RANDOC	METERS	RNGLOS(1) IS THE UPPER BOUNDARY OF THE ITH RANGE CLASS OF PRBLOS ARRAY.
RNGPST(10)		HIT	METERS	RNGPST(1) IS THE DESIGNATOR-TO-TARGET RANGE FOR THE ITH POSTURE BREAKDOWN DISTRIBUTION IN THE PSTTBL ARRAY.
RNGTTF(20)		HIT	METERS	RNGTTF(1) IS THE DESIGNATOR-TO-TARGET RANGE FOR THE ITH SET OF "TRUE TARGET FACTORS" IN THE TTF ARRAY.
RN1, RN2, . . . RN22		(LOCAL)		THE RANDOM NUMBERS RESULTING FROM CALLS OF URAN31 (I.E., RN1=URAN31 (IRI)) SOME OF THE RN'S (RN9 FOR EXAMPLE) ARE NO LONGER PRESENT IN THE PROGRAM).
ROCODE		RECNAM		RECORD NAME KEY FOR RANDOM OCCURRENCE DATA RECORD ON WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
RODATA(22)		(LOCAL)		ARRAY TO WHICH THE ELEMENTS OF THE RANDOM OCCURRENCE DATA BLOCK ARE EQUIVALENCED.
ROLBL(4,2)		HEADING		ARRAY OF ALPHANUMERIC CONSTANTS USED TO CREATE LABELING OF LOS MODE SECTION OF OUTPUT.
RSCODE		RECNAM		RECORD NAME KEY FOR RESPONSE TIME DATA RECORD ON WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
RSPDAT(49)		(LOCAL)		ARRAY TO WHICH THE ELEMENTS OF THE RESPONSE TIME DATA BLOCK ARE EQUIVALENCED.
RSPLEL(3,3)		HEADNG		ARRAY OF ALPHANUMERIC CONSTANTS USED TO CREATE LABELING OF RESPONSE TIME SECTION OF OUTPUT.
RUMC		RSPTIM	METERS	RANGE FROM TARGET UNMASK TO FOOTPRINT CENTROID.
SEGLNG		(LOCAL)	METERS	LENGTH OF LINE-OF-SIGHT SEGMENT DRAWN FOR THE CURRENT REPLICATION.
SEGLOS(11,11)		RNGLOS	NONE, METERS	SEGLOS(I,J) FOR J-1 IS THE ITH PROBABILITY VALUE USED FOR BRACKETING A UNIFORM RANDOM NUMBER WHEN INTERPOLATING IN CUMULATIVE LOS SEGMENT LENGTH DISTRIBUTION. SEGLOS(I,J) FOR J>1 IS THE ITH LINE-OF-SIGHT SEGMENT LENGTH VALUE USED FOR INTERPOLATING IN RANGE CLASS J-1. (SEGLOS(1,1) IS THE PROBABILITY THAT A LINE-OF-SIGHT SEGMENT IN RANGE CLASS J-1 IS LESS THAN OR EQUAL TO SEGLOS(1,J)).
SEP(5)		SYMBOL		AN ARRAY OF ALPHANUMERIC SYMBOLS THAT CAN BE USED AS KEYWORD OR OPTION WORD SEPARATORS (" ", ",", "(", ")", " / ", "?") THEY ARE STORED IN A1 FORMAT.
SEQNML	LOGICAL	LOGFLG		FLAG TO INDICATE WHETHER SEQUENCE NUMBERS (FROM CDC EDITOR) ARE PRESENT IN COLUMNS 73-78 OF INPUT FILE. (TRUE INDICATES SEQUENCE NUMBERS; FALSE INDICATES NO SEQUENCE NUMBERS).
SHRTEC	LOGICAL	LOGFLG		FLAG TO INDICATE WHETHER SHORT OR LONG FORM OF SUBROUTINE ECHO'S OUTPUT IS TO BE USED. (SHRTEC=TRUE FOR SHORT FORM; SHRTEC=FALSE FOR LONG FORM).
SKSEN		PEDESC F.P.	JOULES/M ²	COPPERHEAD SEEKER SENSITIVITY.
SLCOM3	LOGICAL	F.P.		FLAG TO INDICATE WHETHER A COMMENT IS A PROGRAM GENERATED "TRIPLE \$" COMMENT. (IF TRUE, IT IS; IF FALSE, IT IS NOT).
SLVERS		STITLE		VERSION NUMBER OF PROGRAM (INCREASED AS PROGRAM IS MODIFIED)
SMK2(3,3,2)		SMOKED	TYPE 1 SMK ROUNDS	SMK2(I,J,K) IS THE NUMBER OF TYPE 1 SMOKE ROUNDS REQUIRED TO CREATE A SMOKE SCREEN FOR 30 MINUTES ACROSS 1 KILOMETER OF FRONT SUFFICIENT TO PREVENT THE USE OF COPPERHEAD AGAINST TARGETS IN THE SMOKED AREA GIVEN PASQUILL ATMOSPHERIC STABILITY CATEGORY 2*1 OR 2*1-1, WIND SPEED J, AND RELATIVE HUMIDITY K.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
SMK5(3,3,2)		SMOKED	TYPE 2 SMK ROUNDS	SAME AS SMK2 EXCEPT FOR TYPE 2 SMOKE ROUND.
SPCL(70)	LOGICAL	LOGFLG		SPCL(1) IS TRUE IF A SPECIAL OPTION IS IN EFFECT FOR KEYWORD 1; FALSE IF SPECIAL OPTION NOT IN EFFECT.
TBR		TIME	SECONDS	TIME BETWEEN SUCCESSIVE ROUNDS FIRED ON A SINGLE REPLICATION.
TCRIT		RANDOC	SECONDS	CRITICAL TIME. LENGTH OF TIME INTERVAL AT TERMINAL PHASE OF TRAJECTORY DURING WHICH COPPERHEAD SEEKER MUST BE TRACKING A SPOT DESIGNATED ON (OR NEAR) THE TARGET.
TF		(LOCAL)		FACTOR USED FOR INTERPOLATING WITH RESPECT TO RANGE IN PROBABILITY OF ENGAGEMENT TABLES WHEN IMUTS=2.
TFRAC		(LOCAL)		WEIGHTING FACTOR USED FOR INTERPOLATING WITH RESPECT TO RANGE IN TARGET POSTURE DISTRIBUTION TABLE.
TFRAC1		(LOCAL)		WEIGHTING FACTOR USED FOR INTERPOLATING WITH RESPECT TO RANGE IN "TRUE TARGET FACTOR" TABLE.
TF1		(LOCAL)		WEIGHT FACTOR APPLIED FOR SHORTER RANGE WHEN INTERPOLATING WITH RESPECT TO RANGE IN PROBABILITY OF ENGAGEMENT TABLES WITH IMUTS=1.
TF2		(LOCAL)		WEIGHT FACTOR APPLIED FOR LONGER RANGE WHEN INTERPOLATING WITH RESPECT TO RANGE IN PROBABILITY OF ENGAGEMENT TABLES WITH IMUTS=1.
TGTHD		PEDESC (F.P.)	DEGREES	ANGLE BETWEEN FOOTPRINT CENTROID-TO-GUN LINE AND TARGET DIRECTION OF TRAVEL.
TIM		RUNDAT		ALPHANUMERIC DISPLAY OF TIME AT WHICH PROGRAM WAS RUN (OBTAINED FROM COMPUTER'S INTERNAL CLOCK).
TINNOW		TIME	SECONDS	CURRENT TIME IN SECONDS SINCE TARGET UNMASKED FROM TERRAIN.
TIMRA		TIME	SECONDS	TIME AT WHICH ROUND CURRENTLY BEING CONSIDERED WILL ARRIVE (IMPACT).
TIMRA1		TIME	SECONDS	TIME AT WHICH FIRST ROUND OF THIS REPLICATION WILL ARRIVE (IMPACT).

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON/CLK</u>	<u>UNITS</u>	<u>DEFINITION</u>
TMEAN(2,2)		RSPTIM	SECONDS	TMEAN(I,J) IS THE MEAN TIME REQUIRED TO ENTER COPPERHEAD FIRE MISSION REQUEST IN DIGITAL MESSAGE DEVICE FOR CONDITION I, J. (I=1 INDICATES DAY, I=2 INDICATES NIGHT, J=1 INDICATES PRE-PLANNED TARGET, J=2 INDICATES TARGET OF OPPORTUNITY).
TMPLBL(8)		DISPLY		ALPHANUMERIC CONSTANTS USED TO LABEL OUTPUT HEADINGS WHEN TEMPORARY OR OVERRIDE OPTION IS USED.
TOF		TIME	SECONDS	TIME OF FLIGHT FOR COPPERHEAD ROUND (TIME FROM FIRING UNTIL IMPACT).
TOFARY(16)		FLTTIM	SECONDS	TOFARY(I) IS THE TIME OF FLIGHT OF COPPERHEAD ROUND (USING PREFERRED ELEVATION AND CHARGE) FOR GUN-TO-TARGET RANGE OF I KILOMETERS.
TR		(LOCAL)	SECONDS	NOMINAL DESIGNATOR-FDC-BATTERY RESPONSE TIME.
TRARRY(3,2)		RSPTIM	SECONDS	TRARRY(I,J) IS THE NOMINAL DESIGNATOR-FDC-BATTERY RESPONSE TIME FOR MISSION TYPE I AND COMMUNICATION TYPE J. (I=1 INDICATES PRE-PLANNED TARGET, I=2 INDICATES TARGET OF OPPORTUNITY, I=3 INDICATES PRIORITY PRE-PLANNED TARGET, J=1 INDICATES DIGITAL COMMUNICATION, J=2 INDICATES VOICE COMMUNICATION).
TRBAR		RSPTIM	SECONDS	NOMINAL DESIGNATOR-FDC-BATTERY RESPONSE TIME PLUS TIME OF FLIGHT.
TRBAR1		(LOCAL)	SECONDS	SAME AS TRBAR.
TSIGMA(2,2)		RSPTIM	SECONDS	TSIGMA(I,J) IS THE STANDARD DEVIATION OF THE DISTRIBUTION OF TIME REQUIRED TO ENTER COPPERHEAD FIRE MISSION REQUEST IN DIGITAL MESSAGE DEVICE FOR CONDITION I,J. (I,J HAVE SAME MEANING AS FOR THE VARIABLE TMEAN).
TTF(3,20,2)		HIT		TTF(I,J,K) IS THE PROBABILITY COPPERHEAD HITS THE TARGET GIVEN SEEKER ENGAGEMENT WITH DESIGNATOR TYPE I, RANGE J, AND TARGET POSTURE K. (I=1,2,3 INDICATE DESIGNATOR TYPES 1,2,3 RESPECTIVELY; J INDICATES JTH RANGE OF RGTTF ARRAY; K=1 INDICATES FULLY EXPOSED TARGET, K=2 INDICATES HULL DEFILADE TARGET).
TVEL		TARGET	METERS/ SECONDS	TARGET SPEED.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
T1		(LOCAL)	SECONDS	FIRST TIME DELAY COMPONENT OF DELTAT. (ACCOUNTS FOR MULTIPLE VEHICLES AND MULTIPLE ROUNDS). ALSO, NOMINAL RESPONSE TIME VALUE FOR PRINTING IN COMMENT.
T2		(LOCAL)	SECONDS	SECOND TIME DELAY COMPONENT OF DELTAT. (ACCOUNTS FOR ACTUAL RESPONSE TIME AND DETECTION TIME VERSUS NOMINAL RESPONSE TIME).
T3		(LOCAL)	SECONDS, M/S	THIRD TIME DELAY COMPONENT OF DELTAT. (ACCOUNTS FOR POSITIONING OF FOOTPRINT WHEN RUMC IS PLAYED). ALSO, TARGET SPEED VALUE FOR PRINTING IN COMMENT.
T4		(LOCAL)	KM	GUN-TO-TARGET RANGE VALUE FOR PRINTING IN COMMENT.
T5		(LOCAL)		TARGET REFLECTIVITY VALUE FOR PRINTING IN COMMENT.
T6		(LOCAL)	DEGREES	ANGLE T VALUE FOR PRINTING IN COMMENT.
T7		(LOCAL)	METERS	DEFLECTION BIAS (POINT OF CLOSEST APPROACH TO FOOTPRINT CENTROID) VALUE FOR PRINTING IN COMMENT.
T8		(LOCAL)	DEGREES	TARGET HEADING VALUE FOR PRINTING IN COMMENT.
T9		(LOCAL)	JOULES/M ²	SEEKER SENSITIVITY VALUE FOR PRINTING IN COMMENT.
U(70)		UVALUE		U(1) IS A VALUE ASSOCIATED WITH KEYWORD 1 AND USED IN THE CURRENT CASE (EQUIVALENT TO UVALUE).
UDT		UVALUE		NO LONGER USED.
UIDIGT		UVALUE		UIDIGT=1.0 INDICATES DIGITAL DESIGNATOR-FDC COMMUNICATIONS; UIDIGT=2 INDICATES VOICE DESIGNATOR-FDC COMMUNICATIONS.
UIDN		UVALUE		UIDN=1 INDICATES DAY; UIDN=2 INDICATES NIGHT.
UMC		UVALUE		USED MISSION CODE. UMC=1,2,3 MEAN RESPECTIVELY PRE-PLANNED TARGET, TARGET OF OPPORTUNITY, PRIORITY PRE-PLANNED TARGET.
UPRT		UVALUE	SECONDS	USED PARAMETERIZED RESPONSE TIME.
UPAN31		FUNCTION VALUE		UPAN31 IS THE UNIFORM (PSEUDO-) RANDOM NUMBER GENERATOR USED IN THIS PROGRAM.
USERC	LOGICAL	F.P.		USERC=TRUE INDICATES A USER'S COMMENT; USERC=FALSE INDICATES A PROGRAM GENERATED COMMENT.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
USMK(2)		UVALUE		USMK(1) IS THE NUMBER OF TYPE 1 SMOKE ROUNDS USED. USMK(2) IS THE NUMBER OF TYPE 2 SMOKE ROUNDS USED.
UVALUE(70)				EQUIVALENT TO U ARRAY.
VEL		F.P.	METERS/ SECONDS	TARGET SPEED.
VELTBL(2,2)		RNGLOS	METERS/ SECONDS	VELTBL(I,J) IS TARGET SPEED IN CONDITIONS I,J. (I=1 FOR DAY, I=2 FOR NIGHT, J=1 FOR SUMMER, J=2 FOR WINTER).
VERDAT(2)		STITLE		DATE (ALPHANUMERIC CONSTANT) OF LAST CHANGE TO CURRENT PROGRAM.
VIS(11)		WEATHR	KM	VIS(I) IS THE ITH METEOROLOGICAL VISIBILITY RANGE LIMIT VALUE USED IN THE WEATHER DATA.
VISLIM		RANGE	KM	THE METEOROLOGICAL VISIBILITY RANGE LIMIT VALUE FOR THE CURRENT REPLICATION.
VSMK		(LOCAL)	KM	LENGTH OF FRONT THAT CAN BE SMOKE WITH GIVEN NUMBER OF ROUNDS AND GIVEN WEATHER CONDITIONS. INTERMEDIATE VARIABLE IN CALCULATION OF PSMKKL.
W(6,11,2)		WEATHR		W(I,J,K) IS THE PROBABILITY (GIVEN A CLOUD CEILING) OF HAVING WEATHER WITH CLOUD CEILING ALTITUDE I, METEOROLOGICAL VISIBILITY RANGE LIMIT J, AND CLOUD FREE LOS CONDITION K (K=1 INDICATES CLOUD FREE LOS; K=2 INDICATES NO CLOUD FREE LOS).
WCFSUM(2)		WEATHR		SCFSUM(I) IS THE SUM OF ELEMENTS PRGCF(1,1) THROUGH PRGCF(L,NVL,1). (USED TO NORMALIZE RANDOM NUMBERS BEFORE SAMPLING FROM PRGCFL ARRAY).
WCODE		RECNAME		RECORD NAME KEY FOR WEATHER DATA RECORD ON WORD ADDRESSABLE MASS STORAGE FILE (TAPE 11).
WDATA(193)		(LOCAL)		ARRAY TO WHICH THE ELEMENTS OF THE WEATHER DATA BLOCK ARE EQUIVALENT.
WK(6)		LOCAL		WORKING AREA ARRAY USED BY IMSL ROUTINE GGAMA.
WINDSPD(3,3)		WEATHR		WINDSPD(I,J) IS THE PROBABILITY OF HAVING JTH WINDSPEED GIVEN PASQUILL ATMOSPHERIC STABILITY CATEGORY 2+1 OR 2+1-1.

GLOSSARY - CONTINUED

<u>VARIABLE</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
WSUM(2)		WEATHR		WSUM(1) IS THE SUM OF ELEMENTS W(1,1,1) THROUGH W(6,11,1). (USED TO NORMALIZE RANDOM NUMBERS BEFORE SAMPLING FROM W ARRAY).
X		(LOCAL)		USED AS INTERMEDIATE VARIABLE FOR CALLING FPRCNT.
XIN		(F.P.)		PERCENTAGE VALUE SENT TO FPRCNT FUNCTION FOR FORMATTING.
XMTTIM(2,2)		RSPTIM	SECONDS	XMTTIM(I,J) IS THE TIME REQUIRED TO TRANSMIT THE REQUEST FOR COPPERHEAD FIRE FROM DESIGNATOR OPERATOR TO FDC USING COMMUNICATION MODE I FOR A MISSION TYPE J. (I=1 FOR DIGITAL COMMUNICATION, I=2 FOR VOICE; J=1 FOR PRE-PLANNED TARGET, J=2 FOR TARGET OF OPPORTUNITY).
XMYADF(2)		RSPTIM	SECONDS	XMYADF(1) IS THE TIME REQUIRED TO TRANSMIT REQUEST FOR COPPERHEAD FIRE FROM DESIGNATOR OPERATOR TO FDC VIA VOICE AFTER TRYING DIGITAL (AND FAILING DIGITAL) FOR MISSION TYPE I. (I=1 FOR PRE-PLANNED TARGET, I=2 FOR TARGET OF OPPORTUNITY.)
XNV		(LOCAL)		NUMBER OF CURRENTLY UNKILLED VEHICLES IN TARGET.
XOUT		(LOCAL)		NAME OF VALUES OUTPUT AFTER SPECIAL FORMATTING.
XSMPL		(F.P.)		RANDOM SAMPLE VALUE RETURNED BY SUBROUTINE SMPLCD.
XVALUE(8,9,2)		XVALUE		XVALUE(I,J,1) IS THE ITH ALLOWABLE VALUE FOR THE JTH PARAMETER USED TO CONSTRUCT (OR DECIPHER) THE PEDATA RECORD NAME. XVALUE(I,J,2) IS THE CODE NUMBER CORRESPONDING TO XVALUE(I,J,1). (NOTE THAT NOT ALL POSSIBLE I,J COMBINATIONS NEED CORRESPOND TO VALUES ACTUALLY ALLOWED IN CURRENT CASES.)
XWTHR		(LOCAL)		ALPHANUMERIC ABBREVIATION FOR NAME OF MONTH WHOSE WEATHER DATA IS BEING USED.
Y		(LOCAL)		INTERMEDIATE VARIABLE NAME USED IN CALL TO FPRCNT.
YOUT		(LOCAL)		REFORMATTED PERCENTAGE VALUES.
TWTHR		(LOCAL)		ALPHANUMERIC TIME OF DAY VALUE FOR WEATHER DATA.
ZOUT		(LOCAL)		REFORMATTED PERCENTAGE VALUES.

CHAPTER 13

13. CONVERSION TO OTHER COMPUTERS

The COPE model and its preprocessors were written for use on the US Army Ballistic Research Laboratories Control Data Corporation CYBER 76 (CDC 7600) computer. These programs use some of the features that are peculiar to CDC machines. This chapter points out some of the changes to COPE and its preprocessors that will be required to convert them to run on other machines.

13.1 Conversion To Other CDC Computers.

COPE and its preprocessors have been run on other CDC machines (a CDC 6600 at Aberdeen Proving Ground and a CDC 6400 and a CDC 6500 at FT Leavenworth, KS).

Two changes were required. The first was to change the Hollerith string delimiters from '(single quote) to whatever the host machine uses to delimit Hollerith strings (" double quote on the FT Leavenworth machine, ~~≠~~ on machines using CDC's standard, * at some other installations). This includes strings on STOP statements.

The second change was to define the cotangent function in PAM. The BRL CDC machines have the cotangent (COT) function as a library function; some other machines do not. This can be easily fixed by defining $COT(X) = 1./TAN(X)$ as an in-line or statement function in PAM. Of course an alternate fix is to replace each occurrence of COT(X) by $(1./TAN(X))$.

Once these changes were made, the COPE program and its preprocessors ran without any trouble on the other CDC machines.

13.2 Conversion To Non-CDC Computers.

The author cannot anticipate every problem the user may encounter in converting COPE and its preprocessors to run on other machines; however, there are certain changes that will have to be made and of which the author is aware. The only machine (other than the CDC machines) that the author uses to any significant extent is the UNIVAC 1108, so most of the comments in this section deal with changes the author knows will have to be made to get the programs working on the UNIVAC machines.

13.2.1 Hollerith Delimiters and STOP's. The first change that may be required is to change the Hollerith string delimiters from ' to whatever is used on the host machine.

The CDC machines allow a STOP statement with a message (the message is in Hollerith string delimiters itself) which is printed when that STOP is encountered. Other machines may allow only a number with the stop, only a shorter message (CDC allows up to 70 characters; UNIVAC allows 6 characters), or nothing at all. So depending on the STOP

messages allowed, all of the STOP statements in COPE and its preprocessors may require changes.

13.2.2 A FORMATS and Alphanumeric Arrays. The CDC machines use a 60 bit word and a six bit character code so that each word holds 10 characters. Hence, most of the alphanumeric reads, writes, encodes, and decodes use A10 format and most data statements use 10H format to set alphanumeric values.

For machines that use shorter words or 8 bit character codes, the number of characters per word will probably be fewer. This means that the A10 formats in COPE and its preprocessors (and, indeed, all An formats where n > number of characters per word on the host machine) must be changed. A similar remark applies to all 10H (or nH with n as above) formats in data statements. Reducing these formats to the limits of the host machine will sometimes require that arrays containing the alphanumeric data be redimensioned. For example, the B array in subroutine INPUT is currently dimensioned B(8,10); if used on the UNIVAC 1108 which has 5 characters per word, it would have to be redimensioned B(14,10) so that each row could still contain a full card image (80 characters).

13.2.3 Word Addressable Mass Storage Files. The CDC system subroutines to open, read from, write to, and close a word addressable mass storage file are respectively OPENMS, READMS, WRITMS, and CLOSMS. The array INDEX11, the system subroutine SYSTEMC, and the user routines NOREC and NOREC2 are also connected with COPE's use of word addressable files.

The most convenient way to adapt COPE's handling of these files to the host machine is probably to write subroutines OPENMS, READMS, WRITMS, and CLOSMS that use the host machine's equivalent file type and commands, and which perform the same operations as on CDC.

In the specific case of the UNIVAC, for example, OPENMS would be a subroutine which first does a DEFINE FILE, then reads from the direct access file (using UNIVAC terminology -- this is equivalent to the CDC word addressable file) the current values of the INDEX11 array. Subroutine READMS would search the INDEX11 array for the desired record name and then when the name was found and the corresponding address v looked up, it would do a READ (11'v); if the record name were not found, NOREC would be called (or NOREC2 in PREPMS). Subroutine WRITMS would add a record name and address to INDEX11 (or perhaps just modify the addresses if the record name were already present) and then do a WRITE (11'v). CLOSMS would do a WRITE (11'v) of INDEX11 to TAPE11.

Non-CDC machines may also require redimensioning of INDEX11 to allow record names to take up two (or more) words of characters. Also additional bookkeeping of record addresses may be required in PREPMS.

The calls to SYSTEMC and the defining of IRAY can be dispensed with if error prints are added that occur when attempting to read a non-existent record.

13.2.4 ENCODE and DECODE. Although most computers with FORTRAN IV compilers have ENCODE and DECODE statement's, there are some that may not and there are others that differ from CDC's.

If the host machine does not have ENCODE and DECODE statements, it would probably be worthwhile to write subroutines that perform the functions of ENCODE and DECODE and then replace all ENCODES and DECODES in COPE with calls to those subroutines.

If the host machine has ENCODE and DECODE a check should be made to see that they take the same arguments in the same order as the CDC ENCODE and DECODE statements and, if not, the necessary changes should be made.

13.2.5 DATE and TIME. CDC has two subroutines DATE and TIME that respectively give an 8 character date and a 9 character time. These are only called to obtain date and time values to be printed on the title page of the output. If the user's computer does not have equivalent subroutines, the calls to DATE and TIME may be deleted.

13.2.6 STRACE. The subroutine STRACE causes a printing of traceback information from the current subroutine back to the main program. This information gives the subroutine and line from which STRACE was called, the subroutine and line from which that subroutine was called, and so on back to the main program.

STRACE is available on both CDC and UNIVAC but it may not exist in some FORTRAN implementations (or it may have a different name). In either case, the user will need to check this point.

13.2.7 EOF. The CDC machine has a function called EOF. The value of EOF (IUNIT) is zero if the end of file has not been encountered on I/O unit IUNIT. After a READ, a test of EOF will yield a non-zero value if the end-of-file on that I/O unit has been encountered.

The CDC statements:

```
READ (IUNIT, 100) iolist  
IF (EOF(IUNIT).NE.0.0) GO TO 99
```

are equivalent to:

```
READ (IUNIT, 100, END = 99) iolist
```

on most other machines.

13.2.8 COTANGENT. The PAM program uses the cotangent function (COT). If this is not a library function on the user's computer, then it should either be defined as a function (a statement function or external function) or else each occurrence of COT(X) should be replaced by (1./TAN(X)).

13.2.9 Im.n FORMAT. The CDC FORTRAN includes an Im.n format which specifies that an integer is to be written right justified in a field m characters wide with a minimum of n characters printed ($m \geq n$). This means that leading zeros are generated when the output value requires fewer than n digits. For example, if an integer variable with value 23 were to be written under I5.4 format, the result would be: b0023. Here b represents a blank so we have a field width of m=5 with n=4 characters printed even though 2 would have sufficed.

The Im.n is a standard format in FORTRAN 77, but since COPE is written in FORTRAN IV that may be of no help to the user whose FORTRAN IV compiler does not include it. On some machines a similar format (such as the UNIVAC Jn format) may be substituted; on other machines the user may have to simulate this format by using one of several different formats depending on the number of digits required to write the integer value.

13.3 GGAMA Function (IMSL).

The main COPE program uses the International Mathematical and Statistical Library (IMSL) function GGAMA. It is called from function GAMMA which is in turn called from GETTIM (it is used when the third method of playing response time is used).

If the IMSL is available on the user's computer, then it should be made available when COPE is loaded and run.

If the IMSL is not available to the user, then a function GGAMA should be written by the user or obtained from another library. It should take a random number seed IRN, an alpha parameter value (ALPHA), and return a gamma distributed random deviate R (with $\alpha = \text{ALPHA}$ and $\beta = 1$). The arguments IN and WK are arguments used by the IMSL routine and may or may not be used by the user's GGAMA.

CHAPTER 14

14. PREPMS PREPROCESSOR PROGRAM

14.1 General Remarks

The PREPMS preprocessor program performs three functions:

(1) to create or modify records on the word addressable data base file (TAPE 11) by reading input cards (or card images) and writing their contents (in appropriate format and order) to TAPE 11.

(2) to generate and print a list of the names of the current records on TAPE 11, and

(3) to generate and print a list of the PE data records currently on TAPE 11 along with their respective parameter values (see section 5.4 for the relation of 9 parameters to PE data record names).

Because the PREPMS program is quite straightforward in design and shares many subroutines with COPE, it is not documented as thoroughly as the main COPE program; however, this chapter gives enough information to set up the input, understand the output, and generally understand what the program does.

A re-reading of Chapter 5 may be helpful in understanding what follows.

14.2 PREPMS Program Design.

The PREPMS program consists of seven subprograms: the main program PREPMS, the block data subprogram BDATA2, the subroutines NOREC2, PENAME, and SEPRC2, and the functions IDCHAR and NUMRIC.

The subprograms PENAME, IDCHAR, and NUMRIC are identical to the subprograms of the same name in COPE. BDATA2 is merely a subset of BDATA1 of COPE; that is, it contains data statements that fill in the three common blocks LOGFLG, SYMBOL, and XVALUE as does BDATA1, but it does not fill in any of the other common blocks of BDATA1 as they are not needed by PREPMS.

The subroutine SEPRC2 is nearly identical to SEPREC of COPE. The only difference is that the message about sequence numbers in columns 73-78 of the input cards is written directly to OUTPUT instead of encoded and written via a call to COMMNT.

The subroutine NOREC2 is called (as a result of using CDC's SYSTEMC feature) whenever the program attempts to read a record named OPTNNUMS from TAPE 11 but does not find such a record. This should happen only when PREPMS is creating a new TAPE 11; when PREPMS is modifying an

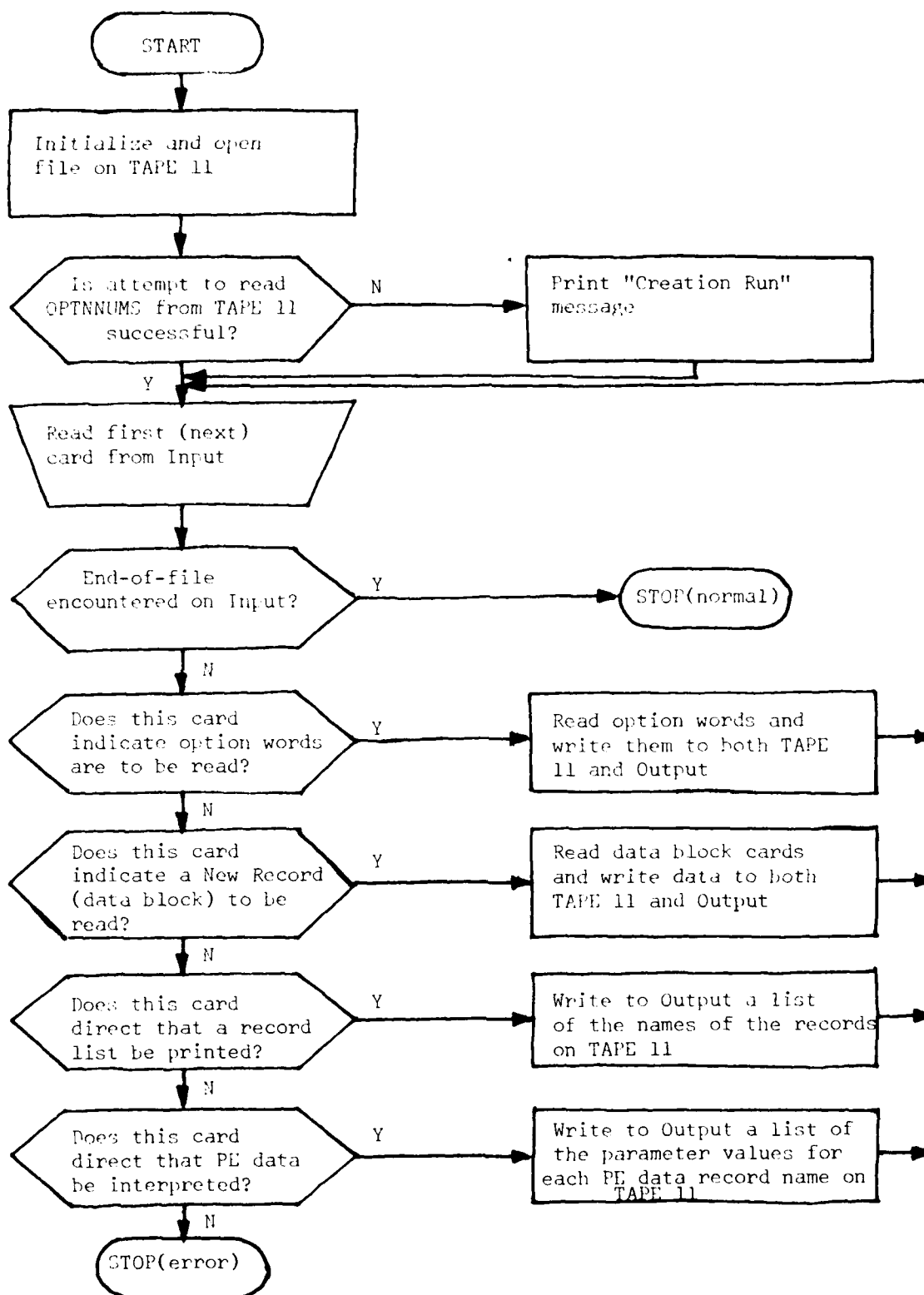


FIGURE 14-1 Logic Flow of PREPMS

existing TAPE11, there will already be an OPTNNUMS record on TAPE11. When no such record is found, NOREC2 is called; it then prints a message saying that the current job is a creation run and allows the program to continue.

The main program PREPMS is different from anything in COPE. Its main logic flow is shown in Figure 14-1. It begins by initializing (via DATA statements), calling SYSTEMC to set an error recovery procedure, and opening the file TAPE11. If it successfully reads the record named OPTNNUMS, it continues; if not, NOREC2 is called, a message is printed, and execution resumes. Next it attempts to read a card; if successful, the program continues; if unsuccessful, it has hit the end of file and a stop with normal termination message is issued.

The main program examines the card just read to determine what action to take next. Five possibilities exist:

(1) The card indicates a set of option words follows. In this case, the option words are read, printed on OUTPUT and written to TAPE 11.

(2) The card indicates a new record is to be read. In this case, the data for the new data block record is read, printed on OUTPUT, and written to TAPE11.

(3) The card indicates a list of the names of the records currently on TAPE11 is desired. In this case, such a list is generated and printed on OUTPUT.

(4) The card indicates PE data record names are to be interpreted. In this case, the names of the PE data records currently on TAPE11 are examined and the corresponding parameter values are determined. A list is then printed giving each PE data record's name and the corresponding parameter values.

(5) None of the above in which case a stop with error message occurs.

Note: The terms INPUT and OUTPUT refer respectively to the files TAPE5 and TAPE6 which correspond to I/O units 5 and 6. TAPE11 corresponds to I/O unit 11 and is another name for the word addressable data base file. Note that OUTPUT (or TAPE6) normally corresponds to a line printer and INPUT (or TAPE5) corresponds to either a punched card reader or a mass storage file of card images.

14.3 Glossary of PREPMS Variables.

The majority of the PREPMS variables have the same meaning as their namesakes in the COPE program. Therefore, the glossary in this section gives definitions only for those variables different from the main COPE model.

GLOSSARY OF PREPMS VARIABLES

<u>VARIABLES</u>	<u>TYPE</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
CLABEL	REAL	(LOCAL)		THE RECORD NAME OF THE OPTION WORD RECORD JUST READ. NORMALLY THIS WILL BE THE ABBREVIATED FORM OF THE KEYWORD TO WHICH THE OPTION WORDS BELONG.
ICASE	INTEGER	(LOCAL)		THE TYPE NUMBER OF THE DATA BLOCK RECORD JUST READ. ICASE = 1 FOR WEATHER DATA, 2 FOR ACQUISITION DATA, 3 FOR RESPONSE TIME DATA, 4 FOR DIRECT FIRE SUPPRESSION DATA, 5 FOR RANDOM OCCURRENCE DATA, 6 FOR PEDATA, 7 FOR POSTURE DISTRIBUTION DATA, 8 FOR INVARIANT DATA, AND 9 FOR SADARM DATA (9 IS NOT USED IN RUNNING PREPMS FOR COPE).
ILIST(2000)	INTEGER	(LOCAL)		ILIST (21-1) IS THE NAME OF THE ITH RECORD ON TAPE 11 ILIST (21) IS A NUMBER (PRINTED IN OCTAL) GIVING THE LENGTH AND ADDRESS ON DISK (PACKED) OF THE ITH RECORD ON TAPE 11.
IOUT	INTEGER	(LOCAL)		IOUT IS A COUNTER USED TO RECORD THE NUMBER OF ILIST ENTRIES TO BE PRINTED.
IPECOD	INTEGER	(LOCAL)		IPECOD IS THE NAME OF THE PE DATA RECORD CURRENTLY BEING EXAMINED TO DETERMINE PARAMETER VALUES.
IRLT1, IRLT2, ..., IRLT9	INTEGER	(LOCAL)	WORDS	IRLT1 IS THE RECORD LENGTH OF TYPE 1 DATA BLOCK RECORDS (WEATHER DATA); IRLT2 IS THE RECORD LENGTH OF TYPE 2 DATA BLOCK RECORDS (ACQUISITION DATA); - - -; IRLT9 IS THE RECORD LENGTH OF TYPE 9 DATA BLOCK RECORDS (SADARM DATA).
IUNIT	INTEGER	(LOCAL)		I/O UNIT FROM WHICH PE DATA IS TO BE READ (MUST BE EITHER 4 OR 5).
I2	INTEGER	(LOCAL)		DESIGNATOR TYPE NUMBER (1 = GLLD, 2 = MULE, 3 = LTD).
NOPTNM(70)	INTEGER	(LOCAL)		NOPTNM(I) IS THE NUMBER OF OPTION WORD CHARACTER STRINGS CURRENTLY AVAILABLE FOR USE WITH KEYWORD NUMBER 1.
NOPTI	INTEGER	(LOCAL)		NUMBER OF THE KEYWORD CORRESPONDING TO THE SET OF OPTION WORDS JUST READ.
NREAD	INTEGER	(LOCAL)		NUMBER OF OPTION WORD CHARACTER STRINGS TO BE READ IN THIS SET OF OPTION WORDS.

GLOSSARY OF PREPMS VARIABLES - CONTINUED

<u>VARIABLES</u>	<u>TYPL</u>	<u>COMMON BLOCK</u>	<u>UNITS</u>	<u>DEFINITION</u>
OPT	REAL	(LOCAL)		TEMPORARY NAME FOR THE VALUE TO BE LOADED INTO CHARA (1,1J). IT IS THE NUMBER OF THE OPTION FOR ITS OPTION WORD POSITION (GIVEN BY CHARA (2,1J) OR XLEV) TO WHICH THE CHARACTER STRING TO FOLLOW BELONGS.
PECHK	REAL	(LOCAL)		PECHK IS THE FIRST FOUR CHARACTERS (LEFT JUSTIFIED WITH BLANK FILL) OF THE NAME OF THE RECORD ON TAPE11 CURRENTLY BEING EXAMINED TO DETERMINE WHETHER IT IS A PC DATA RECORD.
RLABEL	REAL	(LOCAL)		NAME OF CURRENT DATA BLOCK RECORD.
T1, T3, T4, T5, T6, T7, T8, T9	REAL	(LOCAL)		THESE CORRESPOND RESPECTIVELY TO THE FIRST, THIRD, FOURTH, ----, NINTH ARGUMENTS OF SUBROUTINE PENAME (AND ITS ENTRY PEJUNT) T1 IS NOMINAL RESPONSE TIME, T3 IS TARGET VELOCITY, T4 IS GUN-TARGET RANGE, T5 IS TARGET REFLECTIVITY, T6 IS ANGLE T, T7 IS DEFLECTION BIAS, T8 IS TARGET HEADING, AND T9 IS SEEKER SENSITIVITY.
X	REAL	(LOCAL)		TEMPORARY NAME FOR VALUE TO BE LOADED INTO HREAD (DEFINED ABOVE).
XLEV	REAL	(LOCAL)		XLEV IS TEMPORARY NAME FOR VALUE TO BE LOADED INTO CHARA(2,1J). IT IS THE OPTION WORD NUMBER FOR WHICH THE CHARACTER STRING TO FOLLOW IS A POSSIBLE CHOICE.
Y	REAL	(LOCAL)		Y IS A TEMPORARY NAME FOR THE VALUE TO BE LOADED INTO NOPT1 (DEFINED ABOVE).
ZCASE	REAL	(LOCAL)		TEMPORARY NAME FOR VALUE TO BE LOADED INTO ICASE (DEFINED ABOVE).

To find the definition of a variable in PREPMS, one should first look in the glossary of this section. Then, if the variable is not found, look in the glossary of COPE variables in Chapter 12.

Note that all variables in common block SADARB are used only with the SADARM variant of COPE (called SOPE) and are not defined herein. This remark also applies to the WPNDAT array.

14.4 PREPMS INPUTS.

The PREPMS inputs are divided into four types:

- (1) option word records
- (2) data block records
- (3) REPORT RECORD LIST directives, and
- (4) INTERPRET PEDATA directives.

The second type (data block records) is further divided into eight types of data blocks. The sections concerning data block record types (sections 14.4.2(1) through 14.4.2(8)) also give instructions for setting up data to be read by COPE using the TEMPORARY feature (see section 4.5).

Note that if two or more option word records are read corresponding to the same keyword (see section 14.4.1) then only the last one read will have any effect. Similarly, if two or more data block records are read under the same record name, only the last one read will be retained on TAPE11. This means that any modifications to either type 1 or type 2 input (as listed above) require that the entire desired record contents be read (not just the additions).

Also, note that inputs of the four types may be arranged in any order provided that the integrity of the multiple card (or card image) input types (types 1 and 2) is preserved. However, types 3 and 4 only have effect on the file (TAPE11) as it exists at the time they are encountered; hence, they are usually put at the end of the input.

14.4.1 Option Word Records. Each option word record consists of one card (or card image) of type 1 and NREAD cards of type 2 following the type 1 card.

The general form of the type 1 card is:

OPTION WORDS, keyword, m, n

or

OPTWDS, keyword, m, n

where: keyword is replaced by the character string giving the abbreviated form of the keyword to which the option words are to apply;

m = NREAD, a positive integer giving the number of type 2 cards to be read for this option word; and

n = NOPT1, a positive integer giving the keyword number (see section 5.3.9) of the keyword to which the option words are to apply.

The general form of each type 2 card is:

mm, nn, Option word character string

where: mm = NOPTWC, the number of the option word choice to which the character string belongs (NOPTWC is a positive integer);

nn = NOPTW, a positive integer equal to one plus the number of the option word for which the character string is an allowed choice; and

Option word character string is the actual character string allowed as a choice.

The subsequent use of option word character string in a COPE run as a choice for option word NOPTW-1 on a keyword input card for keyword number NOPT1 causes option number NOPTWC to be activated for option word NOPTW-1. (See example in section 5.3.8; for a second example, see appendixes C and D.)

Note: While blanks may be freely inserted in COPE inputs (keywords and option words) with no effect (the single exception being blanks occurring within 10 spaces after a "\$" - see section 4.5.3), there should be no internal blanks in the character strings on type 2 cards.

Also, no more than 2 option word character strings applying to the same keyword are allowed to correspond to a given NOPTWC, NOPTW pair.

Finally, no more than 30 type 2 cards are allowed with any type 1 card.

14.4.2 Data Block Records. There are eight data block record types:

- (1) Weather data
- (2) Acquisition data
- (3) Response time data
- (4) Direct Fire Suppression data
- (5) Random Occurrence data
- (6) PE (probability of engagement data)
- (7) Target Posture Distribution data
- (8) Invariant data

The formatting of the input required for each of these types is presented in a separate subsection of this section.

Note that these eight data blocks correspond exactly to the eight data blocks that can be used with the TEMPORARY option in COPE. Furthermore, the formats used by COPE under the TEMPORARY option and by PREPMS are identical except for the first card (or card image) of each set. The sections that follow give the set-ups required for use with either program.

The first card (header card) in each case is free format: blanks may be freely inserted, fields are defined by separators (commas, for example). Subsequent cards of each data block type are formatted FORTRAN inputs and care must be taken to conform to the specified formats.

1. Weather Data Block Input.

When reading weather data into PREPMS for transfer onto TAPE11, the first card (which indicates a weather data record is to be input) must be of the form:

NEW RECORD, record name, 1

or

NEW REC, record name, 1

where record name is the name of the weather data record to be read. (See section 5.3.1 for the form of weather data record names).

When using the TEMPORARY option with COPE, the weather data is headed by a card of the form:

TEMPORARY, WEATHER or TEMPORARY, W

or

TEMP, WEATHER or TEMP, W

In either case (whether reading weather data in PREPMS or COPE), the remaining cards of the weather data are as follows:

WEATHER DATA

CARD SET

Card No. 2

Variable name	UNITS	Format	columns	Description
NCC		15	1-5	Number of different cloud ceiling altitude values used with the current weather data. (If NCC >6, then array dimension changes are required).
NVL		15	6-10	Number of different visibility range limits used with the current weather data. (If NVL ≠ 11, then array dimension changes and program modifications are usually required).

WEATHER DATA

CARD SET

Card No. 3

Variable name	UNITS	Format	columns	Description
PRCLCG		F10.4	1-10	Probability that there is a cloud ceiling.
PRCFLS(1)		F10.4	11-20	Probability that there is a cloud free line-of-sight given that there is a cloud ceiling.
PRCFLS(2)		F10.4	21-30	Probability that there is a cloud free line-of-sight given that there is no cloud ceiling (i.e., scattered clouds only).
			14-10	

WEATHER DATA

CARD SET

Card No. 4^a

Variable name	UNITS	Format	columns	Description
PRGCFL(1,1)		F10.4	1-10	PRGCFL(I,1) is the probability that the visibility range limit is equal to I kilometers given that there is a cloud ceiling and a cloud free line-of-sight. Note: The visibility range limit is considered to be I kilometers if $(I-1/2)\text{km} < \text{visibility range limit} < (I+1/2)\text{km}$.
PRGCFL(2,1)		F10.4	11-21	
⋮				
PRGCFL(8,1)		F10.4	71-80	
			14-11	

Card No. 4b

Variable name	UNITS	Format	columns	Description
PRGCFL(9,1)		F10.4	1-10	PRGCFL(I,1) is described on previous page.
PRGCFL(11,1)		F10.4	21-30	

WEATHER DATA

CARD SET

Card No. 5a

Variable name	UNITS	Format	columns	Description
PRGCFL(1,2)		F10.4	1-10	PRGCFL(I,2) is the probability that the visibility range limit is equal to I kilometers given that there is no cloud ceiling and that there is a cloud free line-of-sight. The note on card 4a applies.
PRGCFL(2,2)		F10.4	11-20	
:				
:				
PRGCFL(8,2)		F10.4	71-80	

CARD SET

Card No. 5b

14-14

CARD SET

Card No. 6a

Variable name	UNITS	Format	columns	Description
W(1,1,1)		F10.4	1-10	W(I,J,1) is the probability of having the Ith cloud ceiling altitude and the Jth visibility range limit given that there is a cloud ceiling but no cloud free line-of-sight.
W(2,1,1)		F10.4	11-20	
W(3,1,1)		F10.4	21-30	
W(4,1,1)		F10.4	31-40	
W(5,1,1)		F10.4	41-50	
W(6,1,1)		F10.4	51-60	

WEATHER DATA CARD SET

Card No. 6b

Variable name	UNITS	Format	columns	Description
W(1,2,1)		F10.4	1-10	W(I,J,1) is as defined on previous page.
W(2,2,1)		F10.4	11-20	
W(3,2,1)		F10.4	21-30	
W(4,2,1)		F10.4	31-40	
W(5,2,1)		F10.4	41-50	
W(6,2,1)		F10.4	51-60	
			14-16	

Cards 6c through 6k are similar to cards 6a and 6b except each successive card contains $W(I,J,1)$ values for a higher value of J :

card 6a	contains	$W(1,1,1)$	through	$W(6,1,1)$
" 6b	"	$W(1,2,1)$	"	$W(6,2,1)$
" 6c	"	$W(1,3,1)$	"	$W(6,3,1)$
" 6d	"	$W(1,4,1)$	"	$W(6,4,1)$
" 6e	"	$W(1,5,1)$	"	$W(6,5,1)$
" 6f	"	$W(1,6,1)$	"	$W(6,6,1)$
" 6g	"	$W(1,7,1)$	"	$W(6,7,1)$
" 6h	"	$W(1,8,1)$	"	$W(6,8,1)$
" 6i	"	$W(1,9,1)$	"	$W(6,9,1)$
" 6j	"	$W(1,10,1)$	"	$W(6,10,1)$
" 6k	"	$W(1,11,1)$	"	$W(6,11,1)$

Each of these cards contains six values,

$\{W(1,J,1), W(2,J,1), W(3,J,1), W(4,J,1), W(5,J,1), \text{ and } W(6,J,1)\}$,

in 6F10.4 format (exactly as illustrated for cards 6a and 6b on the previous two pages).

CARD SET

Card No. 7a

Variable name	UNITS	Format	columns	Description
W(1,1,2)		F10.4	1-10	W(I,J,2) is the probability of having the Ith cloud ceiling altitude and the Jth visibility range limit given that there is no cloud ceiling and no cloud free line-of-sight.
W(2,1,2)		F10.4	11-20	
W(3,1,2)		F10.4	21-30	
W(4,1,2)		F10.4	31-40	
W(5,1,2)		F10.4	41-50	
W(6,1,2)		F10.4	51-60	

CARD SET

Variable name	UNITS	Format	columns	Description
W(1,2,2)		F10.4	1-10	W(I,J,2) is as defined on previous page.
W(2,2,2)		F10.4	11-20	
W(3,2,2)		F10.4	21-30	
W(4,2,2)		F10.4	31-40	
W(5,2,2)		F10.4	41-50	
W(6,2,2)		F10.4	51-60	

Cards 7c through 7k are similar to cards 7a and 7b except each successive card contains $W(I,J,2)$ values for a higher value of J :

card 7a	contains	$W(1,1,2)$	through	$W(6,1,2)$
" 7b	"	$W(1,2,2)$	"	$W(6,2,2)$
" 7c	"	$W(1,3,2)$	"	$W(6,3,2)$
" 7d	"	$W(1,4,2)$	"	$W(6,4,2)$
" 7e	"	$W(1,5,2)$	"	$W(6,5,2)$
" 7f	"	$W(1,6,2)$	"	$W(6,6,2)$
" 7g	"	$W(1,7,2)$	"	$W(6,7,2)$
" 7h	"	$W(1,8,2)$	"	$W(6,8,2)$
" 7i	"	$W(1,9,2)$	"	$W(6,9,2)$
" 7j	"	$W(1,10,2)$	"	$W(6,10,2)$
" 7k	"	$W(1,11,2)$	"	$W(6,11,2)$

Each of these cards contains six values,

$\{W(1,J,2), W(2,J,2), W(3,J,2), W(4,J,2), W(5,J,2), \text{ and } W(6,J,2)\},$

in 6F10.4 format (exactly as illustrated for cards 7a and 7b on the previous two pages).

WEATHER DATA

CARD SET

Card No. 8a

Variable name	UNITS	Format	columns	Description
VISLIM(1)	km.	F10.4	1-10	VISLIM (I) is the Ith visibility range limit value. Any choice other than VISLIM(I) = 1 kilometers will require program changes to COPE.
VISLIM (2)	km.	F10.4	11-20	
VISLIM (3)	km.	F10.4	21-30	
'	'	'		
'	'	'		
'	'	'		
'	'	'		
VISLIM (8)	km.	F10.4	71-80	
			14-21	

WEATHER DATA

CARD SET

Card No. 8b

Variable name	UNITS	Format	columns	Description
VISLIM (9)	km.	F10.4	1-10	VISLIM (I) is as described on previous page.
'	'	'		
'	'	'		
'	'	'		
VISLIM (11)	km.	F10.4	21-30	

WEATHER DATA CARD SET

Card No. 9

Variable name	UNITS	Format	columns	Description
CLOUD (1)	meters	F10.4	1-10	CLOUD (I) is the Ith cloud ceiling altitude in meters used for the current set of weather data.
CLOUD (2)	meters	F10.4	11-20	
CLOUD (3)	meters	F10.4	21-30	
CLOUD (4)	meters	F10.4	31-40	
CLOUD (5)	meters	F10.4	41-50	
CLOUD (6)	meters	F10.4	51-60	
			14-23	

WEATHER DATA CARD SET

Card No. 10

Variable name	UNITS	Format	columns	Description
PASQL (1)		F10.4	1-10	PASQL(I) is the probability of occurrence of the Ith Pasquill atmospheric stability category. (Category A corresponds to I=1, B to I=2, etc.)
PASQL (2)		F10.4	11-20	
PASQL (3)		F10.4	21-30	
PASQL (4)		F10.4	31-40	
PASQL (5)		F10.4	41-50	
PASQL (6)		F10.4	51-60	
			14-24	

Card No. 11

Variable name	UNITS	Format	columns	Description
WNDS PD (1)		F10.4	1-10	WNDS PD (I) is the probability that the windspeed is in speed bracket I. (Speed bracket I=1 includes windspeeds from 0 to 5 knots, I=2 includes from 5 to 15 knots and I=3 includes 15 knots and over).
WNDS PD (2)		F10.4	11-20	
WNDS PD (3)		F10.4	21-30	

WEATHER DATA CARD SET

Card No. 12

Variable name	UNITS	Format	columns	Description
HUMID (1)		F10.4	1-10	HUMID (1) is the probability that the relative humidity is less than 65%; HUMID (2) is the probability that the relative humidity is greater than (or equal to) 65%.
HUMID (2)		F10.4	11-20	

2. Acquisition Data Block Input.

When reading acquisition data into PREPMS for transfer onto TAPE11, the first card (which indicates an acquisition data record is to be read) must be of the form:

NEW RECORD, record name, 2

or

NEW REC, record name, 2

where record name is the name of the acquisition data record to be read (see section 5.3.2 for the form of acquisition data record names).

When using the TEMPORARY option with COPE, the acquisition data is headed by a card of the form:

TEMPORARY, ACQUISITION RANGE DISTRIBUTION

or

TEMPORARY, ACQRNGDIST

or

TEMP, ACQUISITION RANGE DISTRIBUTION

or

TEMP, ACQRNGDIST

In either case (whether reading acquisition data in PREPMS or COPE), the remaining cards of the acquisition data are as follows:

ACQUISITION DATA CARD SET

Card No. 2

Variable name	UNITS	Format	columns	Description
NRP		I5	1-5	Number of range points used in the CRNGD array (Cumulative distribution of acquisition (unmasking) ranges.)
NPP		I5	5-10	Number of points used within each range class to describe cumulative distribution of line-of-sight segment lengths.
NRNGCL		I5	11-15	Number of range class boundary value to be used (equals one less than the number of range classes) Note: Zero is always assumed to be the lower boundary of the shortest range class so it need not be explicitly entered as input nor counted as a boundary value for NRNGCL.

ACQUISITION DATA

CARD SET

Card No. 3a

Variable name	UNITS	Format	columns	Description
CRNGD(1,1)		F10.4	1-10	CRNGD(1,1) is the probability that the designator-to-target range at time of target unmasking is less than or equal to the value in CRNGD(1,2)
CRNGD(2,1)		F10.4	11-20	
CRNGD(3,1)		F10.4	21-30	<u>Note:</u> CRNGD(1,1) = 0.0 and CRNGD(NRP,1) = 1.0 must be satisfied. Also, the entries must increase as I increases.
/		/	/	
/		/	/	
/		/	/	
/		/	/	
CRNGD(8,1)		F10.4	71-80	<u>Note:</u> If $NRP \leq 8$, only one card of this type will be needed; if $NRP > 8$, then a second card must continue with CRNGD(9,1) through CRNGD(NRP,1). NRP cannot exceed 11 with current dimensions.

ACQUISITION DATA CARD SET

Card No. 3b

Variable name	UNITS	Format	columns	Description
CRNGD(1,2)	meters	F10.4	1-10	CRNGD(I,2) is the range value corresponding to the cumulative probability value in CRNGD(I,1). The values must increase as I increases.
CRNGD(2,2)	meters	F10.4	11-21	
CRNGD(8,2)	meters	F10.4	71-80	<u>Note:</u> If $NRP \leq 8$, only one card of this type will be needed; if $NRP > 8$, then a second card must continue with CRNGD(9,2) through CRNGD(NRP,2) NRP cannot exceed 11 with current dimensions.

ACQUISITION DATA CARD SET

Card No. 4a

Variable name	UNITS	Format	columns	Description
SEGLOS(1,1)		F10.4	1-10	SEGLOS(I,1) is the probability that the length of a line-of-sight segment in range class K is less than or equal to SEGLOS(I, K+1) meters. <u>Note:</u> SEGLOS(1,1) = 0.0 SEGLOS(NPP,1) = 1.0 and SEGLOS(I,1) < SEGLOS(I+1,1) for I = 1,2, --- NPP-1. If NPP < 8, only one card of this type will be needed; if NPP > 8, then a second card must continue with SEGLOS (9,1) through SEGLOS(NPP,1). NPP cannot exceed 11 with current dimensions.
SEGLOS(2,1)		F10.4	11-20	
/		/		
/		/		
/		/		
/		/		
SEGLOS(8,1)		F10.4	71-80	

ACQUISITION DATA CARD SET

Card No. 4b

Variable name	UNITS	Format	columns	Description
SEGLOS(1,2)	meters	F10.4	1-10	SEGLOS(I,2) is the line-of-sight segment length value corresponding to the cumulative probability value in SEGLOS(I,1) when the range from D.O. to target unmask falls in the first range class. <u>Note:</u> SEGLOS(I,2) < SEGLOS(I+1,2) for I = 1,2,---, NPP-1. <u>Note:</u> If NPP ≤ 8, only one card of this type is needed; if NPP > 8, then a second card must continue with SEGLOS(9,2) through SEGLOS(NPP,2). NPP cannot exceed 11 with current dimensions.
SEGLOS(2,2)	meters	F10.4	11-20	
SEGLOS(2,3)	meters	F10.4	21-30	
/		/	/	
/		/	/	
/		/	/	
/		/	/	
/		/	/	
/		/	/	
/		/	/	

The SEGLOS array is read from additional cards similar to 4b up to a maximum of 11 cards (or card pairs if NPP >8).

Each card (or card pair if NPP >8) has the following content:

card 4a	contains	SEGLOS(1,1)	through	SEGLOS(NPP,1)
" 4b	"	SEGLOS(1,2)	"	SEGLOS(NPP,2)
" 4c	"	SEGLOS(1,3)	"	SEGLOS(NPP,3)
" 4d	"	SEGLOS(1,4)	"	SEGLOS(NPP,4)
" 4e	"	SEGLOS(1,5)	"	SEGLOS(NPP,5)
" 4f	"	SEGLOS(1,6)	"	SEGLOS(NPP,6)
" 4g	"	SEGLOS(1,7)	"	SEGLOS(NPP,7)
" 4h	"	SEGLOS(1,8)	"	SEGLOS(NPP,8)
" 4i	"	SEGLOS(1,9)	"	SEGLOS(NPP,9)
" 4j	"	SEGLOS(1,10)	"	SEGLOS(NPP,10)
" 4k	"	SEGLOS(1,11)	"	SEGLOS(NPP,11)

If NPP >8 then each card above is replaced by a card pair. In this case, the first card of each pair contains eight values and the second card contains NPP-8 values (each card is in 8F10.4 format). SEGLOS(1,J) through SEGLOS(NPP,J) contain (for J ≥ 2) the line-of-sight segment lengths in meters for range class J-1 corresponding respectively to the cumulative probability values in SEGLOS(1,1) through SEGLOS(NPP,1). The total number of cards (or card pairs if NPP >8) will be NRRGCL + 1 (one for probabilities plus one for each range class). Hence, if NRRGCL = 10, all cards 3a through 3k will be used; but, if NRRGCL < 10, then only the first NRRGCL + 1 cards (or card pairs if NPP >8) shown above will be used.

ACQUISITION DATA

CARD SET

Card No. 5

Variable name	UNITS	Format	columns	Description
RNGCLB(1)	meters	F10.4	1-10	RNGCLB(I) is the Ith non-zero range class boundary. That is, range class 1 runs from zero to RNGCLB(1) meters and, for $I > 1$, range class I runs from RNGCLB(I-1) meters to RNGCLB(I) meters. (When the DO-to-target unmask range lies in range class I, then the (I + 1)st row of the SEGLOS array is sampled to obtain LOS segment lengths.) Note that $0 < \text{RNGCLB}(1) < \dots < \text{RNGCLB}(I) < \dots < \text{RNGCLB}(I+1) < \dots < \infty.$ Also note that if NRRGCL < 8, then only one card of this type is needed; but, if NRRGCL > 8, then a second card must continue with RNGCLB(9) through RNGCLB(NRRGCL). Finally, RNGCLB cannot exceed 10 with current dimensioning.
RNGCLB(2)	meters	F10.4	11-20	
/	/	/	/	
/	/	/	/	
/	/	/	/	
RNGCLB(8)	meters	F10.4	71-80	

ACQUISITION DATA CARD SET

Card No. 6a

Variable name	UNITS	Format	columns	Description
VELTBL(1,1)	M/S	F10.4	1-10	VELTBL(I,J) gives the target velocity default value in the current terrain under day/night condition I and weather condition J. I = 1 for day time 2 for nighttime J = 1 for "good" weather 2 for "bad" weather
VELTBL(2,1)	M/S	F10.4	11-20	

ACQUISITION DATA CARD SET

Card No. 6b

Variable name	UNITS	Format	columns	Description
VELTBL(1,2)	M/S	F10.4	1-10	(see previous page)
VELTBL(2,2)	M/S	F10.4	11-20	

3. Response Time Data Block Input.

When reading response time data into PREPMS for transfer to TAPE11, the first card (which indicates that a response time data record is to be read) must be of the form:

NEW RECORD, record name, 3

or

NEW REC, record name, 3

where record name is the name of the response time record to be read (see section 5.3.3. for response time record names).

When using the TEMPORARY option with COPE, the response time data is headed by a card of the form:

TEMPORARY, RESPONSE TIME

or

TEMPORARY, RESPTIME

or

TEMP, RESPONSE TIME

or

TEMP, RESPTIME

Note: When the first method (section 2.2.9) of modeling delay times in COPPERHEAD mission communication and processing is played, in COPE, the response time distribution used is the one defined by data statements in BDATA1 (namely, the FSRT array - see section 4.7.6).

Because this array is sampled to obtain response times, the values on cards 4, 5, and 6 of this card set are not used. However, cards must still be input (though they may be blank) so that the statements which read the later cards of this set will pick up the correct data.

The record name used with the data for the first method of playing response time must be RSPDAT0123 unless one makes code changes in COPE to allow for others.

In any case (whether reading response time data in PREPMS or COPE), the remaining cards of the response time data are as follows:

RESPONSE TIME CARD SET

Card No. 2

Variable name	UNITS	Format	columns	Description
NDT		I5	1-5	Number of points to be entered in each row of DETTMA array. NDT ≤ 10 with current dimensions.

RESPONSE TIME CARD SET

Card No. 3a

Variable name	UNITS	Format	columns	Description
DETTMA (1,1)		F10.4	1-10	DETTMA (I,1) is the probability that the "detection" time is less than or equal to DETTMA(I,2) seconds.
DETTMA (2,1)		F10.4	11-20	
:		:		Note: DETTMA (1,1) = 0.0 and DETTMA (I) < DETTMA (I +1)
:		:		
DETTMA(8,1)		F10.4	71-80	Also note that if NDT < 8, only one card of this type is required; if NDT > 8, then a second card is required with DETTMA (9,1) through DETTMA (NDT,1).

RESPONSE TIME CARD SET

Card No. 3b

Variable name	UNITS	Format	columns	Description
DETTMA (2,1)	seconds	F10.4	1-10	DETTMA (I,2) is the "detection" time in seconds corresponding to DETTMA (I,1). Note: DETTMA (I) < DETTMA (I+1) Also, if $NDT \leq 8$ only one card is needed; if $NDT > 8$, then DETTMA (9,2) through DETTMA (NDT,2) continue on a second card.
DETTMA (2,2)	seconds	F10.4	11-20	
:	:	:	:	
:	:	:	:	
DETTMA (8,2)	seconds	F10.4	71-80	

RESPONSE TIME CARD SET

Card No. 4

Variable name	UNITS	Format	columns	Description
BATRTM (1)	seconds	F10.4	1-10	BATRTM (I) is the response time at the battery for a Copperhead mission of type I.
BATRTM (2)	seconds	F10.4	11-20	I = 1 for preplanned targets
BATRTM (3)	seconds	F10.4	21-30	I = 2 for targets of opportunity
				I = 3 for priority preplanned targets
XMVADF (1)	seconds	F10.4	31-40	XMVADF (I) is the time required to transmit message via voice comms after failing to transmit via digital comms.
XMVADF (2)	seconds	F10.4	41-50	I = 1 for pre-planned targets (priority or not)
				I = 2 for targets of opportunity
				<i>Note:</i> The first way of modeling delay times does not use the data on cards 4, 5, and 6; hence, they may be left blank.
				The second way of modeling delay times does not use any RESPONSE TIME data block values, so no record is prepared corresponding to the PARAM option.
				Finally, the third delay time model uses all data in the data block.
				See sections 2.2.9, 4.4.15 and 9.23 for further details on the three delay time models.
			14-41	

RESPONSE TIME

CARD SET

Card No. 5

Variable name	UNITS	Format	columns	Description
BCSPTM (1,1)	seconds	F10.4	1-10	BCSPTM (I,J) is the battery computer system processing time for commo type I and mission type J. I = 1 for digital commo I = 2 for voice commo J = 1 for pre-planned target (priority or not) J = 2 for target of opportunity
BCSPTM (2,1)	seconds	F10.4	11-20	
BCSPTM (1,2)	seconds	F10.4	21-30	
BCSPTM (2,2)	seconds	F10.4	31-40	
XMTTIM (1,1)	seconds	F10.4	41-50	XMTTIM (I,J) is the commo transmission time for commo type I and mission type J where I and J are as for BCSPTM above.
XMTTIM (2,1)	"	F10.4	51-60	
XMTTIM (1,2)	"	F10.4	61-70	
XMTTIM (2,2)	"	F10.4	71-80	

RESPONSE TIME

CARD SET

Card No. 6

Variable name	UNITS	Format	columns	Description
TMEAN (1,1)	seconds	F10.4	1-10	TMEAN (I,J) is the mean time for DO to enter data in the digital message device in day/night condition I and mission type J. I = 1 for day time I = 2 for night J = 1 for pre-planned target (priority or not) J = 2 for target of opportunity
TMEAN (2,1)	seconds	F10.4	11-20	
TMEAN (1,2)	seconds	F10.4	21-30	
TMEAN (2,2)	seconds	F10.4	31-40	
TSIGMA (1,1)	seconds	F10.4	41-50	TSIGMA (I,J) is the standard deviation of the distribution of times required for the DO to enter data in digital message device under day/night condition I and mission type J. I and J are as defined above for TMEAN.
TSIGMA (2,1)	seconds	F10.4	51-60	
TSIGMA (1,2)	seconds	F10.4	61-70	
TSIGMA (2,2)	seconds	F10.4	71-80	

RESPONSE TIME

CARD SET

Card No. 7

Variable name	UNITS	Format	columns	Description
TRARRY (1,1)	seconds	F10.4	1-10	TRARRY (I,J) is the nominal response time for mission type I and commo type J. I = 1 for pre-planned targets I = 2 for targets of opportunity I = 3 for priority pre-planned targets. J = 1 for digital commo J = 2 for voice commo Note: These nominal response times are not to include time-of-flight. (TOF is added in COPE itself to obtain TRBAR).
TRARRY (2,1)	seconds	F10.4	11-20	
TRARRY (3,1)	seconds	F10.4	21-30	
TRARRY (1,2)	seconds	F10.4	31-40	
TRARRY (2,2)	seconds	F10.4	41-50	
TRARRY (3,2)	seconds	F10.4	51-60	

4. Direct Fire Suppression Data Block Input.

When reading direct fire suppression data into PREPMS for transfer to TAPE 11, the first card (which indicates that a direct fire suppression data record is to be read) must be of the form:

NEW RECORD, record name, 4

or

NEW REC, record name, 4

where record name is the name of the direct fire suppression record to be read (see section 5.3.4 for direct fire suppression record names).

When using the TEMPORARY option with COPE, the direct fire suppression data is headed by a card of the form:

TEMPORARY DIRECT FIRE SUPPRESSION

or

TEMPORARY, DFIRE SUPPR

or

TEMP, DIRECT FIRE SUPPRESSION

or

TEMP, DFIRE SUPPR

In either case (whether reading direct fire suppression data in PREPMS or COPE), the remaining cards of the direct fire suppression data are as follow:

DIRECT FIRE SUPPRESSION CARD SET

Card No. 2

Variable name	UNITS	Format	columns	Description
NDFSP		I5	1-5	Number of points to be entered for each of the three rows of the DFDOKL array. (Number of range points for DFDOKL array). With current dimensions, NDFSP $\leq$ 10.

DIRECT FIRE SUPPRESSION CARD SET

Card No. 3a

Variable name	UNITS	Format	columns	Description
DFDOKL (1,1)	meters	F10.4	1-10	DFDOKL (I,1) is the range in meters corresponding to a probability of DO being suppressed (obscured) of DFDOKL (I,2) and a probability of DO being killed of DFDOKL (I,3) DFDOKL (1,1) = 0.0 and DFDOKL (I,1) < DFDOKL (I+1,1). If NDFSP > 8, then a second card is required for DFDOKL (9,1) through DFDOKL (NDFSP,1).
DFDOKL (2,1)	meters	F10.4	11-20	
DFDOKL (3,1)	meters	F10.4	21-30	
DFDOKL (4,1)	meters	F10.4	31-40	
DFDOKL (5,1)	meters	F10.4	41-50	If NDFSP > 8, then a second card is required for DFDOKL (9,1) through DFDOKL (NDFSP,1).
DFDOKL (6,1)	meters	F10.4	51-60	
DFDOKL (7,1)	meters	F10.4	61-70	
DFDOKL (8,1)	meters	F10.4	71-80	

DIRECT FIRE SUPPRESSION CARD SET

Card No. 3b

Variable name	UNITS	Format	columns	Description
DFDOKL (1,2)		F10.4	1-10	DFDOKL (1,2) is the probability that direct fire dust from the target at range DFDOKL (1,1) obscures the DO (but he is not killed).
DFDOKL (2,2)		F10.4	11-20	
DFDOKL (8,2)		F10.4	71-80	
				If NDFSP > 8, then a second card is needed for DFDOKL (9,2) through DFDOKL (NDFSP,2).

DIRECT FIRE SUPPRESSION CARD SET

Card No. 3c

Variable name	UNITS	Format	columns	Description
DFDOKL (1,3)		F10.4	1-10	DFDOKL (1,3) is the probability that direct fire from the target at range DFDOKL (1,1) kills the DO. If NDFSP > 8, then a second card is needed for DFDOKL (9,3) through DFDOKL (NDFSP,3)
DFDOKL (2,3)		F10.4	11-20	
⋮		⋮	⋮	
DFDOKL (8,3)		F10.4	71-80	

5. Random Occurrence Data Block Input.

When reading random occurrence data in PREPMS for transfer to TAPE 11, the first card (which indicates that a random occurrence data record is to be read) must be of the form:

NEW RECORD, record name, 5

or

NEW REC, record name, 5

where record name is the name of the random occurrence record to be read (see section 5.3.5 for random occurrence record names).

When using the TEMPORARY option with COPE, the random occurrence data is headed by a card of the form:

TEMPORARY,RANDOM OCCURRENCE DISTRIBUTION

or

TEMPORARY, RODIST

or

TEMP, RANDOM OCCURRENCE DISTRIBUTION

or

TEMP, RODIST

In either case (whether reading random occurrence data in PREPMS or COPE), the remaining cards of the random occurrence data are as follow:

RANDOM OCCURRENCE DATA CARD SET

Card No. 2

Variable name	UNITS	Format	columns	Description
NPLOSR		I5	1-5	Number of range classes for probability of LOS data for random occurrence. Current dimensions require NPLOSR ≤ 10 .

RANDOM OCCURRENCE DATA CARD SET

Card No. 3

Variable name	UNITS	Format	columns	Description
TCRIT	seconds	F10.4	1-10	Critical time for laser designation (i.e., the length of the time interval immediately prior to impact during which the target must be continuously lased in order to have a successful Copperhead mission).

RANDOM OCCURRENCE DATA CARD SET

Card No. 4

Variable name	UNITS	Format	columns	Description
RNGPLS (1)	meters	F10.4	1-10	RNGPLS (I) is the upper limit of the Ith range class for random occurrence line of sight data. Range class 1 extends from zero to RNGPLS (1); range class I for I > 1 extends from RNGPLS (I) to RNGPLS (I+1). $RNGPLS(I) < RNGPLS (I+1)$ If NPLOSR > 8, then a second card must be used to contain RNGPLS (9) through RNGPLS (NPLOSR).
RNGPLS (2)	meters	F10.4	11-20	
⋮	⋮	⋮	⋮	
RNGPLS (8)	meters	F10.4	71-80	

RANDOM OCCURRENCE DATA CARD SET

Card No. 5

Variable name	UNITS	Format	columns	Description
PRBLOS (1)		F10.4	1-10	PRBLOS (I) is the probability that a single vehicle in range class I is in view of the DO for the final TCRIT seconds of the Copperhead flight.
PRBLOS (2)		F10.4	11-20	
PRBLOS (8)		F10.4	71-80	
				If NPLOSR $\geq$ 8, then a second card is needed to contain PRBLOS(9) through PRBLOS(NPLOSR).

6. PE Data Block Input.

When reading PE data in PREPMS for transfer to TAPE 11, the first card (which indicates that a PE data record is to be read) must be of the form:

NEW RECORD, record name, 6

or

NEW REC, record name, 6

where record name is the name of the PE data record to be read (see section 5.4 for PE record names).

When using the TEMPORARY option with COPE, the PE data are headed by a card of the form:

TEMPORARY, PEDATA

or

TEMP, PEDATA

In either case (whether reading PE data in PREPMS or COPE), there is an optional card that can be inserted next:

USE TAPE 4

If this card is used, then the program (PREPMS or COPE) reads the remaining cards of the PE data from I/O unit 4 instead of the normal I/O unit 5; if this card is not used, then the PE data are read from I/O unit 5. In either case (whether I/O unit 4 or 5 is used), the remaining PE data cards are as described after this note:

Note: Because the PAM program (see Chapter 16) can write the PE data records directly to TAPE 11 without any need to run PREPMS, it is very unlikely that the need will arise to use PREPMS (or COPE) to read PE data as described in this section; however, the capability is provided.

PE DATA _____ CARD SET

Card No. 2a - 2dd

Variable name	UNITS	Format	columns	Description
INDEX (I,J,1)		I5	1-5	$INDEX(I,J,K) = 10*(I-1) + 2*(J-1) + K$ where $K = 1, 2$ $J = 1, 2, \dots, 5$ $I = 1, 2, \dots, 6$ The values are read two to a card and the values are arranged varying K first, then J, and then I. This procedure yields a set of 30 cards. The Nth such card having the values (2N-1) and 2N on it in 2I5 format. The index array is used to access the PETBL array.
INDEX (I,J,2)		I5	6-10	

PE DATA

CARD SET

Card No. 3a - 3b²⁴
(600 cards)

Variable name	UNITS	Format	columns	Description
PETBL (I,J,1)		F10.4	1-10	PETBL (I,J,K) is the probability of engagement value for DO-to-target range K, visibility range J, and index value I.
PETBL (I,J,2)		F10.4	11-20	
PETBL (I,J,3)		F10.4	21-30	The values are input seven per card so that K varies first, then J, and finally I. K = 1,2,...,7 J = 1,2,...,10 I = 1,2,...,60.
PETBL (I,J,4)		F10.4	31-40	
PETBL (I,J,5)		F10.4	41-50	This makes for a total of 600 cards with seven entries per card in 7F10.4 format.
PETBL (I,J,6)		F10.4	51-60	
PETBL (I,J,7)		F10.4	61-70	

7. Posture Data Block Input.

When reading posture data in PREPMS for transfer to TAPE 11, the first card (which indicates that a Target Posture data record is to be read) must be of the form:

NEW RECORD, record name, 7

or

NEW REC, record name, 7

where record name is the name of the target posture data to be read (see section 5.3.6 for target posture record names).

When using the TEMPORARY option with COPE, the target posture data are headed by a card of the form:

TEMPORARY, TARGET POSTURE DISTRIBUTION

or

TEMPORARY, TGTPSTDIST

or

TEMP, TARGET POSTURE DISTRIBUTION

or

TEMP, TGTPSTDIST

In either case (whether reading target posture data in PREPMS or COPE), the remaining cards of the target posture data are as follow:

POSTURE DATA CARD SET

Card No. 2

Variable name	UNITS	Format	columns	Description
NRNGPS		I5	1-5	Number of range classes used for target posture data. Current dimensions require $NRNGPS \leq 10$.

POSTURE DATA

CARD SET

Card No. 3

Variable name	UNITS	Format	columns	Description
RNGPST (1)	meters	F10.4	1-10	RNGPST (I) is the upper limit of the Ith range class for posture data. The first range class runs from zero to RNGPST (1); the Ith range class extends from RNGPST (I) to RNGPST (I+1). Note RNGPST (I) < RNGPST (I+1) Also, if NRNGPS > 8, the a second card is required: for RNGPST (9) through RNGPST (NRNGPS).
RNGPST (2)	meters	F10.4	11-20	
RNGPST (3)	meters	F10.4	21-30	
I	I	I	I	
I	I	I	I	
I	I	I	I	
RNGPST (8)	meters	F10.4	71-80	

POSTURE DATA

CARD SET

Card No. 4a

Variable name	UNITS	Format	columns	Description
PSTTBL (1,1)		F10.4	1-10	PSTTBL (I,1) is the probability that the target is completely obscured when at DO-to-target range RNGPST (I).
PSTTBL (2,1)		F10.4	11-20	
{		{	{	
{		{	{	If NRNGPS > 8, then a second card is required for PSTTBL (9,1) through PSTTBL (NRNGPS,1).
PSTTBL (8,1)		F10.4	71-80	

POSTURE DATA CARD SET

Card No. 4b

Variable name	UNITS	Format	columns	Description
PSTTBL (1,2)		F10.4	1-10	PSTTBL (1,2) is the probability that the target is fully exposed when at DO-to-target range RNGPST (1).
PSTTBL (2,2)		F10.4	11-20	
				As usual, if NRNGPS > 8, a second card is needed for PSTTBL (9,2) through PSTTBL (NRNGPS,2).
PSTTBL (8,2)		F10.4	71-80	

POSTURE DATA

CARD SET

Card No. 4c

Variable name	UNITS	Format	columns	Description
PSTTBL (1,3)		F10.4	1-10	PSTTBL (1,3) is the probability that the target is hull deflade (turret exposed) when at a 90-to-target range of RNGPST (1). Again, if NRNGPS > 8, a second card is needed for PSTTBL (9,3) through PSTTBL (NRNGPS,3).
PSTTBL (2,3)		F10.4	11-20	
PSTTBL (8,3)		F10.4	71-80	

8. Invariant Data Block.

When reading invariant data in PREPMS for transfer to TAPE 11, the first card (which indicates that an invariant data record is to be read) must be of the form:

NEW RECORD, record name, 8

or

NEW REC, record name, 8

where record name is the name of the invariant data record to be read (see section 5.3.7 for invariant data record names).

When using the TEMPORARY option with COPE, the invariant data is headed by a card of the form:

TEMPORARY, INVARIANT

or

TEMP, INVARIANT

In either case (whether reading invariant data in PREPMS or COPE), the remaining cards of the invariant data are as follow:

INVARIANT DATA CARD SET

Card No. 2

Variable name	UNITS	Format	columns	Description
NRNGTT		I5	1-5	The number of range classes used with the LDWSS probability of hit numbers. NRNGTT $\leq$ 20 with current dimensions.

INVARIANT DATA

CARD SET

Card No. 3a - 3t

Variable name	UNITS	Format	columns	Description
RNGTTF (J)	meters	F10.4	1-10	RNGTTF (J) is the Ith range at which probability of hit data is entered. RNGTTF (1) = 0.0 and RNGTTF (I) < RNGTTF (I+1).
TTF (1,J,1)		F10.4	11-20	TTF (I,J,K) is the probability of hit at range J using designator type I against a target in posture K. I = 1 for GLLD I = 2 for MULE I = 3 for LTD K = 1 for fully exposed K = 2 for hull defilade (turret exposed) There will be NRNGTT cards of this type with each card containing the same data points as its predecessor but for the next J value. There will be a maximum of 20 such cards.
TTF (1,J,2)		F10.4	21-30	
TTF (2,J,1)		F10.4	31-40	
TTF (2,J,2)		F10.4	41-50	
TTF (3,J,1)		F10.4	51-60	
TTF (3,J,2)		F10.4	61-70	

INVARIANT DATA

CARD SET

Card No. ⁴

Variable name	UNITS	Format	columns	Description
DLTT (1)	seconds	F10.4	1-10	DLTT (I) is the Ith response time delay value to be used in interpolating with respect to delay time in the PETBL array. DLTT (1) = 0.0 and DLTT (I) < DLTT (I+1).
DLTT (2)	seconds	F10.4	11-20	
DLTT (3)	seconds	F10.4	21-30	
DLTT (4)	seconds	F10.4	31-40	
DLTT (5)	seconds	F10.4	41-50	
DLTT (6)	seconds	F10.4	51-60	

INVARIANT DATA

CARD SET

Card No. 5a

Variable name	UNITS	Format	columns	Description
PKTBL (1,1)		F10.4	1-10	PKTBL (I,1) is the probability of kill given a Copperhead hit against target type I in a fully exposed posture,
PKTBL (2,1)		F10.4	11-20	
PKTBL (3,1)		F10.4	21-30	
PKTBL (4,1)		F10.4	31-40	
PKTBL (5,1)		F10.4	41-50	
PKTBL (6,1)		F10.4	51-60	
PKTBL (7,1)		F10.4	61-70	

AD-A100 285

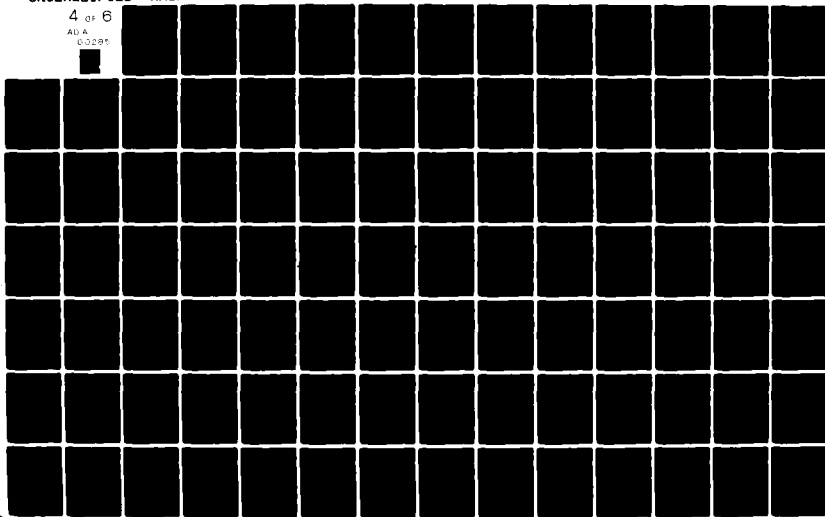
ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY ABERDEEN PROV--ETC F/G 19/1
COPPERHEAD OPERATIONAL PERFORMANCE EVALUATION (COPE): COMPUTER --ETC(U)
MAR 81 R S SANDMEYER
AMSAA-TR-318

UNCLASSIFIED

NL

4 OF 6

AD-A
00285



CARD SET

5b

14-69

14.4.3 REPORT RECORD LIST Directive. The third type of input to PREPMS is a line of the form:

REPORT RECORD LIST

or

REP REC LIST

Inserting this line (card or card image) in the PREPMS input will cause the program to print a list of the names of the records on TAPE 11 at the time the line is encountered.

The usual practice is to place a line of this type in the input after all of the option word records and data block records so that the list produced will include all of the records created by the current run.

An example of the list produced is shown in Appendix C.

14.4.4 INTERPRET PEDATA Directive. The fourth type of input to PREPMS is a line of the form:

INTERPRET PEDATA

Inserting a line of this form in the PREPMS input will cause the program to print a list of the names of the current PE data records on TAPE 11 as well as the values of the nine parameters corresponding to each such record (see section 5.4).

14.5 PREPMS Output.

PREPMS writes to two files, TAPE 11 and TAPE 6 (also called OUTPUT). TAPE 11 receives the data block records, the option word records, the record named OPTNNUMS (which is automatically maintained by PREPMS with no user attention), and the record index array which is automatically maintained by the CDC word addressable mass storage file facilities. TAPE 6 (usually a line printer) is a sequential file on which PREPMS echoes the inputs that create the data block records and option name records as well as prints the lists generated by INTERPRET PEDATA and REPORT RECORD LIST directives.

A sample of PREPMS output (TAPE 6) is given in Appendix C, so further discussion is not given here.

14.6 PREPMS Diagnostic Messages.

Since several of the subroutines used by PREPMS are similar to those used in COPE, any diagnostic messages not found in this section (which gives only the PREPMS-peculiar messages) should be given in Chapter 7.

14.6.1 Messages of Informational Nature Not Terminating Program.

1. **PREPMS** NON-FATAL WARNING! YOU HAVE SEQUENCE NUMBERS IN COLS. 73-78 OF TAPE5. THEY WILL BE REMOVED BY THE PROGRAM; HOWEVER, THE RESULTS MAY BE SUSPECT.

This message is generated in SEPRC2 and is caused when the computer detects editor sequence numbers in the INPUT file (see section 7.1).

This message does not stop the program; however, it is very probable that the wrong data has been used if this message is printed. Great care should be taken to assure that the OUTPUT shows the correct (intended) values and that no sequence numbers were taken as input.

Note that the sequence numbers are removed only from those lines entered in free format. All lines entered as formatted data will still have sequence numbers which may be read instead of the intended data (for columns 73-78).

To avoid this problem, one should always use an 80 character line length when editing the input file for PREPMS (i.e., F,CH=80 on CDC).

2. **PREPMS** THIS IS A CREATION RUN FOR RECORD OPTNUMS.

This message is not really a diagnostic but merely indicates to the user that this run of PREPMS creates a new TAPE 11. If this message is not present, it means that the current PREPMS run is modifying an existing TAPE 11 rather than creating a new one.

14.6.2 STOPS With Messages in Day Files.

1. STOP IN PREPMS: ERROR NUMBER 1.

This stop occurs when the program has read an input card which should be the first card of one of the four input types of section 14.4, but does not recognize the card as one of those four types.

This error is most likely to happen when either the card is mispunched or the previous input set (especially if it was type 1 or type 2 input) has the wrong number of cards.

2. STOP IN PREPMS: ERROR NUMBER 2.

This error occurs when the third field of a type 1 card of a type 1 input is not a number. (That is, the option word record input type starts with a card that has a non-numeral for the value of NREAD. See section 14.4.1 for the correct form of this card type).

3. STOP IN PREPMS: ERROR NUMBER 3.

This error occurs when the fourth field of a type 1 card (the value of NOPT1) of a type 1 input is not a number. (See section 14.4.1 for the correct form of this card type).

4. STOP IN PREPMS: ERROR NUMBER 4.

This error occurs when the first field of a type 2 card of a type 1 input is not a number. Again, see section 14.4.1 for the correct form of this card.

5. STOP IN PREPMS: ERROR NUMBER 5.

This error occurs when the second field of a type 2 card of a type 1 input is not a number. See section 14.4.1 for the correct form of this card.

6. STOP IN PREPMS: ERROR NUMBER 6.

This error occurs when the third field of the first card of a type 2 input is not a number. See section 14.4.2 (subsections 1 through 8) for the proper first card form for this input type.

7. STOP IN PREPMS: PROGRAM TERMINATION.

This means that the program reached the end of a file on TAPE 5 (INPUT) without any errors detected. It is the message that accompanies a successful run, but of course it is no guarantee that the data are correct (only that they are formatted in a valid form).

8. STOP IN SEPRC2: ERROR NUMBER 1.

This stop occurs only if the number of fields on a free formatted input card is greater than 10 or less than or equal to 0.

CHAPTER 15

15. PRBLOS PREPROCESSOR

The PRBLOS preprocessor is used to calculate the probability that a target randomly located along a route of approach within a certain designator-to-target range band will be in the line-of-sight of the designator for at least the final critical seconds of the COPPERHEAD trajectory.

This chapter gives: (1) the mathematical derivation of the method for computing the probability mentioned above, (2) a description of the program, (3) a glossary of program variables, (4) description of inputs, and (5) description of outputs. A program listing along with sample inputs and the resulting outputs are given in Appendix B.

15.1 Method For Calculating PRBLOS.

For a particular designator-to-target range band (e.g., ranges 2000m to 3000m) define F and G to be the cumulative distribution functions for in-view and out-of-view segment lengths respectively. That is, the probability that a randomly selected LOS segment for the given range band is of length less than or equal to X is F(X); the probability that a randomly selected out-of-view segment in the given range band is of length less than or equal to X is G(X).

Let t_c be the critical time; that is the time interval at the end of the COPPERHEAD trajectory during which the target must be designated in order for the COPPERHEAD to track and hit the target. Also, let v_t be the target velocity (target velocity is assumed constant).

Then the history of a target's exposure can be thought of as a sequence of alternating in-view and out-of-view segments. For example, the sequence $f_1, g_1, f_2, g_2, f_3, g_3, \dots, h_n$, where the f_i 's are in-view segment lengths forming a sample from a population of segment lengths distributed according to the cumulative distribution F, the g_i 's are out-of-view segment lengths forming a sample from a population of segment lengths distributed according to cumulative distribution G, and h_n is the last segment which is either an f_i or a g_i depending on whether the final segment is in-view or out-of-view.

Then for a particular target's exposure (i.e., in-view, out-of-view) history, one can calculate its probability of being in-view as
probability target is in-view = fraction of target's path in-view =

$$\frac{\sum_{i=1}^N f_i}{\sum_{i=1}^N f_i + \sum_{i=1}^M g_i}$$

where M is equal to either N or N-1 depending on whether the last segment is respectively out-of-view or in-view.

In our problem though, we do not want the probability that the target is in-view, but rather the probability that it is in-view and will remain in-view for the next t_c time units. This means that the target must be in an in-view segment and be at least $t_c * v_t$ distance from the end of that segment. The probability that the target is in such a location is equal to the fraction of the target's path that consists of points that are in-view and at least a distance $t_c * v_t$ from the next out-of-view segment.

For each in-view segment either (1) $f_i \leq t_c * v_t$ in which case that in-view segment contributes nothing to the total path length satisfying the conditions, or (2) $f_i > t_c * v_t$ distance from the next out-of-view segment.

Therefore, we have:

probability target is in-view and at least $t_c * v_t$ distance from next out-of-view segment =

fraction of target's path that is in-view and at least $t_c * v_t$ distance from the next out-of-view segment =

$$\frac{\sum_{i=1}^N \max \{0, f_i - t_c * v_t\}}{\sum_{i=1}^N f_i + \sum_{i=1}^M g_i}$$

where M and N are as before and $\max \{0, f_i - t_c * v_t\}$ is the length contributed by the i th in-view segment to the total path length that is in-view and at least $t_c * v_t$ away from the next out-of-view segment.

Note: $\max \{0, f_i - t_c * v_t\}$ means that the i th term of the sum is to be the maximum of the two quantities 0 and $(f_i - t_c * v_t)$.

Let $S_i = \max \{0, f_i - t_c * v_t\}$ then the S_i 's can be obtained by sampling f_i 's from a population with cumulative distribution F and then applying the formula $S_i = \max \{0, f_i - t_c * v_t\}$. This is equivalent to sampling from a population with cumulative distribution S where:

$$S(0) = F(t_c * v_t) \text{ and}$$

$$S(X) = F(x + t_c * v_t) \text{ for } x > 0.$$

Hence, our probability that the target is in-view and at least $t_c \cdot v_t$ from the next out-of-view segment can be written:

$$\frac{\frac{1}{N} \sum_{i=1}^N \max \{0, f_i - t_c \cdot v_t\}}{\frac{1}{N} \sum_{i=1}^N f_i + \frac{1}{N} \sum_{i=1}^M g_i}$$

Now to obtain an overall estimate of this probability (which we call PRBLOS) we can imagine that we let our target wander back and forth over the various paths in the given range band. As this is done, $N \rightarrow \infty$ (and also $M/N \rightarrow 1$) so that

$$\frac{1}{N} \sum_{i=1}^N f_i \rightarrow M_f \text{ (mean of } f_i \text{ population)}$$

$$\frac{1}{N} \sum_{i=1}^M g_i \rightarrow \frac{M}{N} \left\{ \frac{1}{M} \sum_{i=1}^M g_i \right\} \rightarrow \frac{M}{N} M_g \rightarrow M_g \text{ (mean of } g_i \text{ population)}$$

$$\frac{1}{N} \sum_{i=1}^N \max \{0, f_i - t_c \cdot v_t\} = \frac{1}{N} \sum_{i=1}^N s_i \rightarrow M_s \text{ (mean of } s_i \text{ population)}$$

Therefore,

$$\text{PRBLOS} = \frac{M_s}{M_f + M_g}$$

is used as the probability that a vehicle randomly located in a given range band is in-view (from the designator's position) and will remain in-view for at least t_c time units.

15.2 Description of PRBLOS Program and Glossary of Variables.

The program is extremely simple and consists of nothing but the steps required to perform the calculations described in 15.1 as well as to read the input and print the results.

The program can be summarized as four main steps:

- (1) Read the input for the next case and print it for checking purposes; if no input, stop.
- (2) Form the modified distribution (the one with cumulative distribution function S in section 15.1) and print it.
- (3) Calculate the means of the three distributions F , G , and S (i.e., M_f , M_g , M_s) and from that calculate PRBLOS.
- (4) Print PRBLOS and return to step (1) to see whether there is another case.

The subroutine AVEL of the program computes the mean of a distribution with cumulative distribution function F by using the formula:

$$M_f = \int_L^U xf(x)dx = UF(U) - LF(L) - \int_L^U F(x)dx$$

where f is the probability density function (i.e., $f = F'$ almost everywhere) and U and L are respectively the upper and lower bounds of the distribution (in our cases, $0 \leq L < U < \infty$).

Since F is defined in the program by an array of ordered pairs (x_i, F_i) $i = 1, 2, \dots, K$ where F_i = probability that x is less than or equal to x_i , it is possible to approximate the right most integral by:

$$(*) \int_L^U F(x)dx \approx \frac{1}{2} \sum_{i=1}^{K-1} (x_{i+1} - x_i)(F_{i+1} + F_i).$$

Hence, the subroutine AVEL calculates the mean of a population with cumulative distribution function F using the formula:

$$M_f = x_K F_K - x_1 F_1 - \frac{1}{2} \sum_{i=1}^{K-1} (x_{i+1} - x_i)(F_{i+1} + F_i).$$

The actual variable names used in the program are different from those used in sections 15.1 and 15.2; however, they are defined in the variable glossary so understanding should be fairly easy.

GLOSSARY OF PRBLOS VARIABLES

VARIABLE	TYPE	COMMON BLOCK	UNITS	DEFINITION
A(6)		(LOCAL)		The A array contains up to 60 alphanumeric characters used as a case title
AVE		(LOCAL) (F.P.)	meters	AVE is the value returned by subroutine AVEL as the mean length of the distribution with which it (AVEL) was called
AVEIV		(LOCAL)	meters	AVEIV is the mean length of the in-view segment lengths (M_f of section 15.1)
AVEIVM		(LOCAL)	meters	AVEIVM is the mean length of the modified in-view segment lengths (M_g of section 15.1)
AVEOOV		(LOCAL)	meters	AVEOOV is the mean length of the out-of-view segment lengths (M_g of section 15.1)
CD		(LOCAL)	meters	CD is the critical distance; that is, the distance that the target will cover during the critical time ($t_c \cdot v_t$ in the notation of section 15.1)
I		(LOCAL)		I is used as a DO-LOOP control variable and as a subscript value.
J		(LOCAL)		J is used as a DO-LOOP control variable and as a subscript value.
K		(LOCAL)		K is used as a DO-LOOP control variable and as a subscript value.
NIV		(LOCAL)		NIV is the number of points (ie, pairs, PRIV(1,1) and PRIV(1,2) used in the current case to describe the cumulative in-view segment length distribution ($NIV \leq 20$ with current program dimensions)
NIVM		(LOCAL)		NIVM is the number of points (ie, pairs PRIVM(1,1) and PRIVM(1,2) used in the current case to describe the cumulative modified in-view segment length distribution ($NIVM \leq 20$ with current program dimensions).
NOOV		(LOCAL)		NOOV is the number of points (ie, pairs PROOV(1,1) and PROOV(1,2) used in the current case to describe the cumulative out-of-view segment length distribution ($NOOV \leq 20$ with current program dimensions).
NU		(LOCAL)		NU is equal to one less than the number of points used to describe the cumulative distribution whose mean is being calculated by subroutine AVEL. It is the number of terms in the sum of equation (*) of section 15.2.
NI		(LOCAL) (F.P.)		NI is the first dimension of the P array. In this program, NI is always equal to 20.

GLOSSARY OF PRBLOS VARIABLES - CONTINUED

VARIABLE	TYPE	COMMON BLOCK	UNITS	DEFINITION
N2		(LOCAL) (F.P.)		N2 is the second dimension of the P array. In this program, N2 is always equal to 2.
N3		(LOCAL) (F.P.)		N3 is the number of points used to describe the cumulative distribution whose mean is being calculated by subroutine AVEL.
P(N1,N2)		(LOCAL) (F.P.)	P(I,2) is in meters	P(I,1) is the probability that a segment length (randomly selected from the population whose mean is now being calculated by subroutine AVEL) is less than or equal to P(I,2). (where I=1, 2, ..., N3).
PRBIVM		(LOCAL)		This is the variable PRBLOS of section 15.1 (Since the program card uses PRBLOS as the program name it was necessary to call the variable by another name) It is the probability that a vehicle randomly located along an approach path in the given range band is currently in view and will remain in view for at least t_c seconds.
PRIV(20,2)		(LOCAL)	PRIV (I,2) is in meters	PRIV is the array describing the cumulative distribution of in-view segment lengths (F of section 15.1) PRIV (I,1) is the probability that a randomly chosen in-view segment is of length PRIV(I,2) or less.
PRIVM(20,2)		(LOCAL)	PRIVM(I,2) is in meters	PRIVM is the array describing the cumulative distribution of modified in-view segment lengths (S of section 15.1). PRIVM(I,1) is the probability that a randomly chosen modified in-view segment is of length PRIVM(I,2) or less.
PROOV(20,2)		(LOCAL)	PROOV(I,2) is in meters	PROOV is the array describing the cumulative distribution of out-of-view segment lengths (G of section 15.1). PROOV(I,1) is the probability that a randomly chosen out-of-view segment is of length PROOV(I,2) or less.
SUM		(LOCAL)		SUM is the value of the summation of equation (*) of section 15.2.
TC		(LOCAL)	seconds	TC is the critical time (t_c of section 15.1) It is the duration of the interval at the end of the Copperhead trajectory during which the target must be (nearly) continuously lased in order that the round can track and hit the target.
TGTVEL		(LOCAL)	meters/sec.	TGTVEL (v_t of section 15.1) is the velocity of the target.

15.3 PRBLOS Inputs.

For each case that is to be run with the PRBLOS program, a set of input cards (or card images) composed of card types 1 through 5b must be prepared. Multiple cases can be run by simply stacking card-sets for the various cases one after another. The first card of each case has room (columns 21 through 80) for an identifying alphanumeric string which is reproduced on output for easier case identification. The program runs all cases until it either runs out of input or encounters an illegal input value.

Since the normal use of the PRBLOS program is to calculate the values to fill the RANDOM OCCURRENCE Data Block of COPE (in particular the PRBLOS array), each case run of the PRBLOS program corresponds to a given terrain, target velocity, and range band combination. For consistency, the PRIV data for PRBLOS should be obtained from the same source as the SEGLOS data for COPE. (Indeed, there is no reason why for a given terrain the same values cannot be used for PRIV(I,1) as for SEGLOS (I,1) and for PRIV(I,2) as for SEGLOS (I,J) in the (J-1)th range band.)

TGTVEL, TC, A CARD SET

CARD NO. 1

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
TGTVEL	meters/sec.	F10.4	1-10	The velocity of the target
TC	seconds	F10.4	11-20	The critical time
A(1)		A10	21-30	A(1) through A(6) can be used to hold up to 60 characters of case heading which are copied from this card to the first output line of each case. This has no effect on calculations but enables the user to label the cases.
A(2)		A10	31-40	
A(3)		A10	41-50	
A(4)		A10	51-60	
A(5)		A10	61-70	
A(6)		A10	71-80	

NIV CARD SET

CARD NO. 2

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
NIV		15	1-5	This is the number of points to be read to describe the cumulative in-view segment length distribution. NIV controls reading of PRIV: PRIV (I,1) for I=1,2...,NIV PRIV (I,2) for I=1,2...,NIV NOTE: Current program dimensions require $NIV \leq 20$.

PRIV (I,1)

CARD SET

CARD NO. 3a

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
PRIV(1,1)		F10.5	1-10	PRIV(I,1) is the probability that a randomly chosen in-view segment has length less than or equal to PRIV(I,2). NOTE: $PRIV(I,1) < PRIV(I+1,1)$ for $I = 1, 2, NIV-1$; $0 \leq PRIV(I,1) \leq 1$ for $I = 1, 2, \dots, NIV$; and $PRIV(NIV,1) = 1.0$ If $NIV > 8$, then an additional card (or two) is needed for PRIV(9,1) through PRIV(NIV,1). The additional card(s) should have 8F10.4 format also. PRIV(I,1) and PRIV(I,2) together correspond to the F_i and X_i values respectively (in the notation of section 15.2) for the in-view distribution.
PRIV(2,1)		F10.5	11-20	
PRIV(3,1)		F10.5	21-30	
PRIV(4,1)		F10.5	31-40	
PRIV(5,1)		F10.5	41-50	
PRIV(6,1)		F10.5	51-60	
PRIV(7,1)		F10.5	61-70	
PRIV(8,1)		F10.5	71-80	

PRIV (I,2) CARD SET

CARD NO. 3b

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
PRIV(1,2)	meters	F10.5	1-10	PRIV(I,2) is the in-view segment length corresponding to cumulative probability PRIV(I,1) NOTE: $\text{PRIV}(I,2) \leq \text{PRIV}(I+1,2)$ for $I = 1, 2, \dots, \text{NIV}-1$ If NIV > 8, then an additional card (or two) is needed for PRIV(9,2) through PRIV(NIV,2). The additional card(s) have 8F10.4 format.
PRIV(2,2)	meters	F10.5	11-20	
PRIV(3,2)	meters	F10.5	21-30	
PRIV(4,2)	meters	F10.5	31-40	
PRIV(5,2)	meters	F10.5	41-50	
PRIV(6,2)	meters	F10.5	51-60	
PRIV(7,2)	meters	F10.5	61-70	
PRIV(8,2)	meters	F10.5	71-80	

NOOV

CARD SET

CARD NO. 4

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
NOOV		I5	1-5	NOOV is the number of points to be read to describe the cumulative out-of-view segment length distribution. NOOV controls reading of PROOV: PROOV(I,1) for I=1,2...,NOOV PROOV(I,2) for I=1,2...,NOOV NOTE: Current dimensions require NOOV $\leq$ 20.

PROOV (I,1)

CARD SET

CARD NO. 5a

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
PROOV(1,1)		F10.4	1-10	PROOV(I,1) is the probability that a randomly chosen out-of-view segment has length less than or equal to PROOV(I,2). NOTE: $\text{PROOV}(I,1) \leq \text{PROOV}(I+1,1)$ for $I = 1, 2, \dots, \text{NOOV}-1$; $0 \leq \text{PROOV}(I,1) \leq 1.0$ for $I = 1, 2, \dots, \text{NOOV}$; and $\text{PROOV}(\text{NOOV},1) = 1.0$ If $\text{NOOV} > 8$, then an additional card(or two) is needed for PROOV(9,1) through PROOV(NOOV,1). The additional card(s) should have 8F10.4 format. PROOV(I,1) and PROOV(I,2) correspond to a point on the cumulative distribution G of section 15.1 (a point with coordinates (PROOV(I,2), PROOV(I,1)) in the usual notation.
PROOV(2,1)		F10.4	11-20	
PROOV(3,1)		F10.4	21-30	
PROOV(4,1)		F10.4	31-40	
PROOV(5,1)		F10.4	41-50	
PROOV(6,1)		F10.4	51-60	
PROOV(7,1)		F10.4	61-70	
PROOV(8,1)		F10.4	71-80	

PROOV(I,2) CARD SET

CARD NO. 5b

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
PROOV(1,2)	meters	F10.4	1-10	PROOV(I,2) is the out-of-view segment length corresponding to cumulative probability PROOV (I,1).
PROOV(2,2)	meters	F10.4	11-20	
PROOV(3,2)	meters	F10.4	21-30	<u>NOTE:</u> $PROOV(I,2) \leq PROOV(I+1,2)$ for $I = 1, 2, \dots, NOOV-1;$ $0.0 < PROOV(I,2) \leq 1.0$ for $I = 1, 2, \dots, NOOV.$
PROOV(4,2)	meters	F10.4	31-40	
PROOV(5,2)	meters	F10.4	41-50	If NOOV > 8, then an additional card (or two) is needed for PROOV(9,2) through PROOV(NOOV,2). The additional card(s) have 8F10.4 format.
PROOV(6,2)	meters	F10.4	51-60	
PROOV(7,2)	meters	F10.4	61-70	
PROOV(8,2)	meters	F10.4	71-80	

15.4 PRBLOS Output and Diagnostics.

15.4.1 Output. The output of the PRBLOS program is very brief and easy to understand.

For each case run, the first line of output prints the case identifier (if any). The next line prints the target velocity and critical time values. Then comes the PRIV array preceded by NIV. Next the PROOV array preceded by NOOV and the PRIVM array preceded by NIVM. Finally, it prints the average segment lengths of the three distributions and the PRBLOS value followed by the words "END OF CASE".

In the special case that $t_c \cdot v_t$ is greater than the longest possible in-view segment, a message to that effect is printed along with the value of PRBLOS (namely, 0.0). In this case, the PRIVM array is not printed.

Note that PRBIVM is labeled PRBLOS when printed. (This is because PRBIVM is a second choice name for PRBLOS used only because the program card name PRBLOS prevented the use of a variable PRBLOS in the program).

A program listing as well as sample input and output for PRBLOS is contained in Appendix B.

15.4.2 Diagnostic ("STOP") Messages in Dayfile.

1. STOP END OF INPUT.

This message means that the program has encountered the end of file of TAPE5 (INPUT). This is the normal program termination message, but of course, it is no guarantee that the results are correct.

2. STOP NIV>20 EXCEEDS DIMENSIONS OF PRIV ARRAY.

This message means the user attempted to read in an NIV value greater than 20. It may occur because NIV is not right justified in columns 1-5 of its input card. Of course, it may also occur if NIV is really greater than 20; if this is the case, then the user must either use fewer points to describe PRIV or redimension PRIV in the program (and change N1 in first call of AVEL. Redimensioning of PRIVM and changing N1 in the third call of AVEL may also be required).

3. STOP NOOV>20 EXCEEDS DIMENSIONS OF PROOV ARRAY.

This message means the user attempted to read in an NOOV value greater than 20. It may occur because NOOV is not right justified in columns 1-5 of its input card. Of course, it may also occur if NIV is really greater than 20; if this is the case, then the user must either use fewer points to describe PROOV or redimension PROOV in the program (and change N2 in the second call of AVEL).

CHAPTER 16

16. PAM PREPROCESSOR

Unlike the other programs documented in this report (COPE, PREPMS, and PRBLOS), the PAM program was not written by the author of this report. Since the author (Michael Starks, GWD, AMSAA) of PAM will document it in a separate report, this chapter is limited to a brief explanation of the program's purpose and a description of the input. Appendix A contains a program listing as well as sample inputs and outputs.

16.1 General Description of PE Program (PAM).

The PAM program is used to compute the probability of engagement of the target by the COPPERHEAD seeker. More precisely, probability of engagement (PE) is the probability that the COPPERHEAD seeker receives enough reflected laser energy from the target for the COPPERHEAD round to initiate maneuver toward the target and that the target is within the maneuver footprint of the COPPERHEAD.

The PAM model produces a table of 4200 PE values (6 cloud ceiling altitudes x 10 meteorological visibility limits x 5 delay times x 7 designator-to-target ranges x 2 "MUTS" factors). (The multi-unit target factor ("MUTS" factor) allows TLE to be reduced when the predicted impact point is bracketed by a column of target vehicles.)

Such a table then has PE values as a function of cloud ceiling, visibility limit, delay time, MUTS factor, and designator-to-target range. In addition, the set of PE's in the table is a function of the 9 parameters mentioned in section 5.4 (namely, nominal response time, designator type (power), target velocity, gun-target range, reflectivity of target, angle T, deflection bias (distance between footprint centroid and target path point of closest approach), target heading, and seeker sensitivity). Hence, for each different combination of values of the 9 parameters a different set of PE values is needed which, in turn, requires another run of the PAM program.

The PAM program includes the code mentioned in section 5.4 that generates PE data block record names from the values of the 9 parameters. The subroutine PENAM2 is included in PAM and is identical to the PENAME subroutine in COPE and PREPMS except for the change in name and the fact that the data statement for the XVALUE array is included in PENAM2 rather than in a separate BLOCK DATA subprogram as in COPE and PREPMS. The error stops with messages produced by PENAM2 correspond to those for PENAME as explained in section 7.5.

Assuming that the user runs PAM with values of the 9 parameters that are permitted by PENAME (see section 5.4 and discussion of XVALUE array), then a PE DATA record name is generated. The PE values are loaded in the arrays PE (for writing to TAPE 11 as a PE data record for use by COPE) and PENG (for writing to TAPE 6 (output) to provide a hard copy of the PE values).

In order that the values written to TAPE 11 are accessible by COPE, of course, it is necessary to catalog TAPE 11 and use it with the COPE runs. The interface between PAM and COPE has been covered in section 5.4.

The output of PAM consists mainly of the inputs which are echoed for checking purposes (some of the angles are converted to radians before being printed). The PE values are printed in 70 groups of 60 (6 cloud ceiling altitudes, 10 visibility limits) with each group corresponding to a particular designator-to-target range, time delay, and MUTS factor combination.

A program listing together with sample inputs and outputs is included in in Appendix A.

16.2 PAM Inputs.

The PAM program takes all its inputs in free format (in CDC terminology, it uses list-directed reads). This means that on each card in the following description that takes more than one variable, the values of the variables are punched one after another with commas between successive values. (See sample input in Appendix A).

IMF CARD SET

CARD NO. 1

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
IMF		Free		Mission type flag IMF = 1 for preplanned target (priority of not) IMF = 2 for target of opportunity

ETH CARD SET

CARD NO. 2

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
ETH	Joules/km ²	Free		ETH is the energy threshold. That is, the amount of energy/ area that must reach the seeker to initiate maneuver.

AOF CARD SET

CARD NO. 3

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
AOF	Degrees	Free		AOF is the angle of fall of the COPPERHEAD round (measured from the horizontal).

TH, PCA CARD SET

CARD NO. 4

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
TH	Degrees	Free		Target heading. The angle between the gun-to-target line and the target path when measured counter-clockwise from the gun-target line.
PCA	Degrees	Free		Point of closest approach (deflection bias). This is the distance between the footprint centroid and the closest point on the target's path to the centroid.

AZDT, V, RHO, ED, TR CARD SET

CARD NO. 5

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
AZDT	Degrees	Free		AZDT is the angle T (that is, the angle between the designator-target line and gun-target line measured counter-clockwise from the gun-target line.)*
V	Meters/Sec	Free		V is the target velocity.
RHO		Free		RHO is the target's reflectivity.
ED	Millijoules	Free		Designator energy.
TR	Seconds	Free		Nominal response time (time from DO's call for fire to expected round arrival. It <u>includes</u> time-of-flight in this case.
				*Note: AZDT is the nominal angle T; that is, the angle between the designator-target line and the gun target-line when TR seconds have passed since DO's call for fire.

NK, RNG, ACCX, ACCY CARD SET

CARD NO. 6

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
NK		Free		NK is the Monte Carlo sample size used in the program.
RNG	Meters	Free		Gun-to-target range (must be the range for which footprints are supplied).
ACCX	Meters	Free		ACCX is the COPPERHEAD total delivery error standard deviation in the deflection direction.*
ACCY	Meters	Free		ACCY is the COPPERHEAD total delivery error standard deviation in the range direction.*
				*Note: These are the delivery error standard deviations for <u>unguided</u> COPPERHEAD.

IDRMN, IDRMX, IVMX CARD SET

CARD NO. 7

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
IDRMN	Km	Free		Minimum designator-to-target range.
IDRMX	Km	Free		Maximum designator-to-target range.
IVMX	Km	Free		Maximum visibility range.

NI CARD SET

CARD NO. 8a

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
NI(J)		Free		NI(J) is the number of points used to describe the maneuver footprint for the Jth altitude. J = 1 for highest altitude 2 for second highest altitude . . 6 for lowest altitude The six altitudes for which footprints are entered are the six altitudes used for cloud ceiling altitudes in COPE weather data. Note that there will be a set of cards 8a, 8b, 8c for each footprint for a total of six such sets in all. They must be input in order of ascending J's. With current dimensions $NI(J) \leq 11$ for all J.

THEMH

CARD SET

CARD NO. 8b

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
THEMH(1,J)	degrees	free		THEMH(I,J) is the angle used to represent the Ith point on the boundary of the Jth footprint.
THEMH(2,J)	degrees	free		
THEMH(3,J)	degrees	free		Note that the Jth footprint is represented by a set of points $\{(\rho_i, \theta_i)\}$ in polar coordinates, where
THEMH(4,J)	degrees	free		
THEMH(5,J)	degrees	free		$\theta_i = \text{THEMH}(I,J)$ $\rho_i = \text{DISMH}(I,J)$.
THEMH(6,J)	degrees	free		
THEMH(7,J)	degrees	free		$\text{THEMH}(1,J) = 0.0$ $\text{THEMH}(\text{NI}(J), J) = 180.0$ The angles are measured off the ray in the positive range direction from the centroid of the footprint.
THEMH(8,J)	degrees	free		
THEMH(9,J)	degrees	free		Because the footprint is symmetric about the range axis, only one half of it is read in. It is for this reason that the $\text{THEMH}(I,J)$ values do not exceed 180 degrees. Also note that $\text{THEMH}(I,J) < \text{THEMH}(I+1,J)$ for $I = 1, 2, \dots, \text{NI}(J) - 1$ If $\text{NI}(J) < 11$, then only $\text{NI}(J)$ entries occur on this card rather than the 11 entries shown.
THEMH(10,J)	degrees	free		
THEMH(11,J)	degrees	free		

DISMH

CARD SET

CARD NO. 8c

VARIABLE NAME	UNITS	FORMAT	COLUMNS	DESCRIPTION
DISMH(1,J)	meters	free		DISMH(I,J) is the distance from the centroid to the Ith point on the boundary of the Jth footprint.
DISMH(2,J)	meters	free		
DISMH(3,J)	meters	free		
DISMH(4,J)	meters	free		
DISMH(5,J)	meters	free		
DISMH(6,J)	meters	free		
DISMH(7,J)	meters	free		
DISMH(8,J)	meters	free		
DISMH(9,J)	meters	free		
DISMH(10,J)	meters	free		
DISMH(11,J)	meters	free		If NI(J) < 11, then this card will have only NI(J) entries rather than the 11 shown.

APPENDIX A

LISTING AND SAMPLE CASE OF PAM

APPENDIX A
LISTING AND SAMPLE CASE OF PAM

This appendix contains:

- (1) A listing of the FORTRAN code for the PAM program. In addition to the routines in this listing, PAM also uses the function URAN31 which is included in the COPE listing in Appendix D.
- (2) Three sets of sample case input for PAM.
- (3) Three sample runstreams for PAM (one corresponding to each set of sample input).
- (4) Sample output created by running PAM with the first set of sample input.

Note: The column numbers included in some of the listings in this appendix are not a part of the program code, the data, or the output, but are provided only for the reader's convenience.

00000000001111111111222222222233333333334444444444555555666666677777778888888999999
READ VERTICALLY

[illegible]

000000000011111111112222222222333333333344444444445555555555666666677777777778888888999
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

LISTING A-1 FORTRAN LISTING OF PAM PROGRAM - CONT'D
(PAGE 3 OF 11)

```

1700 PE(JPE,IV,IPR)=TOT(III)
    DO 180 I=1,6
      1PF=10*(IDR-1)+2*(TOT(II)-1)+MYS
      LKUP(IDR,TOT(II),MYS)=1PF
      PKNUP(IPE,TOT(IV),1)=TOT(III)
    1800 DO 190 IC=1,6
      TOT2(IC,IV)=TOT(IIIC)
    1900 CONTINUE
    DO 200 IC=1,6
      DO 200 IA=1,6
        ACQ(IC,IA)=0.
        TOT(II)=0.
    2000 CONTINUE
    2100 CONTINUE
    2200 CONTINUE
    2300 CONTINUE
    2400 CONTINUE
    DO 250 J=1,7
      DO 250 J=1,5
        IF ((K-EQ.2).AND.((J-CT,11) GT TO 250)
          DT=DTSLJ)
          WRITE (6,F430) I,PTAK
          TO=10*(I-1)+2*(J-1)+K
          WRITE (6,F440) ((PENG(IQ,JQ,KQ),KQ=1,6),JQ=1,10)
        2500 CONTINUE
        IUNT=7
    2600 DO 270 I=1,6
      WRITE (IUNIT,F450) ((INDEX(I,J,K),K=1,2),J=1,5)
    2700 CONTINUE
    DO 280 I=1,60
      WRITE (IUNIT,F460) ((PI(I,J,K),K=1,7),J=1,20)
    2800 CONTINUE
      IF ((IUNIT-EQ.7) GT YH 200)
        IUNT=7
    GO TO 260
    2900 CALL WRITING (11,ANTY,F4700,1)COPR,1,0)
    CALL CLOSING (11)
    STOP , IN PARAM NORMAL PROGRAM TERMINATION ,
      * * * F O R M A T   S T A T E M E N T S   * * *
    3000 FORMAT (1H ,10H EXPLANING TARGET)
    3100 FORMAT (1H ,20H TARGET OF PROGRAM(TY)
    3200 FORMAT (F12.8)
    3300 FORMAT (1H ,40H T=,F12.8)
    3400 FORMAT (1H ,20H TARGET HEADING ANGLE = ,F5.2,13H PI (15
      1)
    3500 FORMAT (1H ,60H AZI=,F5.2,3H V=,F5.2,60H FID= ,F5.2,5H ED= ,F4.2,5H039920
      1 1P= ,F7.2,60H AOE= ,F7.2)
    3600 FORMAT (1H ,50H WP= ,14.60) PHIC= ,F3.0,14.60 ACCY= ,F6.2,60H ACCY = ,F6.6,039940
      11,3)
    3700 FORMAT (1H ,60H FID= ,14.60) IDPHY= ,14.70 IDWHY= ,14)
    3800

```

[illegible]

LISTING A-1 FORTRAN LISTING OF PAM PROGRAM - CONT'D

(PAGE 4 OF 17)

```

00000000011111111222222223333333344444445555555666666677777777888888899 COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (READ VERTICALLY)

380 FORMAT (1H,12H)FM(1,J)=/(12F10.1)
390 FORMAT (1H,5H J=,14,0H) H(1,J)=,14)
400 FORMAT (1H,12H DISM(1,J)=/(11F10.1)
410 FORMAT (1H,10H ICODE=,14,0H INCR=,020)
420 FORMAT (1H,10H VLOCIT=,14,2,10H MEAN PESP TIME=,17,2,10H DESIGNO=0010
      1ATOR POWER=,14,2,15H REFLECTIVITY=,14,7)
430 FORMAT (1H,10H,18H DESIGNATION RANGE=,14,13H DELTA TIME=,18,4,20H0030
      15H ONE OR SEVERAL TARGETS=,14)
440 FORMAT (6F10.5)
450 FORMAT (2110)
460 FORMAT (7F10.5)
      END

00000000011111111222222223333333344444445555555666666677777777888888899 COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (READ VERTICALLY)

```

LISTING A-1 FORTRAN LISTING OF PAM PROGRAM - CONT'D
(PAGE 5 OF 11)

```

SUBROUTINE DWRITE (X,Y,X,XT,YT,HP,NP)
DIMENSION XT(NP), YT(NP), T(16)
N=ND
N1=(N-1)/2
N2=N/2
N3=NP-N2+1
IF (NP-N) 250,100,100
N4=N1+2
IF (XT(1)-XT(2)) 110,330,260
100 CONTINUE
IF (X-2.*XT(1)+XT(2)) 240,240,120
120 IF (X-2.*XT(NP)+XT(NP-1)) 130,130,240
110 IF (NP-LT,10) GO TO 150
N5=NP-N
N5=N5/2
N6=N5+N5
IF (XT(N6)-LT,X) N4=N6
IF (XT(N6)-LT,X) GO TO 140
IF (N5-GT,1) GO TO 140
150 IF (X-XT(N4)) 180,140,160
160 IF (N4-N3) 170,160,170
170 N4=N4+1
GO TO 150
180 N4=N4-1
N5=N4-N1
GO 100 I=1,N
T(1)=T(N5)
190 N5=N5+1
L=(N+1)/2
TP=T(1)
N6=N4
N7=N4+1
J1=1
N2=N1-1
N1=1.0
DO 230 J=1,N2
N5=N4-N1
N3=N-J
DO 200 I=1,N3
N8=N5+1
T(1)=T(1+1)-T(1))/(XT(N8)-XT(N7))
200 N5=N5+1
GO TO (210,220), J1
210 N1=UNH(X-XT(N6))
J1=2
N6=N6-1
GO TO 230
N1=UNH(X-XT(N7))
J1=1
N7=N7+1
L=L-1
220 TP=TP+UNH(T)
FX=TP
PETUPH
240 WRITE (6,340) X,YT(1),YT(NP)

```

[illegible]

LISTING A-1 FORTRAN LISTING OF PAM PROGRAM - CONT'D

(PAGE 6 OF 11)

0000000011111111222222223333333344444444555555556666666677777777888888889 C O L U M N N U M B E R S
123456789012345678901234567890123456789012345678901234567890 (P E A D V E R T I C A L L Y)

```

      STOP
      250 WRITE (6,350) HP,HD
      STOP
      260 IF (X-2.*XT(1)+XT(2)) 270,240,240
      270 IF (X-2.*XT(HP)+XT(HP-1)) 240,200,200
      280 IF (HP.LT.10) GO TO 300
      H5=HP-H
      290 H5=N5/2
      N6=H4+N5
      IF (XT(N6).GT.X) H4=H6
      IF (H5.GT.1) 50 TO 200
      300 IF (X-XT(N4)) 310,310,180
      310 IF (H4-N3) 320,180,320
      320 N4=N4+1
      GO TO 300
      330 WRITE (6,360) XT(1)
      STOP
C * * * F O R M A T   S T A T E M E N T S * * *
C
      140 FORMAT (23H APG. NOT IN TABLE X=,F14.7,OH XT(1)=,F14.7,10H
      1(NP)=,F14.7,2X,6HDVDINT)
      250 FORMAT (22H TABLE TOO SMALL NP=,I5,6H HD=,I5,2X,6HDVDINT)
      360 FORMAT (23H CONSTANT TABLE XT(1)=,F14.7,2X,6HDVDINT)
      END
C

```

0000000011111111222222223333333344444444555555556666666677777777888888889 C O L U M N N U M B E R S
123456789012345678901234567890123456789012345678901234567890 (P E A D V E R T I C A L L Y)

LISTING A-1 FORTRAN LISTING OF PAM PROGRAM - CONT'D
(PAGE 7 OF 11)


```

0000000001111111122222223333333344444445555555666666777777788888899 COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (HEAD VERTICALLY)

```

```

SUBROUTINE NPAN31 (X1,X2,I)
X3=SQRT(-2.0*ALOG(NPAN31(I)))
X4=6.2831853072*NPAN31(I)
X2=X3*SIN(X4)
X1=X3*COS(X4)
RETURN
END

```

```

040910 SP-NRAN312
040920 SP-NRAN313
040930 SP-NRAN314
040940 SP-NRAN315
040950 SP-NRAN316
040960 SP-NRAN317
040970 SP-NRAN318

```

```

0000000001111111122222223333333344444445555555666666777777788888899 COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (HEAD VERTICALLY)

```

LISTING A-1 FORTRAN LISTING OF PAM PROGRAM - CONT'D

(PAGE 8 OF 11)

LISTING A-1 FORTRAN LISTING OF PAM PROGRAM - CONT'D
(PAGE 10 OF 11)


```

1 .000060
20.0
0.0,0.0
25.5,5.10,2.106.
100,8000.,100.,200.
1,7,10
0
0.0,20.,30.,45.,60.,90.,135.,180.
2000.,1800.,1600.,1400.,1200.,1000.,800.,600.
0
0.20.,30.,45.,60.,90.,135.,180.
2000.,1800.,1600.,1400.,1200.,1000.,800.,600.
0
0.20.,30.,45.,60.,90.,135.,180.
1500.,1400.,1300.,1200.,1100.,1000.,800.,600.
0
0.20.,30.,45.,60.,90.,135.,180.
1500.,1400.,1300.,1200.,1100.,1000.,800.,600.
0
0.20.,30.,45.,60.,90.,135.,180.
1000.,900.,800.,700.,600.,500.,400.,300.
0
0.20.,30.,45.,60.,90.,135.,180.
1000.,900.,800.,700.,600.,500.,400.,300.

```

LISTING A-2A SAMPLE INPUT SET 1 FOR PAM

```

1 000000
20.0
0.0,0.0
25.3,3.10,2.106.
100,8000.,100.,200.
1.7,10
8
0.0,20.,30.,45.,60.,90.,135.,180.
2000.,1800.,1600.,1400.,1200.,1000.,800.,600.
8
0.20.,30.,45.,60.,90.,135.,180.
2000.,1800.,1600.,1400.,1200.,1000.,800.,600.
8
0.20.,30.,45.,60.,90.,135.,180.
1500.,1400.,1300.,1200.,1100.,1000.,800.,600.
8
0.20.,30.,45.,60.,90.,135.,180.
1500.,1400.,1300.,1200.,1100.,1000.,800.,600.
8
0.20.,30.,45.,60.,90.,135.,180.
1000.,900.,800.,700.,600.,500.,400.,300.
8
0.20.,30.,45.,60.,90.,135.,180.
1000.,900.,800.,700.,600.,500.,400.,300.

```

LISTING A-2B SAMPLE INPUT SET 2 FOR PAM

```

1 000040
20.0
0.0,0.0
60.,5.,30.,70.,106.
100,8000.,100.,200.
1,7,10
8
0.,20.,30.,45.,60.,90.,135.,180.
2000.,1800.,1600.,1400.,1200.,1000.,800.,600.
8
0.,20.,30.,45.,60.,90.,135.,180.
2000.,1800.,1600.,1400.,1200.,1000.,800.,600.
8
0.,20.,30.,45.,60.,90.,135.,180.
1500.,1400.,1300.,1200.,1100.,1000.,800.,600.
8
0.,20.,30.,45.,60.,90.,135.,180.
1500.,1400.,1300.,1200.,1100.,1000.,800.,600.
8
0.,20.,30.,45.,60.,90.,135.,180.
1000.,900.,800.,700.,600.,500.,400.,300.
8
0.,20.,30.,45.,60.,90.,135.,180.
1000.,900.,800.,700.,600.,500.,400.,300.

```

LISTING A-2C SAMPLE INPUT SET 3 FOR PAM

JOB CARD. CHANGE THIS CARD TO CONFORM TO INSTALLATION USAGE.
 ACCOUNT CARD. CHANGE THIS CARD TO CONFORM TO INSTALLATION USAGE.
 COMMENT. THIS RUNSTREAM IS USED WITH THE FIRST RUN OF THE PAM PROGRAM;
 THAT IS, THE RUN IN WHICH TAPE11 IS FIRST CREATED.
 COMMENT. THE NEXT STATEMENT DECLARES TAPES TO BE A FILE OF RECORD
 TYPE 2 WITH 80 CHARACTER LINES. THIS IS EQUIPPED SINCE TAPES
 COMMENT. IS A FILE FROM THE "FRONT END" COMPUTER WHICH USES A DIFFERENT
 COMMENT. DEFAULT FILE TYPE FROM THE "BACK END" COMPUTER ON WHICH THIS
 COMMENT. JOB IS TO BE RUN.
 FILE TAPES, PT=2, FL=80, BT=C.
 COMMENT. THE NEXT STATEMENT OBTAINS A COPY OF THE PAM INPUT FROM THE
 COMMENT. "FRONT END" COMPUTER AND PUTS IT IN THE LOCAL FILE TAPES.
 GET PT, TAPES, PAM SAMPLE INPUT, ID=SANDMEYER, ST=MTA.
 COMMENT. THE NEXT STATEMENT ATTACHES THE MACHINE LANGUAGE ("OBJECT
 COMMENT. CODE") CREATED BY RUNNING THE PAM SOURCE CODE THROUGH THE
 COMMENT. FORTRAN COMPILER IN A PREVIOUS JOB.
 ATTACHALGO, PAM SAMPLE BINARY, ID=SANDMEYER.
 COMMENT. THE NEXT STATEMENT RESERVES SPACE ON A PERMANENT FILE DEVICE
 COMMENT. FOR THE FILE WHOSE LOCAL NAME IS TAPE11.
 REQUEST, TAPE11, MPE.
 COMMENT. THE NEXT STATEMENT LOADS AND EXECUTES THE PAM PROGRAM USING
 COMMENT. TAPES AS THE INPUT FILE.
 GO TO TAPES)
 COMMENT. THE NEXT STATEMENT CATALOGS TAPE11 AS A PERMANENT FILE.
 CATALOG, TAPE11, SAMPLE TAPE11, ID=SANDMEYER.
 COMMENT. END OF JOB.

LISTING A-3A SAMPLE RUNSTREAM 1 FOR PAM


```

JOB CAPT. CHANGE THIS CARD TO CONFORM TO INSTALLATION USAGE.
ACCOUNT CARD. CHANGE THIS CARD TO CONFORM TO INSTALLATION USAGE.
COMMENT. THIS RUNSTREAM IS USED WITH RUNS OF THE PAM PROGRAM MADE AFTER
TAP11 HAS BEEN CREATED.
COMMENT. THE NEXT STATEMENT DECLARES TAP5 TO BE A FILE OF RECORD
TYPE Z WITH 80 CHARACTER LINES. THIS IS REQUIRED SINCE TAP5
COMMENT. IS A FILE FROM THE "PRINT FID" COMPUTER WHICH USES A DIFFERENT
COMMENT. DEFAULT FILE TYPE FROM THE "BACK FID" COMPUTER ON WHICH THIS
COMMENT. JOB IS TO BE RUN.
FILE TAP5,PT=Z,F=80,BT=C.
COMMENT. THE NEXT STATEMENT OBTAINS A COPY OF THE PAM INPUT FROM THE
COMMENT. "FROM FID" COMPUTER AND PUTS IT IN THE LOCAL FILE TAP5.
GETF TAP5,PAM,SAMPLEINPUT2,ID=SOURCE,ST=DEF.
COMMENT. THE NEXT STATEMENT ATTACHES THE MACHINE LANGUAGE ("OBJECT
COMMENT. CODE") CREATED BY RUNNING THE PAM SOURCE CODE THROUGH THE
COMMENT. FORTRAN COMPILER IN A PREVIOUS JOB.
ATTACH2 GO,PAM,AMBLEBINARY,ID=SOURCE.
COMMENT. THE NEXT STATEMENT ATTACHES THE ALREADY EXISTING TAP11 ("CODE
COMMENT. DATA BASE" FILE) WHICH WILL BE MODIFIED ("EXTENDED") BY THIS
COMMENT. RUN. THE MULTI-READ OPTION IS TURNED OFF SO THAT THE FILE CAN
COMMENT. BE EXTENDED.
ATTACH TAP11,SAMPLETAP11,ID=SOURCE,MP=0.
COMMENT. THE NEXT STATEMENT LOADS AND EXECUTES THE PAM PROGRAM USING
COMMENT. TAP5 AS THE INPUT FILE.
LGOUTAP5)
COMMENT. THE NEXT STATEMENT EXTENDS TAP11. THIS MEANS THAT THE
COMMENT. CHANGES MADE BY THIS RUN ARE NOW INCORPORATED IN THE PERMANENT
COMMENT. FILE THAT WAS ATTACHED UNDER LOCAL FILE NAME TAP11.
EXTEND TAP11.
COMMENT. END OF JOB.

```

LISTING A-3B SAMPLE RUNSTREAM 2 FOR PAM

JOB CAPC. CHANGE THIS CARD TO CONFORM TO INSTALLATION USAGE.
ACCOUNT CARD. CHANGE THIS CARD TO CONFORM TO INSTALLATION USAGE.
COMMENT. THIS RUNSTREAM IS USED WITH PUNS OF THE PAM PROGRAM MADE AFTER
COMMENT. TAPE11 HAS BEEN CREATED.
COMMENT. THE NEXT STATEMENT DECLARES TAPES TO BE A FILE OF RECORD
COMMENT. TYPE 2 WITH 90 CHARACTER LINES. THIS IS REQUIRED SINCE TAPES
COMMENT. IS A FILE FROM THE "FRONT END" COMPUTER WHICH USES A DIFFERENT
COMMENT. DEFAULT FILE TYPE FROM THE "BACK END" COMPUTER ON WHICH THIS
COMMENT. JOB IS TO BE RUN.
FILE, TAPES, RT=Z, FL=80, BT=C.
COMMENT. THE NEXT STATEMENT OBTAINS A COPY OF THE PAM INPUT FROM THE
COMMENT. "FRONT END" COMPUTER AND PUTS IT IN THE LOCAL FILE TAPES.
GETPE, TAPES, PAMSAMPLE INPUT, ID=SANDBYEP, ST=WEA.
COMMENT. THE NEXT STATEMENT ATTACHES THE MACHINE LANGUAGE ("OBJECT
COMMENT. CODE") CREATED BY RUNNING THE PAM SOURCE CODE THROUGH THE
COMMENT. FORTRAN COMPILER IN A PREVIOUS JOB.
ATTACH, JCM, PAMSAMPLE INAP, ID=SANDBYEP.
COMMENT. THE NEXT STATEMENT ATTACHES THE ALREADY EXISTING TAPE11 ("CORE
COMMENT. DATA BASE" FILE) WHICH WILL BE MODIFIED ("EXTENDED") BY THIS
COMMENT. PUN. THE MULTI-HEAD OPTION IS TURNED OFF SO THAT THE FILE CAN
COMMENT. BE EXTENDED.
ATTACH, TAPE11, SAMPLE TAPE11, ID=SANDBYEP, HP=0.
COMMENT. THE NEXT STATEMENT LOADS AND EXECUTES THE PAM PROGRAM USING
COMMENT. TAPES AS THE INPUT FILE.
LG01(TAPES)
COMMENT. THE NEXT STATEMENT EXTENDS TAPE11. THIS MEANS THAT THE
COMMENT. CHANGES MADE BY THIS PUN ARE NOW INCORPORATED IN THE PERMANENT
COMMENT. FILE THAT WAS ATTACHED UNDER LOCAL FILE NAME TAPE11.
EXTEND, TAPE11.
COMMENT. END OF JOB.

LISTING A-3C SAMPLE RUNSTREAM 3 FOR PAM

	COLUMN NOS.

.890000	1.000000	1.000000	.890000
.890000	.990000	.990000	.890000
.910000	.990000	.990000	.910000
.880000	1.000000	1.000000	.880000
.900000	.990000	.990000	.900000
.920000	1.000000	1.000000	.920000

[illegible][illegible]

1	DELTA TIME = 30.0000	RME	NP	SEVERAL TARGETS = 1
1	1.9000	1.0000	1.0000	1.0000
2	1.9000	1.0000	1.0000	1.0000
3	1.9000	1.0000	1.0000	1.0000
4	1.9000	1.0000	1.0000	1.0000
5	1.9000	1.0000	1.0000	1.0000
6	1.9000	1.0000	1.0000	1.0000
7	1.9000	1.0000	1.0000	1.0000
8	1.9000	1.0000	1.0000	1.0000
9	1.9000	1.0000	1.0000	1.0000
10	1.9000	1.0000	1.0000	1.0000
11	1.9000	1.0000	1.0000	1.0000
12	1.9000	1.0000	1.0000	1.0000
13	1.9000	1.0000	1.0000	1.0000
14	1.9000	1.0000	1.0000	1.0000
15	1.9000	1.0000	1.0000	1.0000
16	1.9000	1.0000	1.0000	1.0000
17	1.9000	1.0000	1.0000	1.0000
18	1.9000	1.0000	1.0000	1.0000
19	1.9000	1.0000	1.0000	1.0000
20	1.9000	1.0000	1.0000	1.0000
21	1.9000	1.0000	1.0000	1.0000
22	1.9000	1.0000	1.0000	1.0000
23	1.9000	1.0000	1.0000	1.0000
24	1.9000	1.0000	1.0000	1.0000
25	1.9000	1.0000	1.0000	1.0000
26	1.9000	1.0000	1.0000	1.0000
27	1.9000	1.0000	1.0000	1.0000
28	1.9000	1.0000	1.0000	1.0000
29	1.9000	1.0000	1.0000	1.0000
30	1.9000	1.0000	1.0000	1.0000
31	1.9000	1.0000	1.0000	1.0000
32	1.9000	1.0000	1.0000	1.0000
33	1.9000	1.0000	1.0000	1.0000
34	1.9000	1.0000	1.0000	1.0000
35	1.9000	1.0000	1.0000	1.0000
36	1.9000	1.0000	1.0000	1.0000
37	1.9000	1.0000	1.0000	1.0000
38	1.9000	1.0000	1.0000	1.0000
39	1.9000	1.0000	1.0000	1.0000
40	1.9000	1.0000	1.0000	1.0000
41	1.9000	1.0000	1.0000	1.0000
42	1.9000	1.0000	1.0000	1.0000
43	1.9000	1.0000	1.0000	1.0000
44	1.9000	1.0000	1.0000	1.0000
45	1.9000	1.0000	1.0000	1.0000
46	1.9000	1.0000	1.0000	1.0000
47	1.9000	1.0000	1.0000	1.0000
48	1.9000	1.0000	1.0000	1.0000
49	1.9000	1.0000	1.0000	1.0000
50	1.9000	1.0000	1.0000	1.0000
51	1.9000	1.0000	1.0000	1.0000
52	1.9000	1.0000	1.0000	1.0000
53	1.9000	1.0000	1.0000	1.0000
54	1.9000	1.0000	1.0000	1.0000
55	1.9000	1.0000	1.0000	1.0000
56	1.9000	1.0000	1.0000	1.0000
57	1.9000	1.0000	1.0000	1.0000
58	1.9000	1.0000	1.0000	1.0000
59	1.9000	1.0000	1.0000	1.0000
60	1.9000	1.0000	1.0000	1.0000
61	1.9000	1.0000	1.0000	1.0000
62	1.9000	1.0000	1.0000	1.0000
63	1.9000	1.0000	1.0000	1.0000
64	1.9000	1.0000	1.0000	1.0000

DEFICIATION PARGE =	1	DELTA TIME = 1.0,000	ONE DP	SEVERAL TARGETS = 1
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.3000	38000	38000	14000	14000
0.3000	38000	38000	14000	14000
0.3600	36000	36000	15000	15000
0.3600	36000	36000	15000	15000
0.4000	40000	40000	10000	10000
0.4000	40000	40000	10000	10000
0.4200	42000	42000	07000	07000
0.4200	42000	42000	07000	07000
0.4000	40000	40000	12000	12000
0.4000	40000	40000	12000	12000

[illegible]

[illegible][illegible]

DESIGNATION RANGE= 5 DELTA TIME= 30.0000 MIN OF SEVERAL TARGETS = 1

DESIGNATION RANGE 5 DELTA TIME 00.0000 DUE IN SEVERAL TARGETS - 1

* DELTA TIME=150,000 ONE OF SEVERAL TARGETS * 1

DESIGNATION RANGE = 5 DELTA TIME = 300,000 OMT OF SEVERAL TARGETS = 1

CIN UPMN MODS.

LISTING A-4 SAMPLE OUTPUT FROM PAM

(PAGE 8 OF 11)

[illegible]

DESIGNATION RANGE = 6 DELTA TIME = 0.0000 ONE OR SEVERAL TARGETS = 1

```

DESIGNATION PAGE= 6 DELTA TIME= 0.0000 DIF OF SEVERAL TARGETS = 2

```

DESIGNATION RANGE = 6 DELTA TIME = 30,000 THE NO. SEVERAL TARGETS = 1

DESIGNATION RANGE= 6 DELTA TIME= 90.0000 ONE OF SEVERAL TARGETS = 1

[illegible]

LISTING A-4 SAMPLE OUTPUT FROM PAM

(PAGE 9 OF 11)

	COLUMN NOS.

[illegible][illegible][illegible][illegible][illegible]

APPENDIX B

LISTING AND SAMPLE CASE OF PRCLOS

APPENDIX B

LISTING AND SAMPLE CASE OF PRBLOS

This appendix contains:

- (1) A listing of the FORTRAN code for the PRBLOS program.
- (2) One set of sample inputs for PRBLOS.
- (3) One sample runstream for PRBLOS.
- (4) Sample output created by running PRBLOS with the sample input.

Note: The column numbers included in some of the listings in this appendix are not a part of the program code, the data, or the output, but are included only for the reader's convenience.


```

JOB CARD. CHANGE TO CONFORM TO INSTALLATION USAGE.
ACCOUNT CARD. CHANGE TO CONFORM TO INSTALLATION USAGE.
COMMENT. THE FOLLOWING CARD ATTACHES THE MACHINE LANGUAGE (OBJECT CODE) FILE
COMMENT. OF THE PPBLOS PROGRAM. THIS FILE WAS CREATED BY PUNCHING THE SOURCE
COMMENT. LISTING OF PPBLOS THROUGH THE EDITOR COMPILER IN A
COMMENT. PREVIOUS JOB.
ATTACH LGD,PPBLOS,INAP,IO=SAIDHFYEP.
COMMENT. THE NEXT CARD DECLARES TAPES TO BE A FILE OF PCOPYD TYPE 2 WITH
COMMENT. 80 CHARACTER RECORDS. THIS CARD IS NECESSARY BECAUSE TAPES IS TO
COMMENT. BE OBTAINED FROM THE "FRONT END" COMPUTER WHICH USES A DIFFERENT
COMMENT. DEFAULT RECORD TYPE FOR ITS FILES.
FILE,TAPES,RT=Z,FIL=80,BT=C.
COMMENT. THE NEXT CARD OBTAINS A COPY OF TAPES (PPBLOS INPUT FILE) FROM
COMMENT. THE "FRONT END" MACHINE.
GETPF,TAPES,PPBLOS,SAMPLE INPUT,IO=SAIDHFYEP,ST=MFA.
COMMENT. THE NEXT CARD LOADS AND EXECUTES THE PROGRAM USING TAPES AS INPUT.
LEND(TAPES)

```

LISTING B-3 SAMPLE RUNSTREAM FOR PPBLOS

CASE: CLOSE TERRAIN: RANGE BAND FROM ZFPO TO 2000 METERS
 TGVFL= 3.0000 CT= 15.0000
 PPIV APPAY: NIV= 11
 0.0000 .1000 .2000 .3000 .4000 .5000 .6000 .7000
 .8000 .9000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
 25.0000 30.0000 40.0000 50.0000 70.0000 100.0000 120.0000 140.0000
 250.0000 500.0000 1100.0000
 PPRGV APPAY: NODV= 12
 0.0000 .6100 .7430 .8000 .8500 .8740 .9000
 .9100 .9500 .9800 1.0000 1.0000 1.0000 1.0000 1.0000
 0.0000 25.0000 50.0000 75.0000 100.0000 120.0000 140.0000 175.0000
 200.0000 400.0000 600.0000 800.0000 1000.0000
 PPIVH APPAY: NIVH= 9
 .2500 .3000 .4000 .5000 .6000 .7000 .8000 .9000
 1.0000 1.0000 25.0000 75.0000 135.0000 235.0000 455.0000
 1045.0000

AVERAGE IN-VIEW SEGMENT LENGTH= 197.2500
 AVERAGE OUT-OF-VIEW SEGMENT LENGTH= 72.6750
 AVERAGE MODIFIED IN-VIEW SEGMENT LENGTH= 155.1250

PRINDE* .5740

END OF CASE

LISTING B-4 SAMPLE OUTPUT FROM PRBLOS
 (PAGE 1 OF 3)

CASE: CLOSE TERRAIN: PANGE BAND FROM 2000 TO 2500 METERS
 16TVEL: 3.0000 CT= 15.0000
 PP1V APPAY: WIV= 11
 0.0000 .1000 .2000 .3000 .4000 .5000 .6000 .7000
 .8000 .9000 1.0000
 25.0000 30.0000 40.0000 50.0000 70.0000 120.0000 150.0000 200.0000
 470.0000 500.0000 1520.0000
 PP00V AF PAY: N00V= 13
 0.0000 .6100 .7430 .8000 .8500 .8740 .9000
 .9100 .9500 .9800 .9900 1.0000
 0.0000 25.0000 50.0000 75.0000 100.0000 125.0000 150.0000 175.0000
 200.0000 400.0000 600.0000 800.0000 1000.0000
 PP1VH APPAY: W1VH= 9
 .2500 .3000 .4000 .5000 .6000 .7000 .8000 .9000
 1.0000
 0.0000 5.0000 25.0000 75.0000 105.0000 155.0000 375.0000 855.0000
 1475.0000

AVERAGE IN-VIEW SEGMENT LENGTH= 275.2500
 AVERAGE OUT-OF-VIEW SEGMENT LENGTH= 72.6750
 AVERAGE MODIFIED IN-VIEW SEGMENT LENGTH= 233.1250

PPBLOS= .6702

END OF CASE

LISTING B-4 SAMPLE OUTPUT FROM PPBLOS
 (PAGE 2 OF 3)

CASE: CLOSE TEPATH: PANGE BAND FROM 2500 TO 9900 METERS									
TTOTVL-	3.0000	CT-	15.0000						
PPIV APPAY: NIV-	11								
0.0000	.1000	.2000	.3000	.4000	.5000	.6000	.7000		
.8000	.9000	1.0000							
25.0000	40.0000	60.0000	100.0000	140.0000	180.0000	210.0000	240.0000		
310.0000	590.0000	1600.0000							
PPROV APPAY: MODV-	13								
0.0000	.6100	.7430	.8000	.8500	.8650	.8740	.9000		
.9100	.9500	.9800	.9900	1.0000					
0.0000	25.0000	50.0000	75.0000	100.0000	125.0000	150.0000	175.0000		
200.0000	400.0000	600.0000	800.0000	1000.0000					
PPIVM APPAY: NIVH-	10								
.1250	.2000	.3000	.4000	.5000	.6000	.7000	.8000		
.9000	1.0000								
0.0000	15.0000	55.0000	95.0000	145.0000	165.0000	195.0000	245.0000		
545.0000	1555.0000								

AVERAGE IN-VIEW SEGMENT LENGTH= 269.2500
 AVERAGE OUT-OF-VIEW SEGMENT LENGTH= 72.5750
 AVERAGE MODIFIED IN-VIEW SEGMENT LENGTH= 225.5675

PPRIMS= .8590

END OF CASE

LISTING R-4 SAMPLE OUTPUT FROM PRBLOS
 (PAGE 3 OF 3)

APPENDIX C

LISTING AND SAMPLE CASE OF PREPMS

APPENDIX C

LISTING AND SAMPLE CASE OF PREPMS

This appendix contains:

(1) A listing of the FORTRAN code for the PREPMS program. In addition to the routines in this listing, PREPMS also uses the functions IDCHAR and NUMRIC and the subroutine PENAME which are included in the COPE listing in Appendix D.

(2) One set of sample inputs for PREPMS.

(3) One sample runstream for PREPMS.

(4) Sample output created by running PREPMS with the sample input (and with the records created by the three sample cases of PAM already written to TAPE 11).

Note: The column numbers included in some of the listings in this appendix are not a part of the program code, the data, or the output, but are included only for the reader's convenience.

[illegible]

170 CONTINUE

```

NOPTNH(NOPTI)=NP(AD
CALL WRITHS (LL,CIAPA,ICUPL,CLARL,-1)
CALL WRITHS (LL,NOPTNH,70,RNOPTNHMS,-1)
GO TO 100

```

```

1800 PLABFL=R(1,2)
    IF (.NOT.NUMERIC(R(10,3))) STOP , ' IN PROGRAM: FPROP NUMBER 6 ,
    DECODE (10,60,R(1,3))ZCASE
    ZCASE=ZCASE
    WRITE (6,700) PLABFL,CASE

```

GN IN 1100, 240, 200, 320, 350, 370, 430, 460, 500), CRASE

1900 200 1911 1911

```

2000 DATA(I)=0.0
      READ (5,620) NCC,NVL
      WRITE (6,650) NCC,NVL
      READ (5,630) PRCCLS(I),I=1,2)
      WRITE (6,660) PRCCLS(I),I=1,2)

```

```

00 210 J=1,2
      READ (5,630) (PGCFL(I,J),I=1,11)
      210 WRITE (6,660) (PGCFL(I,J),I=1,11)

```

00 220 K=1.92
00 220 J=1.11

```

READ (5,6303) (WLS(J,K),J=1,6)
C20 WRITE (6,6600) (WLS(J,K),J=1,6)
READ (5,6303) (WIS(I),I=1,NVL)
READ (5,6303) (CLQUD(I),I=1,6)
WRITE (6,6600) (WIS(I),I=1,NVL)
WRITE (6,6600) (CLQUD(I),I=1,NCC)
READ (5,6303) (PASQL(I),I=1,6)
WRITE (6,6600) (PASQL(I),I=1,6)

```

```

PEAD (5,630) (WINDSPD(I,J),I=1,3)
WRITE (6,660) (WINDSPD(I,J),I=1,3)

```

```

PFAD (5,630) (HMMID(I),I=1,2)
WRITE (6,660) (HMMID(I),I=1,2)
CALL WPITMS (II,VOATA,TP(TI,PLAPEI,-1)
GO TO 530

```

240 D0 250 I=1,IP1 Y2

2-C ACQUAINTANCE
READ (5,620) HPP,HPP,HPPGCL
WHITE (6,650) HPP,HPP,HPPGCL
HPPGCL=HPPGCL+1

```

PEAD 5,630) (CFMGD(I,J),I=1,NPF)
WRITE (6,660) (CPMGD(I,J),I=1,NPF)
DN 270 J=1,NPMP
PEAD 5,630) (SFCMS(I,J),I=1,NPF)
WRITE (6,660) (SFCMS(I,J),I=1,NPF)

```

[illegible]

```

270 CONTINUE
   READ (5,630) (RNGCLB(I),I=1,NPNGCL)
   WRITE (6,660) (PHGCLB(I),I=1,NPHGCL)
   DO 280 J=1,2
     READ (5,630) (VFLTBL(I,J),I=1,2)
     WRITE (6,660) (VFLTBL(I,J),I=1,2)
280 CONTINUE
   CALL WRITMS (I1,ACQDAT,IBL2,FLABFL,-1)
   GO TO 530

290 DO 300 I=1,IBLT3
300 READAT(I)=0.0
   READ (5,620) NDT
   WRITE (6,650) NDT
   DO 310 J=1,2
     READ (5,630) (DETTMA(I,J),I=1,NDT)
     WRITE (6,660) (DETTMA(I,J),I=1,NDT)
310 CONTINUE
   READ (5,630) (BATPTM(I),I=1,3),(XWADF(I),I=1,2)
   WRITE (6,660) (BATPTM(I),I=1,3),(XWADF(I),I=1,2)
   READ (5,630) (XCSGTH(I,J),I=1,2),(XHTSTM(I,J),I=1,2),J=1,2
   WRITE (6,660) (XCSGTH(I,J),I=1,2),(XHTSTM(I,J),I=1,2),J=1,2
   READ (5,630) ((THEAH(I,J),I=1,2),(TSIGMA(I,J),I=1,2),J=1,2)
   WRITE (6,660) ((THEAH(I,J),I=1,2),(TSIGMA(I,J),I=1,2),J=1,2)
   READ (5,630) ((TMEAH(I,J),I=1,2),(TSIGMA(I,J),I=1,2),J=1,2)
   WRITE (6,660) ((TMEAH(I,J),I=1,2),(TSIGMA(I,J),I=1,2),J=1,2)
   READ (5,630) (TPAPFY(I,J),I=1,3),(J=1,2)
   WRITE (6,660) (TPAPFY(I,J),I=1,3),(J=1,2)
   CALL WRITMS (I1,ESPDAT,IBLT3,FLABFL,-1)
   GO TO 530

320 DO 330 I=1,IBLT4
330 DFSDAT(I)=0.0
   READ (5,620) NDFS
   WRITE (6,650) NDFS
   DO 340 J=1,3
     READ (5,630) (DEFNOVL(I,J),I=1,NDFS)
     WRITE (6,660) (DEFNOVL(I,J),I=1,NDFS)
340 CONTINUE
   CALL WRITMS (I1,DFSDAT,IBLT4,FLABFL,-1)
   GO TO 530

350 DO 360 I=1,IBLT5
360 RDATA(I)=0.0
   READ (5,620) HPLNCP
   WRITE (6,650) TCFIT
   READ (5,630) (PHGFLC(I),I=1,NPHNCP)
   WRITE (6,660) (PHGFLC(I),I=1,NPHNCP)
   READ (5,630) (HPLNCP)
   WRITE (6,650) TCFIT
   WRITE (6,660) (PHGFLC(I),I=1,NPHNCP)
   WRITE (6,660) (HPLNCP)
   GO TO 530

```

[illegible]

LISTING C-1 FORTRAN LISTING OF PREPMS PROGRAM - CONT'D
(PAGE 4 OF 12)

00000000011111111122222223333333334444444555555566666667777777788888889 COLUMN NUMBERS
1234567890123456789012345678901234567890123456789012345678901234567890 (READ VERTICALLY)

```

C
CALL WRITMS (11,PODATA,IPLT5,PLAB(L,-1))
GO TO 530

370 DD 380 I=1,IPLT6
380 PDATA(1)=0.0
UNIT=5
READ (5,610) (A(I),I=1,80)
CALL SEPRC2 (A,80,8,10)
IF (B(1),1)-NF,8HMS(TAPF4) GO TO 390
UNIT=4
WRITE (6,600)
GO TO 400
390 ENCODE (20,610,B(1,1)) (A(I),I=1,70)
DECODE (20,670,B(1,1)) (INDEX(1,1),K)=1,2)
400 CONTINUE
DD 410 I=1,6
I=I+1
IF (I,60,1,AND,IU*IT,I,0,5) I=I+1
DD 410 J=I*START,5
READ (I,UNIT,670) (INDEX(I,J),K)=1,2)
WRITE (6,650) (INDEX(I,J),K)=1,2)
410 CONTINUE
DD 420 I=1,60
DD 420 J=1,10
READ (I,UNIT,630) (PTRL(I,J),K)=1,7)
WRITE (6,660) (PTRL(I,J),K)=1,7)
420 CONTINUE
GO TO 530

C
430 DD 440 I=1,IPLT7
440 PDATA(1)=0.0
READ (5,620) NPHGPS
WRITE (6,650) NPHGPS
READ (5,630) (PHGPS(I),I=1,NPHGPS)
WRITE (6,640) (PHGPS(I),I=1,NPHGPS)
DD 450 J=1,2
READ (5,640) (PSTR(I,J),I=1,NPHGPS)
WRITE (6,640) (PSTR(I,J),I=1,NPHGPS)
450 CONTINUE
CALL WRITMS (11,PODATA,IPLT7,PLAB(L,-1))
GO TO 5

C
460 DD 470 I=1,IPLT8
470 AINVA(1)=0.0
READ (5,620) NPHGCTT
WRITE (6,650) NPHGCTT
DD 480 J=1,NPHGCTT
READ (5,620) (PHGCTT(I,J),I=1,NPHGCTT)
WRITE (6,660) (PHGCTT(I,J),I=1,NPHGCTT)
480 CONTINUE
DD 490 J=1,2
READ (5,630) (PSTR(I,J),I=1,NPHGCTT)
WRITE (6,640) (PSTR(I,J),I=1,NPHGCTT)
490 CONTINUE
CALL WRITMS (11,PODATA,IPLT8,PLAB(L,-1))
GO TO 5

```

00000000011111111122222223333333334444444555555566666667777777788888889 COLUMN NUMBERS
1234567890123456789012345678901234567890123456789012345678901234567890 (READ VERTICALLY)

LISTING C-1 FORTRAN LISTING OF PREPMS PROGRAM - CONT'D

[illegible]

```

SUBROUTINE SEPRC2 (A,H1,B,H2,H3)
      COMMON /LOGLG/ FIRSTL, SEQNMU, SPCL(70), SHRTEC, FTSILL
      LOGICAL FIRSTL, SEQNMU, SPCL, SHRTEC, FTSILL
      COMMON /SYMBOL/ BLANK, RECDUT, AFTB(127), ANUMPP(11), SEP(5),
1 DOLLAR
      DIMENSION A(H1), P(H2,H3), JH(5,10)
      DIMENSION NC(R0), PR(R0,10), IC(R0,10), NUMFLG(10)
      DO 110 I=1,10
        NUMFLG(I)=0
        DO 100 J=1,80
          IC(J,I)=0
100 CONTINUE
110 CONTINUE
120 NC(I)=0
      DO 130 J=1,H2
        DO 140 I=1,80
          J=J+1
          IF (IC(J,I).NE.0) GO TO 230
          IF (NC(I).NE.0) GO TO 230
          GO TO 210
130 CONTINUE
140 NC(I)=DOCHAR(A,H1,J)
      IF (.NOT.FIRSTL) GO TO 200
      FIRST=.FALSE.
      DO 170 I=73,78
        IF (NC(I).NE.0) GO TO 230
170 CONTINUE
      DO 180 I=63,72
        IF (NC(I).NE.3) GO TO 230
180 CONTINUE
      DO 190 I=79,80
        IF (NC(I).NE.3) GO TO 230
190 CONTINUE
      SEQNMU=.TRUE.
      WRITE (6,330)
      WRITE (6,340)
      GO TO 210
200 IF (.NOT.SEQNMU) GO TO 230
210 DO 220 I=73,78
      NC(I)=3
220 CONTINUE
230 CONTINUE
      IF (SEP(5).EQ.0) INPUT CAPD.
      I=1

```

[illegible]

LISTING C-1 FORTRAN LISTING OF PREPMS PROGRAM - CONT'D

(PAGE 10 OF 12)


```

00000000011111112222222333333344444455555566666677777788888899 C COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (PEAD VERTICALLY)

      SUBROUTINE NMPFC?
      C
      C   WRITE (6,100)
      C
      C   RETURN
      C
      C *** F O R M A T   S T A T I S T I C S ***
      C
      C 100 FORMAT (12H **PPPP** ,44H THIS IS A CPATION PUM FOR PECUPD OPTN036720 SR-NOREC11
      C      1HUMS )
      C      END
      C

00000000011111112222222333333344444455555566666677777788888899 C COLUMN NUMBERS
1234567890123456789012345678901234567890123456789012345678901234567890 (PEAD VERTICALLY)

```

LISTING C-1 FORTRAN LISTING OF PREPMS PROGRAM - CONT'D

(PAGE 12 OF 12)

	COLUMB MDS.

[illegible]

101-99
MFW PFC, DFC 1400021, 1

C-18

Variable	Mean	Standard deviation	Skewness	Kurtosis
1. Age	63.7	2.2	.69	.03
2. Sex	0.5	0.2	.023	.04
3. Education	0.7	0.34	.221	.067
4. Income	0.3	0.03	.007	.013
5. Health	0.2	0.03	.013	.023
6. Social desirability	0.1	0.07	.003	.003
7. Self-esteem	0.1	0.07	.007	.023
8. Life satisfaction	0.1	0.07	.007	.023
9. Depression	0.1	0.07	.007	.023
10. Anxiety	0.1	0.07	.007	.023
11. Stress	0.1	0.07	.007	.023
12. Loneliness	0.1	0.07	.007	.023
13. Social support	0.1	0.07	.007	.023
14. Coping strategies	0.1	0.07	.007	.023
15. Resilience	0.1	0.07	.007	.023
16. Life events	0.1	0.07	.007	.023
17. Personality traits	0.1	0.07	.007	.023
18. Cognitive functioning	0.1	0.07	.007	.023
19. Emotional stability	0.1	0.07	.007	.023
20. Overall well-being	0.1	0.07	.007	.023

Variable	Mean	SD	Min	Max
Age	34.5	10.2	18	65
Gender	0.52	0.50	0	1
Marital Status	0.68	0.47	0	1
Education	12.3	2.1	9	16
Income	45.2	15.8	20	80
Health Status	0.75	0.43	0	1
Employment	0.81	0.39	0	1
Stress Level	3.2	1.5	1	5
Life Satisfaction	4.1	1.2	2	5
Resilience	3.8	1.4	2	5
Optimism	4.3	1.1	3	5
Gratitude	4.5	1.0	3	5
Forgiveness	4.2	1.1	3	5
Empathy	4.4	1.0	3	5
Compassion	4.3	1.1	3	5
Kindness	4.5	1.0	3	5
Generosity	4.2	1.1	3	5
Patience	4.1	1.2	3	5
Self-control	4.3	1.1	3	5
Emotional Stability	4.4	1.0	3	5
Psychological Well-being	4.6	0.9	3	5
Life Purpose	4.7	0.8	3	5
Meaning in Life	4.8	0.7	3	5
Existential Well-being	4.9	0.6	3	5
Transcendental Well-being	5.0	0.5	3	5

[illegible]

LISTING C-2 SAMPLE INPUT FOR PREPMS - CONT'D
(PAGE 4 OF 15)

	COLUMN NOS.

4,2,4
5,2,5
6,2,6
7,2,7
8,2,8
9,2,9
10,2,10
11,2,11
12,2,12
13,2,13
14,2,14
15,2,15
16,2,16
17,2,17
18,2,18
19,2,19
20,2,20
21,2,21
22,2,22
23,2,23
24,2,24
25,2,25
26,2,26
27,2,27
28,2,28
29,2,29
30,2,30
31,2,31
32,2,32
33,2,33
34,2,34
35,2,35
36,2,36
37,2,37
38,2,38
39,2,39
40,2,40
41,2,41
42,2,42
43,2,43
44,2,44
45,2,45
46,2,46
47,2,47
48,2,48
49,2,49
50,2,50
51,2,51
52,2,52
53,2,53
54,2,54
55,2,55
56,2,56
57,2,57
58,2,58
59,2,59
60,2,60
61,2,61
62,2,62
63,2,63
64,2,64
65,2,65
66,2,66
67,2,67
68,2,68
69,2,69
70,2,70
71,2,71
72,2,72
73,2,73
74,2,74
75,2,75
76,2,76
77,2,77
78,2,78
79,2,79
80,2,80
81,2,81
82,2,82
83,2,83
84,2,84
85,2,85
86,2,86
87,2,87
88,2,88
89,2,89
90,2,90
91,2,91
92,2,92
93,2,93
94,2,94
95,2,95
96,2,96
97,2,97
98,2,98
99,2,99
100,2,100

COLUMN NOS.

LISTING C-2 SAMPLE INPUT FOR PREPMS - CONT'D
(PAGE 8 OF 15)

AD-A100 285

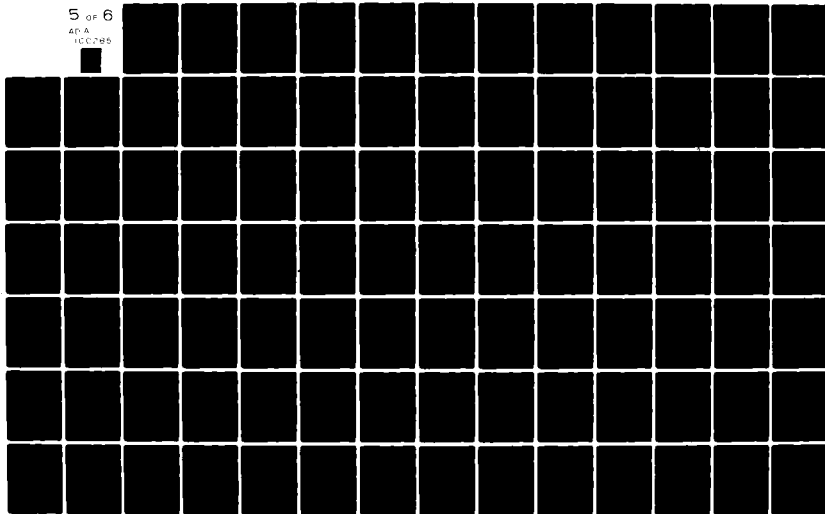
ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY ABERDEEN PROV--ETC F/6 19/1
COPPERHEAD OPERATIONAL PERFORMANCE EVALUATION (COPE): COMPUTER --ETC(U)
MAR 81 R S SANDMEYER
ANSAA-TR-318

UNCLASSIFIED

NL

5 OF 6

AT-A
100285



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524

[illegible][illegible]

LISTING C-2 SAMPLE INPUT FOR PREPMS - CONT'D
(PAGE 11 OF 15)

(PAGE 11 OF 15)

[illegible]

	1.	2.	3.	4.	5.	6.	7.	8.
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0
57	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0
59	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0	0
62	0	0	0	0	0	0	0	0
63	0	0	0	0	0	0	0	0
64	0	0	0	0	0	0	0	0
65	0	0	0	0	0	0	0	0
6								

[illegible]

[illegible]

.0	.007	.0	.007	.0	.01
.007	.017	.031	.034	.270	.179
2.	3.	11.	4.	5.	6.
9.	10.	11.			
3304.	457.	610.	762.	914.	1372.
.04	.21	.29	.46	.0	.0
1.	.43	1.			
.0	.32	.0			
.0	.25	.0			
.0	.453				
NEWFC,SIF?000Z1,1					
6	11				
.602	.22	.49		.024	.0
.013	.0	.003	.01	.0	.04
.047	.071	.384		.01	.003
.020	.0	.003	.0		.07
.017	.04	.279			
.0	.0	.0	.0	.0	.013
.0	.0	.0	.0	.0	.0
.0	.0	.003	.0	.0	.0
.003	.0	.0	.0	.0	.01
.003	.0	.0	.0	.0	.02
.0	.0	.0	.0	.0	.0
.003	.0	.0	.0	.0	.037
.0	.0	.0	.0	.0	.01
.0	.0	.003	.0	.0	.044
.007	.0	.0	.0	.01	.061
.003	.0	.0	.0	.013	.367
.0	.0	.0	.0	.003	.013
.003	.0	.0	.0	.0	.0
.003	.0	.0	.0	.0	.0
.007	.0	.0	.0	.003	.0
.007	.0	.003	.0	.0	.003
.007	.0	.0	.003	.0	.01
.007	.0	.0	.0	.0	.003
.0	.0	.0	.0	.0	.01
.0	.003	.010	.013	.007	.034
1.	2.	3.	4.	5.	6.
9.	10.	11.			
304.	477.	610.	762.	914.	1372.
.0	.0	.0	.48	.13	.30
1.	.7	1.			
.0	.3	.0			
.0	.0	.0			
.043	.057				
OPTMPS,M,14,21					
1-2,JUNF					
1,2,J					

LISTING C-2 SAMPLE INPUT FOR PREPMS - CONT'D
(PAGE 14 OF 15)


```

JOB CARD. CHANGE THIS CARD TO CONFORM TO INSTALLATION USAGE.
ACCOUNT CARD. CHANGE THIS CARD TO CONFORM TO INSTALLATION USAGE.
COMMENT. THIS RUNSTREAM IS USED WITH PING OF THE PREPMS PROGRAM NAME
COMMENT. AFTER TAPE11 HAS BEEN CREATED.
COMMENT. THE NEXT STATEMENT DECLARES TAPES TO BE A FILE OF RECORD
COMMENT. TYPE 7 WITH 80 CHARACTER LINES. THIS IS REQUIRED SINCE TAPES
COMMENT. IS A FILE FROM THE "FRONT END" COMPUTER WHICH USES A DIFFERENT
COMMENT. DEFAULT FILE TYPE FROM THE "BACK END" COMPUTER ON WHICH THIS
COMMENT. JOB IS TO BE RUN.
FILE,TAPES,RT,Z,FL=80,BT=C.
COMMENT. THE NEXT STATEMENT OBTAINS A COPY OF THE PREPMS INPUT FROM THE
COMMENT. "FRONT END" COMPUTER AND PUTS IT IN THE LOCAL FILE TAPES.
GETPT,TAPES,PREPMSAMPLEINPUT,ID=SAIDMEYER,ST=MTA.
COMMENT. THE NEXT STATEMENT ATTACHES THE MACHINE LANGUAGE ("OBJECT
COMMENT. CODE") CREATED BY RUNNING THE PREPMS SOURCE CODE THROUGH THE
COMMENT. FORTRAN COMPILER IN A PREVIOUS JOB.
ATTACH,LOG,PREPMSAMPLESTNARY,ID=SAIDMEYER.
COMMENT. THE NEXT STATEMENT ATTACHES THE ALREADY EXISTING TAPE11 ("CODE
COMMENT. DATA BASE" FILE) WHICH WILL BE MODIFIED ("EXTENDED") BY THIS
COMMENT. RUN. THE MULTI-READ OPTION IS TURNED OFF SO THAT THE FILE CAN
COMMENT. BE EXTENDED.
ATTACH,TAPE11,SAMPLETAPE11,ID=SAIDMEYER,MP=0.
COMMENT. THE NEXT STATEMENT LOADS AND EXECUTES THE PREPMS PROGRAM USING
COMMENT. TAPES AS THE INPUT FILE.
LOG(TAPES)
COMMENT. THE NEXT STATEMENT EXTENDS TAPE11. THIS MEANS THAT THE
COMMENT. CHANGES MADE BY THIS RUN ARE NOW INCORPORATED IN THE PERMANENT
COMMENT. FILE THAT WAS ATTACHED UNDER LOCAL FILE NAME TAPE11.
EXTEND,TAPE11.
COMMENT. END OF JOB.

```

LISTING C-3 SAMPLE RUNSTREAM FOR PREPMS

[illegible]

***** THIS IS A CREATION FOR PROPER OPTIMIZATION *****

NEW RECORD, AINVD0128, 8
RECORD NAMED : AINVD0128 OF TYPE 8 NOW BEING PRAD

91
107[illegible][illegible]

NEW RFCNPD, JUN0600021, 1

NEW RELEASED UNDER E.O. 13526
PFCOP: NAMED : JUN0600021 OF TYPE 1 NEW RING READ

6 11

6 11

[illegible][illegible]

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524

NEW PFC, DEC14, 00021, 1										
PFC001 NAMEIN : DEC14, 00021 OF TYPE 1 MINW BFTNG, PFCAD										
6 11										
.0400	.0070	.0070	.0070	.0000	.0070	.0000	.0000	.0030	.0170	
.0200	.0000	.0000	.0030	.0000	.0030	.0000	.0000	.0030	.0030	
.0230	.0030	.0030	.0000	.0000	.0000	.0000	.0000	.0000	.0100	
.0200	.0030	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0100	
.0460	.0100	.0000	.0200	.0170	.0000	.0000	.0000	.0030	.0300	
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	
.0430	.0030	.0100	.0000	.0070	.0000	.0000	.0000	.0170	.0230	
.0030	.0030	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0100	
.0170	.0030	.0100	.0000	.0000	.0000	.0000	.0000	.0030	.0300	
.0100	.0030	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0300	
.0070	.0000	.0070	.0000	.0170	.0000	.0000	.0000	.0200	.1490	
.0100	.0030	.0000	.0000	.0030	.0030	.0000	.0000	.0030	.0030	
.0100	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	
.0030	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	
.0070	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0070	
.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	
.0070	.0000	.0000	.0000	.0070	.0000	.0000	.0000	.0030	.0000	
.0200	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0030	.0000	
.0000	.0030	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	
.0100	.0070	.0000	.0000	.0000	.0000	.0000	.0000	.0030	.0070	
.0260	.0230	.0130	.0100	.0100	.0100	.0260	.0400	.0400	.0400	
1.0000	2.0000	3.0000	4.0000	4.0000	4.0000	4.0000	6.0000	6.0000	6.0000	
9.0000	10.0000	11.0000	11.0000	762.0000	762.0000	914.0000	1372.0000	1372.0000	2600	
304.0000	457.0000	610.0000	0.0000	0.0000	0.0000	.6600	.0000	.0000	.0000	
0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
1.0000	.6000	2.0000	0.0000	.0030	.0030	.0000	.0000	.0000	.0000	
0.0000	.0000	.0030	.0000	.0000	.0000	.0000	.0000	.0000	.0000	
.0100	.0070	.0000	.0000	.0000	.0000	.0000	.0000	.0030	.0070	
.0100	.0230	.0130	.0100	.0100	.0100	.0260	.0400	.0400	.0400	
1.0000	2.0000	3.0000	4.0000	4.0000	4.0000	4.0000	6.0000	6.0000	6.0000	
9.0000	10.0000	11.0000	11.0000	762.0000	762.0000	914.0000	1372.0000	1372.0000	2600	
304.0000	457.0000	610.0000	0.0000	0.0000	0.0000	.6600	.0000	.0000	.0000	
0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
1.0000	.6000	2.0000	0.0000	.0030	.0030	.0000	.0000	.0000	.0000	
0.0000	.0000	.0030	.0000	.0000	.0000	.0000	.0000	.0000	.0000	
.0100	.0070	.0000	.0000	.0000	.0000	.0000	.0000	.0030	.0070	
.0100	.0230	.0130	.0100	.0100	.0100	.0260	.0400	.04		

[illegible]

LISTING C-4 SAMPLE OUTPUT FROM PREPMS - CONT'D
(PAGE 4 OF 17)

SDN MNM 103

	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000
9.0000	10.0000	11.0000						
94.0000	95.0000	96.0000	97.0000	98.0000	99.0000	100.0000	101.0000	102.0000

0.0000 1.0000

NEW PFC, ACQDAT0122,?
RECOPC NAMED : ACQDAT0122 OF TYPE 2 NOW BEING PFAD

[illegible]

3.0000 0000.2

NEW FIC, ACQDAT0222,2
PFCODE NAMEO : ACQDAT0222 OF TYPE 2 NEW BEING PLACD

11	11	3							
0,000	0,000	1,000	2,000						
6,000	6,000	9,000	1,000						
0,000	400,000	450,000	450,000						
3000,000	3700,000	7900,000							
0,000		1,000	2,000						
8,000		3,000	1,000						
25,000	30,000	40,000							
260,000	500,000	1100,000							
25,000	30,000	40,000							
420,000	900,000	1570,000							
25,000	40,000	60,000							
310,000	590,000	1600,000							
2000,000	7,000,000	9999,000							
2,000	2,000								
2,000	2,000								
0,000	0,000	1,000	2,000						
0,000	400,000	450,000	450,000						
3000,000	3700,000	7900,000							
0,000		1,000	2,000						
8,000		3,000	1,000						
25,000	30,000	40,000							
260,000	500,000	1100,000							
25,000	30,000	40,000							
420,000	900,000	1570,000							
25,000	40,000	60,000							
310,000	590,000	1600,000							
2000,000	7,000,000	9999,000							
2,000	2,000								
2,000	2,000								
0,000	0,000	1,000	2,000						
0,000	400,000	450,000	450,000						
3000,000	3700,000	7900,000							
0,000		1,000	2,000						
8,000		3,000	1,000						
25,000	30,000	40,000							
260,000	500,000	1100,000							
25,000	30,000	40,000							
420,000	900,000	1570,000							
25,000	40,000	60,000							
310,000	590,000	1600,000							
2000,000	7,000,000	9999,000							
2,000	2,000								
2,000	2,000								
0,000	0,000	1,000	2,000						
0,000	400,000	450,000	450,000						
3000,000	3700,000	7900,000							
0,000		1,000	2,000						
8,000		3,000	1,000						
25,000	30,000	40,000							
260,000	500,000	1100,000							
25,000	30,000	40,000							

[illegible]

LISTING C-4 SAMPLE OUTPUT FROM PREPMS - CONT'D

(PAGE 6 OF 17)

CPU MM MS.

[illegible]

COL UHM NDS.

LISTING C-4 SAMPLE OUTPUT FROM PREPMS - CONT'D

(PAGE 10 OF 17)

APPENDIX D

LISTING AND SAMPLE CASE OF COPE

APPENDIX D

LISTING AND SAMPLE CASE OF COPE

This appendix contains:

- (1) A listing of the FORTRAN code for the COPE program.
- (2) One set of sample case inputs for COPE.
- (3) One sample runstream for COPE.
- (4) Sample output created by running COPE with the sample input (and with the TAPE11 created by previous PAM and PREPMS sample case runs). The output produced on TAPE8 by this run (for cases 1 and 2) is included in Listings 7-1 and 7-2 of section 7.6.

Note: The column numbers included in some of the listings in this appendix are not a part of the program code, the data, or the output, but are included only for the reader's convenience.

7	5,*,*,*,1,0HPL0T	,7*1H,			003000	RD-R0ATA55
8	6,*,0,*,1,0H0VPR0T	,7*1H,			004000	BU-R0ATA54
9	6,*,0,*,1,0H0VPR0T	,7*1H,			004010	BU-R0ATA54
10	7,*,0,*,1,0HTEMPAPY	,7*1H,			004020	PD-R0ATA55
11	7,*,0,*,1,0HTEMPAPY	,7*1H,			004030	PD-R0ATA56
12	8,*,0,*,1,0HPEST	,7*1H,			004040	RD-R0ATA57
13	8,*,0,*,1,0HPEST	,7*1H,			004050	RD-R0ATA58
14	9,*,*,*,1,0H0NTRPL	,7*1H,			004060	EP-R0ATA59
15	DATA ((CHAR(I),J),I=1,10),J=14,*,3)/				004070	RD-R0ATA60
16	1 11,*,*,*,1,0H10DFCTH,1,0H10MTR	,6*1H,			004080	RD-R0ATA61
17	1 11,*,*,*,1,0H10DFCTH,1,0H10MTR	,7*1H,			004090	RD-R0ATA62
18	3 12,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,6*1H,			004100	PD-R0ATA63
19	4 12,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004110	PD-R0ATA64
20	5 13,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004120	PD-R0ATA65
21	6 13,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004130	PD-R0ATA66
22	7 14,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004140	RD-R0ATA67
23	8 16,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004150	RD-R0ATA68
24	9 16,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004160	RD-R0ATA69
25	17 17,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004170	RD-R0ATA70
26	DATA ((CHAR(I),J),I=1,10),J=24,*,3)/				004180	RD-R0ATA71
27	1 21,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004190	RD-R0ATA72
28	2 21,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004200	RD-R0ATA73
29	3 22,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004210	RD-R0ATA74
30	4 22,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004220	RD-R0ATA75
31	5 23,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004230	RD-R0ATA76
32	6 23,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004240	RD-R0ATA77
33	7 24,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004250	PD-R0ATA78
34	8 24,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004260	PD-R0ATA79
35	9 25,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004270	RD-R0ATA80
36	10 25,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004280	RD-R0ATA81
37	11 26,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004290	RD-R0ATA82
38	12 27,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004300	PD-R0ATA83
39	13 27,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004310	PD-R0ATA84
40	14 27,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004320	RD-R0ATA85
41	15 28,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004330	PD-R0ATA86
42	DATA ((CHAR(I),J),I=1,10),J=30,*,3)/				004340	RD-R0ATA87
43	1 31,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004350	RD-R0ATA88
44	2 32,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004360	RD-R0ATA89
45	3 32,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004370	RD-R0ATA90
46	DATA ((CHAR(I),J),I=1,10),J=40,*,5)/				004380	PD-R0ATA91
47	1 41,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,6*1H,			004390	PD-R0ATA92
48	2 41,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004400	PD-R0ATA93
49	3 42,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,6*1H,			004410	PD-R0ATA94
50	4 42,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004420	PD-R0ATA95
51	5 43,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,6*1H,			004430	PD-R0ATA96
52	6 43,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004440	PD-R0ATA97
53	7 44,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,6*1H,			004450	PD-R0ATA98
54	8 44,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,7*1H,			004460	PD-R0ATA99
55	9 45,*,*,*,1,0H10MIS1,0H10MIS,1,0H1	,6*1H,			004470	RD-R0ATA100
56	10 45					

[illegible]


```

0000000011111112222223333333334444445555566666777777788888899 COLUMN NUMBERS
12345678901234567890123456789012345678901234567890 (PEAD VERTICAL Y)

FUNCTION GAMMA (ALPHA, IPH)
  DIMENSION WK(6), F(1)
  IPH=1
  CALL CGAMA (IPH, ALPHA, TH, WP, P)
  GAMMA=F(1)
  PETIPIH
  END

0000000011111112222223333333334444445555566666777777788888899 COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (READ VERTICAL Y)

```

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 10 OF 82)

0000000001111111112222222233333333444444445555555566666666777777778888888899999999
 1734567890123456789012345678901234567890123456789012345678901234567890 (PEAD VERTICALLY)

```

C      FUNCTION IDCHAP (A,N,I)
      COMMON /SYMBOL/ BIANP, DECPNT, ALPHET(27), ANUMPR(11), SEP(5),
      1 DOLLAR
      DIMENSION A(N)
      IF (A(1)-HE-BLANK) GO TO 100
      IDCHAP=3
      RETURN
      100 IF (A(1)-HE-DECPNT) GO TO 110
      IDCHAP=2
      RETURN
      110 IF (A(1)-HE-ANUMPR(J)) GO TO 120
      IDCHAP=0
      RETURN
      120 CONTINUE
      DO 130 J=1,5
      IF (A(1)-HE-SEP(J)) GO TO 130
      IDCHAP=4
      RETURN
      130 CONTINUE
      DO 140 J=1,77
      IF (A(1)-HE-ALPHET(J)) GO TO 140
      IDCHAP=1
      RETURN
      140 CONTINUE
      IF (A(1)-HE-DOLLAR) GO TO 150
      IDCHAP=5
      RETURN
      150 IDCHAP=6
      RETURN
      END
  
```

0000000001111111112222222233333333444444445555555566666666777777778888888899999999
 1734567890123456789012345678901234567890123456789012345678901234567890 (PEAD VERTICALLY)

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
 (PAGE 11 OF 82)


```

0000000001111111122222222333333334444444555555566666667777777888888899 C COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (RFAD VERTICALLY)

C SUBROUTINE ABPTL (IABPT)
006890 SP-ABPT112
006900 SP-ABPT113
000110 CH-ABPT 2
000300 CH-FLAG 2
000310 CH-FLAG 3
000320 CH-FLAG 4
000770 CH-PLCTN2
006940 SP-ABPT117
006950 SP-ABPT118
006960 SP-ABPT119
006970 SP-ABPT110
006980 SP-ABPT111
006990 SP-ABPT112
007000 SP-ABPT113
007010 SP-ABPT114
007020 SP-ABPT115
007030 SP-ABPT116
007040 SP-ABPT117
007050 SP-ABPT118
007060 SP-ABPT119
007070 SP-ABPT120
007080 SP-ABPT121

COMMON /ABORT/ HARBPT(20,6), HSARN
COMMON /FLAG/ ENDPFP, INH, IPFPL, IDIGTL, IPPTY, IRNDNC, IT3FLG,
1 ENDFLG
LOGICAL ENDFLG, ENDPFP
COMMON /PLCTN/ KPFP, NPFP, KPF, NPF, NVEHVL

C IPF=1
IF (IABPT.LE.HSARLN) GO TO 100
HARBPT(IABPT,KPF)=HARBPT(IABPT,KPF)+1
GO TO 120

C 100 DO 110 IJ=1EF,NPF
HARBPT(IABPT,IJ)=HARBPT(IABPT,IJ)+1
110 CONTINUE

C 120 ENDPFP=.TRUE.

C RETURN

C END

0000000001111111122222222333333334444444555555566666667777777888888899 C COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (RFAD VERTICALLY)

```

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 14 OF 82)

D-20

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 18 OF 82)

[illegible][illegible]

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 22 OF 82)

[illegible]

C	COMMON /DVALUE/ I:(70), DPRT, DDT, DIDM, DIDIGT, DNMW (?)	009990	SP-DF AUL T1
	COMMON /PUNDA/ DAT, TIM, HCASE, ICCPFG, IPPFG	010000	SP-DF AUL T2
C	ICCPFG=0	000260	CM+DVALUE2
	IPPF=0	000840	CM+PUNDA2
	D111)=1.	010030	SP-DF AUL T6
	D112)=1.	010040	SP-DF AUL T7
	D113)=1.	010050	SP-DF AUL T8
	D114)=1.	010060	SP-DF AUL T9
	D115)=1.	010070	SP-DF AUL T10
	D116)=1.	010080	SP-DF AUL T11
	D117)=1.	010090	SP-DF AUL T12
	D118)=1.	010100	SP-DF AUL T13
	D119)=1.	010110	SP-DF AUL T14
	D120)=1.	010120	SP-DF AUL T15
	D121)=1.	010130	SP-DF AUL T16
	D122)=1.	010140	SP-DF AUL T17
	D123)=1.	010150	SP-DF AUL T18
	D124)=1.	010160	SP-DF AUL T19
	D125)=1.	010170	SP-DF AUL T20
	D126)=1.	010180	SP-DF AUL T21
	D127)=1.	010190	SP-DF AUL T22
	D128)=1.	010200	SP-DF AUL T23
	D129)=1.	010210	SP-DF AUL T24
	D130)=1.	010220	SP-DF AUL T25
	D131)=1.	010230	SP-DF AUL T26
	D132)=1.	010240	SP-DF AUL T27
	D133)=1.	010250	SP-DF AUL T28
	D134)=1.	010260	SP-DF AUL T29
	D135)=1.	010270	SP-DF AUL T30
	D136)=1.	010280	SP-DF AUL T31
	D137)=1.	010290	SP-DF AUL T32
	D138)=1.	010300	SP-DF AUL T33
	D139)=1.	010310	SP-DF AUL T34
	D140)=1.	010320	SP-DF AUL T35
	D141)=1.	010330	SP-DF AUL T36
	D142)=1.	010340	SP-DF AUL T37
	D143)=1.	010350	SP-DF AUL T38
	D144)=1.	010360	SP-DF AUL T39
	D145)=1.	010370	SP-DF AUL T40
	D146)=1.	010380	SP-DF AUL T41
	D147)=1.	010390	SP-DF AUL T42
	D148)=1.	010400	SP-DF AUL T43
	D149)=1.	010410	SP-DF AUL T44
	D150)=1.	010420	SP-DF AUL T45
	D151)=1.	010430	SP-DF AUL T46
	D152)=1.	010440	SP-DF AUL T47
	D153)=1.	010450	SP-DF AUL T48
	D154)=1.	010460	SP-DF AUL T49
	D155)=1.	010470	SP-DF AUL T50
	D156)=1.	010480	SP-DF AUL T51
	D157)=1.	010490	SP-DF AUL T52
	D158)=1.	010500	SP-DF AUL T53
	D159)=1.	010510	SP-DF AUL T54
	D160)=1.	010520	SP-DF AUL T55

[illegible]

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D

(PAGE 25 OF 82)

D-28

(PAGE 26 OF 82)

0000000011111111222222233333334444445555556666667777777788888899 COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (PEAD VERTICALLY)

```

110 CONTINUE
WRITE (IUNIT,220) (MIDSPD(I,J),I=1,3)
WRITE (IUNIT,220) (MIDIDAT(I),I=1,2)
WRITE (IUNIT,210) (MIDAPP,MIDGCL)
WRITE (IUNIT,240) ((CPRCH(I,J),I=1,11),J=1,2)
WRITE (IUNIT,240) ((STGLDS(I,J),I=1,11),J=1,11)
WRITE (IUNIT,240) ((MIDCLR(I),I=1,11)
DO 120 J=1,2
WRITE (IUNIT,220) (VELTBL(I,J),I=1,2)
120 CONTINUE
WRITE (IUNIT,210) (IDUALPPER,I,IGTL,IPPLY,IPHOC,IT3FLC)
WRITE (IUNIT,220) BLONG
WRITE (IUNIT,250) ((DFOCK(I,J),I=1,10),J=1,3)
WRITE (IUNIT,250) (ANGLET,SCIRG,PELDOFF,GTNR,SKSEN)
IF (IOVER(26),NE.2) WRITE (IUNIT,180) PRCOME
IF (SHRTEC) GO TO 150
DO 130 I=1,6
WRITE (IUNIT,260) (INDEX(I,J,K),K=1,2),J=1,5)
130 CONTINUE
DO 140 I=1,60
WRITE (IUNIT,270) (I,((PTTBL(I,J,K),K=1,7),J=1,10)
140 CONTINUE
150 CONTINUE
WRITE (IUNIT,210) HPGTT
DO 160 J=1,20
WRITE (IUNIT,220) PNCIT(J),((ITF(I,J,K),K=1,2),I=1,3)
WRITE (IUNIT,280) (IDLY(I),I=1,6)
WRITE (IUNIT,280) ((PWTBL(I,J),I=1,10),J=1,2)
WRITE (IUNIT,210) HPGPS
WRITE (IUNIT,280) (FNGPS(I),I=1,10)
WRITE (IUNIT,280) ((PTTBL(I,J),I=1,10),J=1,3)
WRITE (IUNIT,220) (BATPTBL(I),I=1,3),((XMAOF(I),I=1,2)
WRITE (IUNIT,220) ((RCS,TBL(I,J),I=1,2),J=1,2),((XMTIN(I,J),I=1,2)
1,J=1,2)
WRITE (IUNIT,220) ((TIMEAN(I,J),I=1,2),J=1,2),((TSIGNAL(I,J),I=1,2),
1,J=2,2)
WRITE (IUNIT,220) ((ALPHA(I,J),I=1,2),J=1,2),((RETA(I,J),I=1,2),
1,J=2,2)
WRITE (IUNIT,210) HD1
WRITE (IUNIT,280) ((NETTHA(I,J),I=1,10),J=1,2)
WRITE (IUNIT,220) ((TPAPRY(I,J),I=1,3),J=1,2)
WRITE (IUNIT,220) (PSPTM,TPRAP,PRCNT,PRCVI,PCPMSC,PPDOW,PUPC)
WRITE (IUNIT,210) IDSGTC
IF (IPNOC.EQ.0) GO TO 170
WRITE (IUNIT,220) TCFIT
WRITE (IUNIT,210) HPIOSP
WRITE (IUNIT,280) (PHGPI(I),I=1,10)
WRITE (IUNIT,280) (PRALG(I),I=1,10)
170 CONTINUE
WRITE (IUNIT,200) HCA5F
C
C
C *** F D F M A T S T A T I S T I C S ***

```

0000000011111111222222233333334444445555556666667777777788888899 COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (PEAD VERTICALLY)

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 30 OF 82)


```

000000000111111111222222223333333344444444555555566666667777777788888889 COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (READ VERTICALLY)

```

```

C 190 FORMAT (1H,28H NAME 'IF PDATA RECORD USED',1X,A10)
190 FORMAT (1H,34H TEMPORARY OPTION USED FOR PDATA )
200 FORMAT (1H,13(4H) - 1,21H DATA CHECK FOR CAST ,13,13(4H) - )
210 FORMAT (1H,16(15)
220 FORMAT (1H,8F10.4)
230 FORMAT (1H,7F10.4,110,F10.4,110)
240 FORMAT (1H,11F10.4)
250 FORMAT (1H,10F10.4)
260 FORMAT (1H,18HINDEX APPAY FOR 1,15,/(215))
270 FORMAT (1H,15HPE ARRAY FOR 1,15,/(1H,7F10.4))
280 FORMAT (1H,10F10.4)
290 FORMAT (1H,8(6H*****),27H END OF DATA CHECK FOR CASE,14,1X,8(6H*****),27H
1*****))
END

```

```

000000000111111111222222223333333344444444555555566666667777777788888889 COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (READ VERTICALLY)

```

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 31 OF 82)

[illegible]

(PAGE 37 OF 82)


```

IF (IOWEP(21),HF,0) GO TO 160
DECODE (10,760,IVALUE(1,5)) XHTHP,YHTHP
ENCODE (40,770,CNUT(1,5)) XHTHP
ENCODE (40,780,CNUT(1,6)) YHTHP
NXLTIN=7
GO TO 170
160 IF (IOWEP(21),FQ,1) ENCODE (40,760,CNUT(1,5)) IVALUE(21)
IF (IOWEP(21),FQ,2) ENCODE (40,800,CNUT(1,6))
NXLTIN=6
170 ENCODE (40,430,CNUT(1,NXLTIN+1))
ENCODE (40,420,CNUT(1,NXLTIN+1))
ENCODE (40,420,CNUT(1,NXLTIN+3))
ENCODE (40,430,CNUT(1,NXLTIN+4))
FHCDE (40,810,CNUT(1,NXLTIN+5))
ENCODE (40,430,CNUT(1,NXLTIN+6))
IF (IOWEP(22),NE,0) GO TO 180
DECODE (10,820,IVALUE(22)) JTPPH
ENCODE (40,830,CNUT(1,NXLTIN+7)) JACQBI (J,ITPPH),I=1,2)
GO TO 190
180 IF (IOWEP(22),FQ,1) ENCODE (40,840,CNUT(1,NXLTIN+7)) IVALUE(22)
IF (IOWEP(22),FQ,2) ENCODE (40,850,CNUT(1,NXLTIN+7))
190 ENCODE (40,860,CNUT(1,NXLTIN+8)) CPQBI (I,FPHNGC+1),I=1,4)
NXLTIN=NXLTIN-1
ENCODE (40,430,CNUT(1,NXLTIN+10))
ENCODE (40,420,CNUT(1,NXLTIN+11))
ENCODE (40,420,CNUT(1,NXLTIN+13))
ENCODE (40,430,CNUT(1,NXLTIN+14))
ENCODE (40,870,CNUT(1,NXLTIN+15))
ENCODE (40,430,CNUT(1,NXLTIN+16))
FHCDE (40,380,CNUT(2,NXLTIN+17)) FSPJHI
IF (ISPL(32)) GO TO 200
NXLTIN=NXLTIN+1
ENCODE (40,880,CNUT(1,NXLTIN+17)) HSNW2,HSNW5
200 CONTINUE
ENCODE (40,890,CNUT(1,NXLTIN+18)) FPCSTI
ENCODE (40,430,CNUT(1,NXLTIN+19))
ENCODE (40,420,CNUT(1,NXLTIN+20))
ICMFLG=0
IF (IDH,FQ,1,AND,YHTHP,FQ,4H2200) ICMFLG=1
IF (IDH,FQ,2,AND,YHTHP,HF,4H2200) ICMFLG=1
IF (ICMFLG,FQ,0) GO TO 210
FHCDE (100,360,CH(1)) DAVLPL (IDH),YHTHP
FHCDE (100,360,CH(1)) JTPPL(1)
CALL CONHNT (CM,10,FALSE,FALSE)
210 CONTINUE
DO 230 I=1,8
IF (IOWEP(20),FQ,0) GO TO 240
IF (IOWEP(20),FQ,1) GO TO 270
IF (IOWEP(20),HF,2) STOP, IN HFAHFP: FPCDF HIMPFP 1 *
ENCODE (100,360,CH(1)) JTPPL(1)
CALL CONHNT (CM,10,FALSE,FALSE)
2150
SP-HEAD155
016620
016630
016640
016650
016660
016670
016680
016690
016700
016710
016720
016730
016740
016750
016760
016770
016780
016790
016800
016810
016820
016830
016840
016850
016860
016870
016880
016890
016900
016910
016920
016930
016940
016950
016960
016970
016980
016990
017000
017010
017020
017030
017040
017050
017060
017070
017080
017090
017100
017110
017120
017130
017140
017150
SP-HEAD156
SP-HEAD157
SP-HEAD158
SP-HEAD159
SP-HEAD160
SP-HEAD161
SP-HEAD162
SP-HEAD163
SP-HEAD164
SP-HEAD165
SP-HEAD166
SP-HEAD167
SP-HEAD168
SP-HEAD169
SP-HEAD170
SP-HEAD171
SP-HEAD172
SP-HEAD173
SP-HEAD174
SP-HEAD175
SP-HEAD176
SP-HEAD177
SP-HEAD178
SP-HEAD179
SP-HEAD180
SP-HEAD181
SP-HEAD182
SP-HEAD183
SP-HEAD184
SP-HEAD185
SP-HEAD186
SP-HEAD187
SP-HEAD188
SP-HEAD189
SP-HEAD190
SP-HEAD191
SP-HEAD192
SP-HEAD193
SP-HEAD194
SP-HEAD195
SP-HEAD196
SP-HEAD197
SP-HEAD198
SP-HEAD199
SP-HEAD200
SP-HEAD201
SP-HEAD202
SP-HEAD203
SP-HEAD204
SP-HEAD205
SP-HEAD206

```

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 42 OF 82)

0000000001111111222222333333444444555555666666777777888888999999 C O L U M N N U M B E R S
123456789012345678901234567890123456789012345678901234567890 (PEAD VERTICALLY)

```

C      SUBROUTINE INITLZ
      COMMON /ABORT/ HAPORT(20,6), HSAIRH
      COMMON /ACHAR/ CHAR(10,120), HPTC
      COMMON /DVALUE/ D(70), DPT, DDT, DIRH, DDIGT, DSHW(2)
      COMMON /FLAG/ ENDFLG, INH, IPPFL, IDIGT, IPPTY, IPHODC, IT3FLG,
1     ENDFLG
      LOGICAL ENDFLG, ENDFEP
      COMMON /ICNDF/ ICNDF
      COMMON /INDEX1/ INDEX1(12001)
      COMMON /RUNDAT/ DAT, TIN, HCASE, ICCPTG, IPPFG
      COMMON /UVALUE/ U(70), UPT, UDT, UJPH, UIDIGT, USMK(2)
      DIMENSION IPATH(70)
      EXTERNAL HNPFC
      EQUIVALENCE (ICNDF,CNDF)
      IPAY(1)=0
      IFAY(5)=LCDF(HNPFC)
      CALL SYSTFHC (106,IPAY)
      CALL OPFIMS (11,INDEX1,70,1)
      ICNDF=888888888
      CALL READMS (11,IPATH,70,ICNDF)
      DO 100 I=1,NPEC
      J=CHAR(1,I)
      CHAR(2,I)=IPATH(J)
100 CONTINUE
      CALL DEAUT
      CALL USET
      HSAIRH=11
      ENDFLG=.FALSE.
      RETURN
      END

```

0000000001111111222222333333444444555555666666777777888888999999 C O L U M N N U M B E R S
123456789012345678901234567890123456789012345678901234567890 (PEAD VERTICALLY)

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 46 OF 82)

[illegible]

```

DIMENSION A(80), B(8,10), IT(10)
DIMENSION INC0(70), CH(10)
DIMENSION VALU(70), DVALU(70)

EQUIVALENCE (ICORR,CORR)
EQUIVALENCE (U(1),VALU(1)), (U(1),DVALU(1))
EQUIVALENCE (WDATA(1,3), ACORAT(161), P*PDAT(40), DFSDAT(31))
DIMENSION PDATAT(2), PDATAT(4260), PDATAT(41), AINUDA(167)
EQUIVALENCE (WDATAT(1),PRCLC)
EQUIVALENCE (ACORAT(1),RPP)
EQUIVALENCE (P*PDAT(1),BATPTM(1))
EQUIVALENCE (DFSDAT(1),DFDML(1))
EQUIVALENCE (PDATAT(1),TCPT)
EQUIVALENCE (P*PDAT(1),INDEX(1))
EQUIVALENCE (P*SDAT(1),RPPG)
EQUIVALENCE (AINUDA(1),DLT(1))

ICAT(1)=0
IPAT(1)=LOC(INDFC)
CALL SYSTEMC (104,IPAT)

DO 110 J=1,13
  CORRUE (J)=BLANK
  DO 100 J=1,70
    CORRUE (J)=BLANK
  100 CORRUE (J)=BLANK
  110 CORRUE
  120 CORRUE=0

  ICHMT=0
  NLPTC=0
  NTPC=0
  MCAF=MEASE+1

  WRITE (8,770) MCAF

  DO 130 READ-A-LINE-OF-INPUT UNTIL "END", "END", OR EOF ENCOUNTERED
  130 READ (5,870) (A(I),I=1,80)
    IF (EOF(5).EQ.0) GO TO 140
    ENDFLAG=.TRUE.
    IF (NLPTC.NE.0) STOP + " INPUT: EOF ENCOUNTERED "
    ENCODE (100,MPO,CMT)
    CALL COMMIT (CMT,10,.FALSE.,.FALSE.)
    GO TO 490

  140 WRITE (8,880) (A(I),I=1,80)

  IF (A(1).NE.1) GO TO 150
  CALL COMMIT (A,80,.TRUE.,.FALSE.)
  GO TO 130

```

0000000000111111112222222233333333444444445555555566666666777777778888888899 COLIMN NUMBERS

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D

(PAGE 48 OF 82)

[illegible][illegible]

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D

(PAGE 49 OF 82)

[illegible]

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 51 OF 82)

```

021220 SP-INPU235
021230 SP-INPU236
021240 SP-INPU237
021250 SP-INPU238
021260 SP-INPU239
021270 SP-INPU240
021280 SP-INPU241
021290 SP-INPU242
021300 SP-INPU243
021310 SP-INPU244
021320 SP-INPU245
021330 SP-INPU246
021340 SP-INPU247
021350 SP-INPU248
021360 SP-INPU249
021370 SP-INPU250
021380 SP-INPU251
021390 SP-INPU252
021400 SP-INPU253
021410 SP-INPU254
021420 SP-INPU255
021430 SP-INPU256
021440 SP-INPU257
021450 SP-INPU258
021460 SP-INPU259
021470 SP-INPU260
021480 SP-INPU261
021490 SP-INPU262
021500 SP-INPU263
021510 SP-INPU264
021520 SP-INPU265
021530 SP-INPU266
021540 SP-INPU267
021550 SP-INPU268
021560 SP-INPU269
021570 SP-INPU270
021580 SP-INPU271
021590 SP-INPU272
021600 SP-INPU273
021610 SP-INPU274
021620 SP-INPU275
021630 SP-INPU276
021640 SP-INPU277
021650 SP-INPU278
021660 SP-INPU279
021670 SP-INPU280
021680 SP-INPU281
021690 SP-INPU282
021700 SP-INPU283
021710 SP-INPU284
021720 SP-INPU285
021730 SP-INPU286
021740 SP-INPU287
021750 SP-INPU288

```

[illegible]

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D

(PAGE 52 OF 82)

[illegible]

Figure 1

D-56


```

0000000001111111222222233333333444444455555556666667777777888888899 COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (PEAD VERTICALLY)

C      CALL FCHD (8)
C      PETUPH
C
C      C * * * F O R M A T   S T A T E M E N T S   * * *
C
770 FORMAT (1H,50(11H*),1X,23HINPUT OPTIONS FOR CASE ,13,1X,50(11H*))
780 FORMAT (A3,7X)
790 FORMAT (1H,23H FPRDR IN INPUT. LINE: ,80A1,11H WITH I1= ,15,/,13022930 SP-INPU405
      1H IS IN ERROR.)
800 FORMAT (A6,12,2,2H27)
810 FORMAT (1H,18HYOUR INPUT LINE: ,1X,80A1)
820 FORMAT (1H,58HUPSTERSDES ONE OF YOUR PREVIOUS INPUT LINES FOR THI022970 SP-INPU410
      15 CASE )
830 FORMAT (6X,12,2X)
840 FORMAT (A3,4,13,3)
850 FORMAT (A6,12,2,12,2)
860 FORMAT (A5,11,3HTRN,11)
870 FORMAT (80A1)
880 FORMAT (1H,80A1)
890 FORMAT (1H,71HND ENDF CAPD FOUND BUT ONE ASSUMED SINCE ENDF OF IMP023050 SP-INPU418
      1UT FILE ENCOUNTERED )
900 FORMAT (1H,3HND ,12H*FIRST CASE*,4H OR ,11H*NEXT CASE*,14H CARD F023070 SP-INPU420
      10HND---,15H* NEXT CASE,PD,16H OPTION ASSUMED )
910 FORMAT (F10,4)
      END

0000000001111111222222233333333444444455555556666667777777888888899 COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (PEAD VERTICALLY)

```

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D

(PAGE 55 OF 82)

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 56 OF 82)


```

000000001111111122222222333333334444444455555555666666667777777788888889 COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (READ VERTICALLY)

      OUTLINE NMPFC
      IN /ICODF/ ICODF
      WRITE (6,100) ICODF
      CALL STPAC
      STOP * IN NMPFC: ATTEMPTED TO READ NON-EXISTENT RECORD *
      * * * F O R M A T   S T A T E M E N T S   * * *
      100 FORMAT (1H,38H INPUT ATTEMPTED TO READ RECORD NAMED ,A10,30H RUI
      1 1H SUCH RECORD WAS FOUND)
      END

000000001111111122222222333333334444444455555555666666667777777788888889 COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (READ VERTICALLY)

```

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 58 OF 82)

[illegible]

```

290 FORMAT (56H SEPTUUS PROBLEM 1 XTST AND NTEST APPAYS DO NOT ACCEPTE
1, 4H IT=, 4H J=, 15)
300 FORMAT (1H 24H IN OUT, PROBLEM 1 J=, 15, 9H PNOFF=, 15, 3, 8H
11LX, 15)
310 FORMAT (1H 12H NTEST APPAYS=, 15, 1H 6A10, 10)
320 FORMAT (1H 14X 14H CAUST OF ARRET)
330 FORMAT (50X, 5H ***** 1X 10H OCCASIONS, 1H (, 15, 1H 6H *****
240 FORMAT (1H 15, 1H 12, 3, 3A10, 17, 1H, A7, 5X, 25H NMBRF TESTED (%
1ED)=, 17, 1H, A7)
350 FORMAT (1H 7X, 14H NMBRF NMBRF, 23, 3H, 14X, 14H, 14X, 14X, 14X, 14X,
14X, 105, 14X, 1106)
360 FORMAT (1H 15, 1H 12, 3X, 3A10, 6, 17, 1H, A7)
370 FORMAT (8X, 24H NMBRF TESTED (% PASSED), 4, 6, 17, 1H, A7)
380 FORMAT (1H 15, 1H ***** KILL 5, 5X, 6, 10X, 15, 1H 6H *****
390 FORMAT (1H 15, 45X, 5H ***** 1X 2, 3H ATTEMPTED ENGAGEMENTS (%
15, 7H) *****
400 FORMAT (1H 15, 53X, 6H ***** 7H USHC (%) *****
410 FORMAT (1H 15, 8X, 3H ***** 1X, 6, 1A10, 5X)
420 FORMAT (8X, 30H PPR KILL / ATMP ENG =
1X, 6, 1A10, 5X)
430 FORMAT (8X, 30H PPR KILL / SHOT =
1X, 6, 1A10, 5X)
440 FORMAT (8X, 16H PPR (SHOT / OCSN) = 1X, A10, 8X, 20H PPR (SHOT / ATMP
ENG) =, 10, 5900
1X, A10, 8X, 20H PPR (ATMP ENG / OCSN) = 1X, A10)
450 FORMAT (8X, 30H PPR (LNS) =
1X, 6, 1A10, 5X)
460 FORMAT (1H 15, 1H 6H USHC, 10H SURVIVABILITY CASES IN WHICH
IGNORAT KILLED = 15, 1H, A7, 5X, 30H NMBRF TESTED (% NOT KILLED)
2, 1H, A7)
5P-OUTP161
5P-OUTP162
5P-OUTP163
5P-OUTP164
5P-OUTP165
5P-OUTP166
5P-OUTP167
5P-OUTP168
5P-OUTP169
5P-OUTP170
5P-OUTP171
5P-OUTP172
5P-OUTP173
5P-OUTP174
5P-OUTP175
5P-OUTP176
5P-OUTP177
5P-OUTP178
5P-OUTP179
5P-OUTP180
5P-OUTP181
5P-OUTP182
5P-OUTP183
5P-OUTP184
5P-OUTP185
5P-OUTP186
5P-OUTP187
5P-OUTP188
5P-OUTP189
5P-OUTP190
5P-OUTP191
5P-OUTP192
5P-OUTP193
5P-OUTP194
5P-OUTP195
5P-OUTP196
5P-OUTP197
5P-OUTP198
5P-OUTP199
5P-OUTP200
5P-OUTP201
5P-OUTP202
5P-OUTP203
5P-OUTP204
5P-OUTP205
5P-OUTP206
5P-OUTP207
5P-OUTP208
5P-OUTP209
5P-OUTP210
5P-OUTP211
5P-OUTP212
5P-OUTP213
5P-OUTP214
5P-OUTP215
5P-OUTP216
5P-OUTP217
5P-OUTP218
5P-OUTP219
5P-OUTP220
5P-OUTP221
5P-OUTP222
5P-OUTP223
5P-OUTP224
5P-OUTP225
5P-OUTP226
5P-OUTP227
5P-OUTP228
5P-OUTP229
5P-OUTP230
5P-OUTP231
5P-OUTP232
5P-OUTP233
5P-OUTP234
5P-OUTP235
5P-OUTP236
5P-OUTP237
5P-OUTP238
5P-OUTP239
5P-OUTP240
5P-OUTP241
5P-OUTP242
5P-OUTP243
5P-OUTP244
5P-OUTP245
5P-OUTP246
5P-OUTP247
5P-OUTP248
5P-OUTP249
5P-OUTP250
5P-OUTP251
5P-OUTP252
5P-OUTP253
5P-OUTP254
5P-OUTP255
5P-OUTP256
5P-OUTP257
5P-OUTP258
5P-OUTP259
5P-OUTP260
5P-OUTP261
5P-OUTP262
5P-OUTP263
5P-OUTP264
5P-OUTP265
5P-OUTP266
5P-OUTP267
5P-OUTP268
5P-OUTP269
5P-OUTP270
5P-OUTP271
5P-OUTP272
5P-OUTP273
5P-OUTP274
5P-OUTP275
5P-OUTP276
5P-OUTP277
5P-OUTP278
5P-OUTP279
5P-OUTP280
5P-OUTP281
5P-OUTP282
5P-OUTP283
5P-OUTP284
5P-OUTP285
5P-OUTP286
5P-OUTP287
5P-OUTP288
5P-OUTP289
5P-OUTP290
5P-OUTP291
5P-OUTP292
5P-OUTP293
5P-OUTP294
5P-OUTP295
5P-OUTP296
5P-OUTP297
5P-OUTP298
5P-OUTP299
5P-OUTP300
5P-OUTP301
5P-OUTP302
5P-OUTP303
5P-OUTP304
5P-OUTP305
5P-OUTP306
5P-OUTP307
5P-OUTP308
5P-OUTP309
5P-OUTP310
5P-OUTP311
5P-OUTP312
5P-OUTP313
5P-OUTP314
5P-OUTP315
5P-OUTP316
5P-OUTP317
5P-OUTP318
5P-OUTP319
5P-OUTP320
5P-OUTP321
5P-OUTP322
5P-OUTP323
5P-OUTP324
5P-OUTP325
5P-OUTP326
5P-OUTP327
5P-OUTP328
5P-OUTP329
5P-OUTP330
5P-OUTP331
5P-OUTP332
5P-OUTP333
5P-OUTP334
5P-OUTP335
5P-OUTP336
5P-OUTP337
5P-OUTP338
5P-OUTP339
5P-OUTP340
5P-OUTP341
5P-OUTP342
5P-OUTP343
5P-OUTP344
5P-OUTP345
5P-OUTP346
5P-OUTP347
5P-OUTP348
5P-OUTP349
5P-OUTP350
5P-OUTP351
5P-OUTP352
5P-OUTP353
5P-OUTP354
5P-OUTP355
5P-OUTP356
5P-OUTP357
5P-OUTP358
5P-OUTP359
5P-OUTP360
5P-OUTP361
5P-OUTP362
5P-OUTP363
5P-OUTP364
5P-OUTP365
5P-OUTP366
5P-OUTP367
5P-OUTP368
5P-OUTP369
5P-OUTP370
5P-OUTP371
5P-OUTP372
5P-OUTP373
5P-OUTP374
5P-OUTP375
5P-OUTP376
5P-OUTP377
5P-OUTP378
5P-OUTP379
5P-OUTP380
5P-OUTP381
5P-OUTP382
5P-OUTP383
5P-OUTP384
5P-OUTP385
5P-OUTP386
5P-OUTP387
5P-OUTP388
5P-OUTP389
5P-OUTP390
5P-OUTP391
5P-OUTP392
5P-OUTP393
5P-OUTP394
5P-OUTP395
5P-OUTP396
5P-OUTP397
5P-OUTP398
5P-OUTP399
5P-OUTP400
5P-OUTP401
5P-OUTP402
5P-OUTP403
5P-OUTP404
5P-OUTP405
5P-OUTP406
5P-OUTP407
5P-OUTP408
5P-OUTP409
5P-OUTP410
5P-OUTP411
5P-OUTP412
5P-OUTP413
5P-OUTP414
5P-OUTP415
5P-OUTP416
5P-OUTP417
5P-OUTP418
5P-OUTP419
5P-OUTP420
5P-OUTP421
5P-OUTP422
5P-OUTP423
5P-OUTP424
5P-OUTP425
5P-OUTP426
5P-OUTP427
5P-OUTP428
5P-OUTP429
5P-OUTP430
5P-OUTP431
5P-OUTP432
5P-OUTP433
5P-OUTP434
5P-OUTP435
5P-OUTP436
5P-OUTP437
5P-OUTP438
5P-OUTP439
5P-OUTP440
5P-OUTP441
5P-OUTP442
5P-OUTP443
5P-OUTP444
5P-OUTP445
5P-OUTP446
5P-OUTP447
5P-OUTP448
5P-OUTP449
5P-OUTP450
5P-OUTP451
5P-OUTP452
5P-OUTP453
5P-OUTP454
5P-OUTP455
5P-OUTP456
5P-OUTP457
5P-OUTP458
5P-OUTP459
5P-OUTP460
5P-OUTP461
5P-OUTP462
5P-OUTP463
5P-OUTP464
5P-OUTP465
5P-OUTP466
5P-OUTP467
5P-OUTP468
5P-OUTP469
5P-OUTP470
5P-OUTP471
5P-OUTP472
5P-OUTP473
5P-OUTP474
5P-OUTP475
5P-OUTP476
5P-OUTP477
5P-OUTP478
5P-OUTP479
5P-OUTP480
5P-OUTP481
5P-OUTP482
5P-OUTP483
5P-OUTP484
5P-OUTP485
5P-OUTP486
5P-OUTP487
5P-OUTP488
5P-OUTP489
5P-OUTP490
5P-OUTP491
5P-OUTP492
5P-OUTP493
5P-OUTP494
5P-OUTP495
5P-OUTP496
5P-OUTP497
5P-OUTP498
5P-OUTP499
5P-OUTP500
5P-OUTP501
5P-OUTP502
5P-OUTP503
5P-OUTP504
5P-OUTP505
5P-OUTP506
5P-OUTP507
5P-OUTP508
5P-OUTP509
5P-OUTP510
5P-OUTP511
5P-OUTP512
5P-OUTP513
5P-OUTP514
5P-OUTP515
5P-OUTP516
5P-OUTP517
5
```

[illegible]

00000000011111112222222222333333333344444444445555555555666666666677777777778888888889 COLUMNS
12345678901234567890123456789012345678901234567890123456789012345678901234567890 (READ VERTICALLY)

```

PE1=PEBL(I1,I1A,IPHG1)
PE2=PEBL(I1,I1A,IRHG2)
PE=PE1+TF*(PE2-PE1)
GO TO 140

C RPACKET DELTAT WITH DLT VALUES CORRESPONDING TO IDLT1 AND IDLT2
C
C 110 DO 120 I=1,NDT
C   IF (IDLT(I).LT.DLT(I)).NP.DELTAT(6E.DLT(I+1)) GO TO 120
C   IDLT1=I
C   IDLT2=I+1
C   GO TO 130
C 120 CONTINUE
C 130 STOP ' IN PEBL: FAILED TO BRACKET DELTA T '

C PERFORM A TWO DIMENSIONAL FOUR-POINT INTERPOLATION USING TWO RANGE
C POINTS FOR EACH OF TWO DELAY TIME VALUES.
C
C 130 FDT=DELTAT-DLT(IIDELT1)
C   FDT2=FDT/(IDLT(IIDELT2)-IDLT(IIDELT1))
C   TF2=TF
C   TF1=1.0-FT2
C   I11=INDEX(IIC,IDELT1,IMITS)
C   I12=INDEX(IIC,IDELT2,IMITS)
C   I1A=MIN(I11,I12)
C   PET11=PEBL(I11,I1A,IPHG1)
C   PET12=PEBL(I11,I1A,IRHG2)
C   PET21=PEBL(I12,I1A,IPHG1)
C   PET22=PEBL(I12,I1A,IRHG2)
C   IF (IDELT2-EG.6) PET2P1=0.0
C   IF (IDELT2-FO.6) PET2P2=0.0
C   PF=FUT1+TF1*PET11+TF2*PET12+FT1*PET21+FT2*PET22
C
C 140 RN22=URAN21(IP22)
C   KTEST(I7,KRF)=KTEST(I7,KPF)+1
C   IF (PH22-LE."E") GO TO 150
C
C   IABPT=17
C   CALL ABRPTL (IABPT)
C
C 150 RETURN
C   END

```

00000000011111112222222222333333333344444444445555555555666666666677777777778888888889 COLUMNS
12345678901234567890123456789012345678901234567890123456789012345678901234567890 (READ VERTICALLY)

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 64 OF 82)

AD-A100 285

ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY ABERDEEN PROV--ETC F/G 19/1
COPPERHEAD OPERATIONAL PERFORMANCE EVALUATION (COPE): COMPUTER --ETC(U)
MAR 81 R S SANDMEYER
AMSAA-TR-318

NL

UNCLASSIFIED

6 of 6

AD-A
100285



END
DATE
JAN 81
7-81
DTIC

LISTING C-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 72 OF 82)


```

000000000111111112222222233333333344444444555555566666667777777788888889 COLUMN NUMBERS
123456789012345678901234567890123456789012345678901234567890 (READ VERTICALLY)

C      SURROUTINE SMPICD (A,ND1,ND2,NPS,NDR,IPNDM,XSMPIC,ICLASS)
C      DIMENSION A(ND1,ND2)
C      ICLASS=0
C      RH=UPAN31(IPNDM)
C      IF (RH.LT.1.0) GO TO 100
C      WRITE (6,130) RH
C      CALL STRACE
C      STOP * IN SMPICD: PAIDOM NUMBER TOO LARGE *

C 100 CONTINUE
C      NORMI=KDR-I
C      DO 110 I=1,NDRM1
C      IF (A(I,I).LE.RH.AND.PH.I.T.A(I,I)) GO TO 120
C 120 CONTINUE
C      CALL STRACE
C      STOP * IN SMPICD: FAILED TO PACKET PAIDOM NUMBER *

C 120 XSMPIC=A(I,HR5)+(PH-A(I,I))*(A(I+1,NPS)-A(I,NPS))/(A(I+1,I)-A(I,I))
C      ICLASS=I
C      RETURN
C
C *** F O R M A T   S T A T E M E N T S ***
C 130 FORMAT (1H,59H IN SMPICD WITH PAIDOM NUMBER GREATER THAN 1.0,
C      I,  RN = ,F10.7)
C      END

000000000111111112222222233333333344444444555555566666667777777788888889 COLUMN NUMBERS
1234567890123456789012345678901234567890123456789012345678901234567890 (READ VERTICALLY)

```

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 76 OF 82)

[illegible][illegible][illegible]

LISTING D-1 FORTRAN LISTING OF COPE PROGRAM - CONT'D
(PAGE 78 OF 82)

SP-WARN002	032030
SP-WARN003	032040
CM-FLAG 2	000300
CM-FLAG 3	000310
CM-FLAG 4	000320
CM-KTEST 2	000490
CM-RANDOM2	000440
CM-RANDOM3	000440
CM-RANDOM4	000660
CM-RPLCTN2	000770
CM-RSPITM2	000790
CM-RSPITM3	000800
CM-RSPITM4	000810
CM-RSPITM5	000820
CM-TIME 2	001020
SP-WARN010	032110
SR-WARN011	032120
SP-WARN012	032130
SP-WARN013	032140
SP-WARN014	032150
SP-WARN015	032160
SP-WARN016	032170
SP-WARN017	032180
SP-WARN018	032190
SP-WARN019	032200
SP-WARN020	032210
SP-WARN021	032220
SP-WARN022	032230

[illegible]

```

FIRST CASE
$$$ THIS COMMENT WILL APPEAR ON THE $$$ CURRENT PAGE
WEATHER, JUNE, 0600
ACQHQ15, CLOSE
PLD, PRINTP, ON
END
NEXT CASE, DP
DESIGTYPE, MILE
RESPTIME, PARAM, 150, N
CONTRL, FCNO, LONG
PLOT, PRINTER, OFF
$$$ THE OVERPLOT CAPD WHICH FOLLOWS OVERPLOTS THE NORMAL CHOICE OF
$$$ PE DATA WITH THE PE DATA RECORD NAMED RECORD1A1AJS
$$$ NOTE THE USE OF THE $$$ BEFORE THE RECORD NAME
OVERPLOT, PE DATA, PFOOD1A1AJS
END
NEXT CASE, PD
MAD, 2200
LUSMONT, PD
$$$ THE NEXT CAPD SIGNALS FORMATTED DATA IS TO BE PLD INTO RANDOM
$$$ DIFFERENCE DATA BLOCK
TEMP, PDUST1

```

15.0
1000.0 1500.0 2000.0 3000.0 7000.0
.6 .5 .4 .3 .2
\$\$ THIS COMMENT WILL BE PRINTED ON CASE 3 HEADING PAGE

TACTICAL UNIT #4
DEPT SUPPLY UNIT
RUNC # 200
SHOP # 6, HYBRIDS, 200, 400
TRP 10
NUMVNTGT #6
DISTAVFL 50
END

NEXT CASE #0
 SELECTIVITY #3
 SOURCE REFERENCE #15
 CUST #20
 PREFERENCE #A
 ANGLE 1 #0
 END

[illegible]

```

JOB CARD. CHANGE THIS CARD TO CONFORM TO INSTALLATION USAGE.
ACCOUNT CARD. CHANGE THIS CARD TO CONFORM TO INSTALLATION USAGE.
COMMENT. THE NEXT CARD DECLARES TAPES TO HAVE RECORD TYPE 7 AND 80 CHARACTERS
COMMENT. PER LINE (RECORD).
FILE,TAPE,PT-Z,PL=80,RT=C.
COMMENT. THE NEXT STATEMENT OBTAINS A COPY OF THE CORE INPUT AND PUTS IT IN
COMMENT. A LOCAL FILE NAMED TAPES.
GET,TAPE,COPESAMPLEINPUT,ID=SAIDMEYEP,ST=HEA
COMMENT. THE NEXT STATEMENT CANNOT TAPES (CORE INPUT) TO BE LISTED. THIS IS
COMMENT. OFTEN A WISE PRECAUTION FOR CHECKING PURPOSES.
COPYSPT,TAPE5.
REWIND,TAPE5.
COMMENT. THE NEXT STATEMENT ATTACHES THE PREVIOUSLY COMPILED (IN ANOTHER JOB)
COMMENT. OBJECT CODE FOR THE COPE PROGRAM.
ATTACH,GO,COPESAMPLEINARY,ID=SAIDMEYEP.
COMMENT. THE NEXT STATEMENT ATTACHES THE COPE "DATA BASE" FILE (TAPE11) WHICH
COMMENT. WAS CREATED AND MODIFIED BY VARIOUS PREPROCESSOR RUNS (CAN E, PEPNS)
ATTACH,TAPE11,SAMPLETAPE11,ID=SAIDMEYEP.
COMMENT. THE NEXT TWO CARDS ATTACH THE INTERNATIONAL MATHEMATICAL AND
COMMENT. STATISTICAL LIBRARY (IMSL) AND DECLARE IT TO BE A LIBRARY TO BE
COMMENT. USED IN SATISFYING EXTERNAL REFERENCE IN THE COPE PROGRAM.
ATTACH,IMSL.
LIBRARY,IMSL.
COMMENT. THE NEXT CARD DECLARES TAPER TO HAVE RECORD TYPE 7 AND 140 CHARACTERS
COMMENT. PER LINE. THIS IS DONE BECAUSE TAPER IS TO BE SAVED ON THE "EXPORT
COMMENT. END" COMPUTER WHERE RECORD TYPE 7 IS THE DEFAULT RECORD TYPE.
FILE,TAPPART-Z,PL=140,RT=C.
COMMENT. THE NEXT STATEMENT LOADS AND EXECUTES THE COPE PROGRAM USING TAPES AS
COMMENT. INPUT.
GO(TAPES)
COMMENT. THE NEXT STATEMENT SAVES A COPY OF TAPER (COPE AUXILIARY OUTPUT)
COMMENT. ON THE "EXPORT END" COMPUTER. THIS TAPER IS USED TO PRODUCE THE
COMMENT. LISTINGS OF SECTION 7-6 OF THE COPE DOCUMENTATION.
SAVE,TAPER,COPEAUXOUTPUT,ID=SAIDMEYEP,ST=HEA.
COMMENT. END OF JOB

```

LISTING D-3 SAMPLE RUNSTREAM FOR COPE

```

FIRST CASE
$$$ THIS COMMENT WILL APPEAR ON THE $$$ COMMENT PAGE
WEATHER,JUNE,0600
ACQUICD1ST,CLOSE
PLMT,PRINTER,IN
END
NEXT CASE,DR
DESIGTYPE,MJLE
RESPTIME,PAPAR,150,H
CONTDL,ECHO,LONG
PLOT,PRINTER,OFF
$ THE OVERIDE CARD WHICH FOLLOWS OVERRIDES THE NORMAL CHOICE OF
$ PE DATA WITH THE PE DATA RECORD NAMED PLOMAJAS
$ NOTE THE USE OF THE $ BEFORE THE PLOMAJAS
OVERIDE,PE,DATA,PLOMAJAS
END
NEXT CASE,PD
W,D,2200
LOSMOT,PD
$ THE NEXT CARD SIGNALS FORMATTED DATA IS TO BE PEAF INTO PANDOM
$ OCCURENCE DATA BLOCK
TEMP,PORIST
5
15.0
1000.0 1500.0 2000.0 3000.0 7000.0
.6 .5 .4 .3 .2
$$ THIS COMMENT WILL BE PRINTED IN CASE 3 HEADING PAGE
TARGET TYPE,4
DEFISUPPP,NONE
PUNC,200
SMOKE,SHKPNDS,200,400
TRP,10
HUNVENTGT,6
DISTRVEN,50
END
NEXT CASE,PP
REFLECTIVITY,.3
SMOKE,PPR,SHOKE,.15
DUST,.20
PROBONVAPH,.P
ANGLE 1,60
PRINT

```

LISTING D-4 SAMPLE OUTPUT FROM COPE
(PAGE 1 OF 11)

C O P P E R H E A D O P E R A T I O N A L P E R F O R M A N C E E V A L U A T I O N

-PROGRAM BY RICHARD SANDHEVER
 -MODEL DEVELOPMENT AND DATA COLLECTION BY:
 DAVID BARNHART JIM IAN CHEPRICK DIANA FREDERICK
 RICHARD SANDHEVER RICHARD SCUNGID MICHAEL STARKS
 EDWARD STAUCH ARNIE YOUNG
 -BASED ON A PROPOSAL BY HERBERT ELLIEN
 -CONF AT THE REQUEST OF DAVID HARDISON (DUSA-OP)

UNITED STATES ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY
 GROUP WAREFARE DIVISION
 SUPPORT HEADQUARTERS ANALYSIS BRANCH
 SYSTEMS EVALUATION SECTION

COPE VERSION 6.2 CURRENT AS OF 6 MARCH 1980
 THIS FILE CONF ON 08/14/80 AT TIME 15.41.16.

LISTING D-4 SAMPLE OUTPUT FROM COPE
 (PAGE 2 OF 11)

```

$$$ $$$ $$$ $$$ $$$ $$$ $$$ $$$ $$$ $$$ $$$ $$$ $$$ $$$ $$$ $$$
** USER ** THIS COMMENT WILL APPEAR ON THE $$$ COMMENT PAGE
          COMMENTS FOR THIS SET OF CASES   $$$ $$$ $$$ $$$ $$$ $$$ $$$ $$$ $$$

```

LISTING D-4 SAMPLE OUTPUT FROM CODE
(PAGE 3 OF 11)

CAUSE OF ABORT

		***** OCCASIONS (10000) *****					
1	DESIGNATOR IN IMPROPER FIRE	84(0.84%)	HUMREP TESTED (% PASSED):	10000(100.00%)			
2	TARGET BEYOND VISIBILITY RANGE	415(4.15%)	HUMREP TESTED (% PASSED):	9916(95.85%)			
3	TARGET BEYOND DESIGNATOR RANGE	0(0.00%)	HUMREP TESTED (% PASSED):	9501(100.00%)			
4	TARGET NOT DETECTED	0(0.00%)	HUMREP TESTED (% PASSED):	9501(100.00%)			
5	COPE KILLED MISSION	7123(71.23%)	HUMREP TESTED (% PASSED):	9501(25.03%)			
6	HOST KILLED MISSION	807(8.07%)	HUMREP TESTED (% PASSED):	2378(62.70%)			
7	DESIGNATOR FAILED-OUT PRE-COMM	484(4.84%)	HUMREP TESTED (% PASSED):	1491(67.54%)			
***** ATTACHED ENGAGEMENTS (1007) *****							
8	COMM OUT (BOTH VOICE & DIGIT)	70(0.70%)	HUMREP TESTED (% PASSED):	1007(93.05%)			
9	FRONT IN TRANSMISSION	21(0.21%)	HUMREP TESTED (% PASSED):	937(97.76%)			
10	DESIGNATOR FAILED-OUT POST-COMM	159(1.59%)	HUMREP TESTED (% PASSED):	916(82.64%)			
11	LOC LOST - NO FIRING	55(0.55%)	HUMREP TESTED (% PASSED):	757(92.73%)			
***** SHOTS (702) *****							
POUND NUMBER :		1	2	3	4	5	6
12	LOS LOST DURING MISSION	30(0.30%)	62(0.62%)	96(0.96%)	151(1.51%)	242(2.42%)	345(3.45%)
	HUMREP TESTED (% PASSED)	702(95.73%)	702(91.17%)	702(86.32%)	702(78.49%)	702(65.53%)	702(50.85%)
13	DESIGNATOR NOT WARNED IN TIME	15(0.15%)	0(0.00%)	0(0.00%)	0(0.00%)	0(0.00%)	0(0.00%)
	HUMREP TESTED (% PASSED)	672(97.77%)	640(100.00%)	606(100.00%)	551(100.00%)	460(100.00%)	357(100.00%)
14	DESIGNATOR IN DEFECT FIRE	176(1.76%)	173(1.73%)	194(1.94%)	183(1.83%)	144(1.44%)	144(1.44%)
	HUMREP TESTED (% PASSED)	657(73.21%)	640(72.97%)	606(67.99%)	551(66.79%)	460(64.35%)	357(59.66%)
15	POUND RELIABILITY FAILURE	20(0.20%)	16(0.16%)	15(0.15%)	16(0.16%)	9(0.09%)	13(0.13%)
	HUMREP TESTED (% PASSED)	481(95.84%)	447(96.57%)	412(96.36%)	368(95.65%)	296(96.96%)	213(93.90%)
16	TARGET OBLSCURED BY MINI-TERRPH	54(0.54%)	61(0.61%)	83(0.83%)	55(0.55%)	46(0.46%)	29(0.29%)
	HUMREP TESTED (% PASSED)	461(98.20%)	441(96.47%)	397(99.00%)	352(94.38%)	287(83.97%)	200(85.50%)
17	POUND DID NOT ENGAGE TARGET	1(0.01%)	1(0.01%)	4(0.04%)	2(0.02%)	0(0.00%)	0(0.00%)
	HUMREP TESTED (% PASSED)	407(100.00%)	300(99.74%)	314(98.73%)	297(99.33%)	241(100.00%)	171(100.00%)
18	POUND ENGAGED BUT DID NOT HIT	105(1.05%)	83(0.83%)	74(0.74%)	54(0.54%)	43(0.43%)	49(0.49%)
	HUMREP TESTED (% PASSED)	406(74.14%)	309(78.66%)	310(76.13%)	295(81.59%)	241(82.16%)	171(71.35%)
19	POUND HIT BUT DID NOT KILL	156(1.56%)	150(1.50%)	104(1.04%)	128(1.28%)	96(0.96%)	64(0.64%)
	HUMREP TESTED (% PASSED)	301(68.17%)	306(68.04%)	276(55.03%)	241(44.80%)	198(51.52%)	127(47.54%)
***** KILLS *****							
		145	147	137	113	102	58 *****
POUND KILL/DCSH :							
		1.45%	1.47%	1.37%	1.13%	1.02%	.58%
POUND KILL/ATHP ENG :							
		14.40%	14.40%	13.31%	11.27%	10.13%	5.76%
POUND KILL/THAT :							
		20.60%	20.94%	18.60%	16.10%	14.53%	8.26%
POUND KILL/DCSH :							
		80.77%	84.54%	80.05%	72.73%	60.77%	47.16%
POUND KILL/ATHP ENG :							
		69.71%	69.71%	69.05%	69.05%	69.05%	69.05%

DESIGNATOR SURVIVABILITY: CASES IN WHICH DESIGNATOR KILLED - 343(11.41%) HUNTER TESTED (% NOT KILLED) - 657(78.54%)

LISTING D-4 SAMPLE OUTPUT FROM COPE

(PAGE 5 OF 11)

***** ПССАИПНС (10000) *****

***** ATTEMPTED ENGAGEMENTS (568) *****

***** SHOTS (428) *****

♦♦♦♦♦ 15

DESIGNATING "INVESTIGATING KILLED" - 10% (1,849)
NUMBER TESTED ("NOT KILLED") = 418(55.98%)

LISTING D-4 SAMPLE OUTPUT FROM COPE

(PAGE 7 OF 11)

***** DISPOSITIONS (10000) *****

[illegible]

DESIGN/TYPE SURVIVABILITY: CASES IN WHICH DESIGN/TYPE KILLED - 0 (0.00%)
NUMBER TESTED (% NOT KILLED) = 2254 (100.0%)

LISTING D-4 SAMPLE OUTPUT FROM COPE

(PAGE 9 OF 11)

CAUSE OF ABORT

		***** OCCASIONS (10000) *****					
1	DESIGNATOR IN IMPROPER FIRE	8410.84%	NUMBER TESTED (% PASSED):	10000(99.16%)			
2	TARGET BEYOND VISIBILITY RANGE	116(1.16%)	NUMBER TESTED (% PASSED):	9916(98.83%)			
3	TARGET BEYOND DESIGNATOR RANGE	010.00%	NUMBER TESTED (% PASSED):	9800(100.00%)			
4	TARGET NOT DETECTED	268(2.68%)	NUMBER TESTED (% PASSED):	9800(97.27%)			
5	SMOKE KILLED MISSION	1440(14.40%)	NUMBER TESTED (% PASSED):	9532(84.89%)			
6	WIFT KILLED MISSION	1636(16.36%)	NUMBER TESTED (% PASSED):	8092(79.78%)			
7	DESIGNATOR FAILED-OUT PRE-COMM	758(7.58%)	NUMBER TESTED (% PASSED):	6456(88.26%)			
***** ATTEMPTED ENGAGEMENTS (5608) *****							
8	COMM OUT (BOTH VOICE & DIGTL)	474(4.74%)	NUMBER TESTED (% PASSED):	5608(91.68%)			
9	FEED IN TRANSMISSION	150(1.50%)	NUMBER TESTED (% PASSED):	5224(97.13%)			
10	DESIGNATOR FAILED-OUT POST-COM	558(5.58%)	NUMBER TESTED (% PASSED):	5074(89.00%)			
11	LOS LOST - NO FIPING	851(8.51%)	NUMBER TESTED (% PASSED):	4516(81.16%)			
***** SHOTS (3665) *****							
FIND NUMBER :		1	2	3	4	5	6
12	LOS LOST DURING MISSION	583(5.83%)	1058(10.58%)	1529(15.29%)	2008(20.08%)	2388(23.88%)	2679(26.79%)
13	DESIGNATOR NOT WASHED IN TIME	3665(100.00%)	3665(100.00%)	3665(100.00%)	3665(100.00%)	3665(100.00%)	3665(100.00%)
14	DESIGNATOR IN IMPROPER FIRE	303(3.03%)	385(3.85%)	357(3.57%)	318(3.18%)	280(2.80%)	219(2.19%)
15	POUND DELIABILITY FAILURE	2484(87.80%)	2607(85.23%)	2136(83.20%)	1657(80.81%)	1277(78.07%)	986(77.79%)
16	TARGET OBSERVED BY MINI-TERPN	99(1.00%)	75(0.75%)	82(0.82%)	58(0.58%)	31(0.31%)	26(0.26%)
17	POUND DID NOT ENGAGE TARGET	1724(99.59%)	1755(99.66%)	1394(99.57%)	1063(99.53%)	812(99.38%)	604(98.68%)
18	POUND ENGAGED BUT DID NOT HIT	503(5.03%)	502(5.02%)	399(3.99%)	261(2.61%)	211(2.11%)	158(1.58%)
19	POUND HIT BUT DID NOT KILL	1717(70.70%)	1749(71.30%)	1388(71.25%)	1058(75.33%)	807(73.85%)	596(73.49%)
***** KILLS *****							
PROBABILITY/DCSH=		6.70%	6.21%	4.85%	4.13%	3.16%	2.36%
PROBABILITY/ATP FHC=		10.14%	10.90%	8.11%	7.25%	5.55%	4.14%
PROBABILITY/INT=		15.77%	16.96%	13.23%	11.27%	8.62%	6.44%
PROBABILITY/DCSH=		60.25%	57.27%	47.30%	36.69%	28.28%	21.83%
PROBABILITY/ATP FHC=		36.65%	34.37%	24.37%	18.11%	13.11%	9.98%

DESIGNATOR SURVIVABILITY: CASES IN WHICH DESIGNATOR FILLED - 541(2.41%) NUMBER TESTED (% NOT KILLED) = 5484(90.30%)

LISTING D-4 SAMPLE OUTPUT FROM COPE

(PAGE 11 OF 11)

APPENDIX E

NOTATIONAL CONVENTIONS

In chapters 4, 5, and 14 of this report, the content of input cards is sometimes displayed with underlining. This underlining serves only to distinguish that which goes on the card itself from the remainder of the text; no actual underlining is punched on any of the input cards themselves.

Items which are displayed in capital letters (with underlining) in the description of a card type are to be copied literally, letter-for-letter onto the input card of that type. Items displayed in lower case letters represent positions on the card at which substitutions are to be made either from a list of allowed alternatives listed (in capital letters) in the descriptive text or by supplying an appropriate numerical value. In the lists of character strings allowed as substitutions for option words, only the underlined (capital) characters are used.

Similarly, in chapter 7 where various messages and error prints are described, the capitalized, underlined part is printed as shown (but without the underlining) whereas the lower case part describes or substitutes for text which may vary from occasion to occasion.

Next page is blank.

APPENDIX F

GLOSSARY OF ABBREVIATIONS

BCS	Battery Computer System
CDC	Control Data Corporation
COPE	(1) Combat Operational Performance Evaluation (2) COPPERHEAD Operational Performance Evaluation
DMD	Digital Message Device
D.O., DO	(1) Designator Operator (2) DO as in FORTRAN DO-LOOP
DT II	Development Test II
FDC	Fire Direction Center
GLLD	Ground Laser Location Designator
HE	High Explosive
IMSL	International Mathematical and Statistical Library
LDWSS	Laser Designator Weapon System Simulation
LOS	Line-of-Sight
LTD	Laser Target Designator
MULE	Modular Unit Laser Equipment
OT II	Operational Test II
PAM	Probability of Acquisition and Maneuver
PE	Probability of (Seeker) Engagement
PRBLOS	Probability of Line-of-Sight
PREPMS	Preprocessor of Mass Storage File
TLE	Target Location Error
USAFAS	US Army Field Artillery School

BIBLIOGRAPHY

1. Amoroso, Michael, J., Tice F. DeYoung, Dennis D. Ladd, and Roger D. Schulz, A Comprehensive Digital Flight Simulation of the Cannon Launched Guided Projectile, Rodman Laboratory, Rock Island, IL, January 1977, R-TR-77-007, UNCLASSIFIED report.
This report documents a flight simulation model that serves much the same purpose as the LDWSS model. This work was not used in PAM or COPE, but may be of interest to the researcher in this field.
2. Chernick, Julian A., Moving Target Location Errors for Ground Targets, US Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD, September 1980, (to be published).
This report contains an updated version of the target location error modeling of moving targets used in PAM and COPE.
3. Chernick, Julian A., Preliminary Analysis of Extended Range COPPERHEAD Operational Performance (U), Ground Warfare Division Interim Note No. G-85, US Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD, January 1980, CONFIDENTIAL report.
This report analyzes the performance of extended range COPPERHEAD using one of the COPE variants (RPVCOPE).
4. Chernick, Julian A., Richard C. Scungio, Michael Starks, Utility of COPPERHEAD with Ground Laser Designation in a European Battlefield Environment (U), Technical Report No. 257, US Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD, December 1978, CONFIDENTIAL report.
This report includes the first efforts to model the effects of weather on COPPERHEAD performance. It contains an analysis of the utility of COPPERHEAD under European weather conditions using particular terrain scenarios.
5. Control Data Corporation, FORTRAN Extended Version 4 Reference Manual, Revision E, CDC Publication No. 60497800, CDC, Publications and Graphics Division, Sunnyvale, CA, July 1979.
This publication documents the version of FORTRAN IV in which the COPE program and its preprocessors are written. It is of special value to anyone attempting to modify COPE for use on any non-CDC machines.
6. Lewis, C.L., A.G. Nichols, and A.W. Lee, User's Guide for the Phase I Laser Designator/Weapon System Simulation (LDWSS) of the COPPERHEAD Guided Projectile System, Vol. I, Technical Report RG-77-25, US Army Missile Command, Redstone Arsenal, AL, July 1977, UNCLASSIFIED report.
This report documents the LDWSS model which was used to generate the probability of hit (given seeker engages target and round can maneuver to target) numbers used with COPE.

7. Starks, Michael, A Monte Carlo Model for Determining COPPERHEAD Probability of Acquisition and Maneuver (U), US Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD, September 1980 (to be published).
This report documents the PAM program and the modeling assumptions underlying it in much greater detail than Chapter 16 of the present report.
8. Cost and Operational Effectiveness Analysis (COPPERHEAD COEA) (U), ACN 18312, US Army Field Artillery School, FT Sill, OK, October 1979, SECRET report.
This report cites the sources of the data used with COPE, includes a section similar to Chapter 2 of the present report, and includes the analysis of COPE results. It also includes, of course, the entire COPPERHEAD COEA write-up.
9. Weaver, Jonathan M., and Lawrence Bowman, Multiple Target Simulation (MUTSI) - A Discrete Monte Carlo Technique That Evaluates the Availability of Multiple Enemy Ground Targets in a GLLD/COPPERHEAD Target of Opportunity Situation, Ground Warfare Division Interim Note G-61, US Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD, July 1979, UNCLASSIFIED report.
This report had some influence on the modeling of multi-vehicle targets in PAM and COPE.

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
12	Commander The Defense Technical Information Center ATTN: TCA Cameron Station Alexandria, VA 22314	1	Commander US Army Materiel Development & Readiness Command ATTN: DRCDE-D 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Materiel Development & Readiness Command ATTN: DRCCP 5001 Eisenhower Avenue Alexandria, VA 22333	1	Commander US Army Materiel Development & Readiness Command ATTN: DRCBSI-L 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Materiel Development & Readiness Command ATTN: DRCDE-F 5001 Eisenhower Avenue Alexandria, VA 22333	1	Commander US Army Materiel Development & Readiness Command ATTN: DRCBSI-D 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Materiel Development & Readiness Command ATTN: DRCPA-S 5001 Eisenhower Avenue Alexandria, VA 22333	1	Commander US Army Materiel Development & Readiness Command ATTN: DRCRE-I 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Materiel Development & Readiness Command ATTN: DRCQA 5001 Eisenhower Avenue Alexandria, VA 22333	1	Commander US Army Armament R&D Command ATTN: DRDAR-LCU-DM Dover, NJ 07801
1	Commander US Army Materiel Development & Readiness Command ATTN: DRCDE-R 5001 Eisenhower Avenue Alexandria, VA 22333	1	Commander US Army Armament R&D Command ATTN: DRDAR-LCU Dover, NJ 07801
		1	Commander US Army Armament R&D Command ATTN: DRDAR-LCW-E Dover, NJ 07801

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
1	Commander US Army Armament R&D Command ATTN: DRDAR-DP Dover, NJ 07801	1	Commander US Army Electronics R&D Command ATTN: DRDEL-SA FT Monmouth, NJ 07703
1	Commander US Army Armament R&D Command ATTN: DRDAR-SCS-M Dover, NJ 07801	1	Project Manager Army Tactical Data Systems ATTN: DRCPM-TDS FT Monmouth, NJ 07703
1	Commander US Army Armament R&D Command ATTN: DRDAR-SEA Dover, NJ 07801	1	Commander US Army Electronics R&D Command ATTN: DRDEL-AP-UA 2800 Powder Mill Road Adelphi, MD 20783
1	Commander US Army Armament R&D Command ATTN: Technical Library Dover, NJ 07801	1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL White Sands Msl Rg, NM 88002
1	Project Manager Cannon Artillery Weapon Systems ATTN: DRCPM-CAWS Dover, NJ 07801	1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-T White Sands Msl Rg, NM 88002
1	Commander Rock Island Arsenal ATTN: Technical Library Rock Island, IL 61299	1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-TFC White Sands Msl Rg, NM 88002
1	Commander US Army Armament Materiel Readiness Command ATTN: DRSAR-PE Rock Island, IL 61299	1	Commander US Army Missile Command ATTN: DRDMI-C Redstone Arsenal, AL 35809
1	Commander Harry Diamond Laboratories ATTN: DELHD-SAB 2800 Powder Mill Road Adelphi, MD 20783		

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
1	Commander US Army Missile Command ATTN: DRDMI-TGT Redstone Arsenal, AL 35809	5	Commandant US Army Field Artillery School ATTN: ATSF-CD-AD FT Sill, OK 73503
1	Commander US Army Missile Command ATTN: DRSMI-DP Redstone Arsenal, AL 35809	1	Commandant US Army Field Artillery School ATTN: ATSF-CTD FT Sill, OK 73503
2	Chief Defense Logistics Studies Information Exchange US Army Logistics Management Center ATTN: DRXMC-D FT Lee, VA 23801	1	Commandant US Army Armor School Combat & Training Development ATTN: ATZK-CD-MS FT Knox, KY 40121
1	Commander US Army Concepts Analysis Agency 8120 Woodmont Avenue Bethesda, MD 20014	1	Commandant US Army Infantry School ATTN: ATSH-CD-CS FT Benning, GA 31905
1	Reliability Analysis Center ATTN: Mr. I.L. Krulac Griffiss AFB, NY 13441	2	HQDA (DUSA-OR) WASH DC 20301
1	Commander US Army Training & Doctrine Comd ATTN: ATCD-TM FT Monroe, VA 23651	1	HQDA (DAMA) WASH DC 20301
1	Commander US Army Training & Doctrine Comd ATTN: ATCD-C FT Monroe, VA 23651		<u>Aberdeen Proving Ground</u>
1	Commander US Army Combined Army Combat Development Activity ATTN: ATCA-CA-A FT Leavenworth, KS 66027	1	Commander, USATECOM ATTN: DRSTE
		1	Commander, USATECOM ATTN: DRSTE CS-A
		1	Director, USABRL
		1	Director, USABRL ATTN: DRDAR-TSB-S (STINFO Branch)
		1	Director, USAHEL Bldg. 520

