NASA CONTRACTOR REPORT 166471

Flight Dynamics Analysis and Simulation of Heavy Lift Airships

Volume III: User's Manual

(NASA-CR-166471-Vol-3) FLIGHT DYNAMICS ANALYSIS AND SIMULATION OF HEAVY LIFT AIRSHIPS. VOLUME 3: USER'S MANUAL Final Report, Sep. 1979 - Dec. 1982 (Systems Technology, Inc.) 91 p HC A05/MF A01 N83-22209
G3/08 Unclas 09709

Robert F. Ringland
Mark B. Tischler
Henry R. Jex
Roger D. Emmen
Irving L. Ashkenas

APR 1983
RECEIVED
NASA STI FACILITY
ACCESS DEPT.

CONTRACT NAS2-10330
December 1982

NASA

NASA CONTRACTOR REPORT 166471

ORIGINAL PAGE IS
OF POOR QUALITY

Flight Dynamics Analysis and Simulation of Heavy Lift Airships

Volume III: User's Manual

Robert F. Ringland
Mark B. Tischler
Henry R. Jex
Roger D. Emmen
Irving L. Ashkenas
Systems Technology, Inc.
Hawthorne, California

Prepared for
Ames Research Center
under Contract NAS2-10330

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

ORIGINAL PAGE IS
OF POOR QUALITY

FOREWORD

This document is the third in a five volume report which describes a comprehensive digital computer simulation of the dynamics of heavy lift airships and generically similar vehicles.

The work was performed by Systems Technology, Inc., Hawthorne, California for the Aeronautical Systems Branch in the Helicopter and Powered Lift Division of the National Aeronautics and Space Administration, Ames Research Center, Moffett Field, California. The simulation development was carried on between September 1979 and January 1982 and is currently installed on the Ames Research Center CDC 7600 computer. The contract technical monitors for NASA were Dr. Mark Ardema, Mr. Alan Faye, and Mr. Peter Talbot. STI's Program Manager was Mr. Irving Ashkenas.

The authors wish to acknowledge the technical contributions of Mr. Robert Heffley, Mr. Thomas Myers, and Mr. Samuel Craig and the further contributions of Mr. Allyn Hall, Ms. Natalie Hokama, and Ms. Leslie Hokama in simulation software development. Special thanks are due to Ms. Kay Wade, Ms. Linda Huffman, Mr. Charles Reaber, and STI's production department for the preparation of the five volumes of this report.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE OF CONTENTS

Page

1. INTRODUCTION.. 1
   A. Description of This and Other Related Manuals.............. 2
   B. Notes to a First Time User.................................. 3

2. INPUT DATA FILES.. 6
   A. Data File: GMDTA... 7
   B. Data File Name: ARODTA..................................... 13
   C. Data File Name: IFCDTA..................................... 14
   D. Data File Name: PLMDTA..................................... 15
   E. Data File Name: TRMDTA..................................... 16
   F. Data File Name: HISDTA..................................... 19
   G. File Names: RG1, RG2, RG3, RG4.............................. 31
   H. Data File: OUTLST.. 33
   I. File Name: INPUT (Tape5 Interactive Responses)............. 34
   J. Data File: ERMSSG.. 36

3. PROGRAM OUTPUT.. 40
   A. Output Listing.. 40
   B. Plotting File.. 43

4. TRIM 46

5. STABILITY DERIVATIVES..................................... 48

6. ERROR PROCESSING.. 51

7. PROGRAM HLAPAY; SIMULATION OF A HEAVY LIFT AIRSHIP CARRYING A PAYLOAD ATTACHED BY CABLES.......................... 53
   A. Introduction... 53
   B. Vehicle-Payload Input Data Files........................... 54
   C. Program HLAPAY Output...................................... 63
   D. HLAPAY Trim Limitations and Difficulties................... 64
   E. HLAPAY Stability Derivatives............................... 65

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE OF CONTENTS (Continued)

Page

8. PROGRAM HLAMOR; SIMULATION OF THE MOORED UNPOWERED HEAVY LIFT AIRSHIP.. 67
   - A. Introduction... 67
   - B. Moored Vehicle Input Data Files........................... 67
   - C. Program HLAMOR Output..................................... 70
   - D. HLAMOR Trim Considerations................................ 70
   - E. HLAMOR Stability Derivatives.............................. 70

9. PLOTTING FILE POST-PROCESSOR PROGRAM PPLOT.......................... 72

10. LINEARIZATION MATRIX FROM GUST SOURCES TO VEHICLE COMPONENTS... 74

11. OVERLAYS, EXECUTION OF THE PROGRAMS ON THE NASA/AMES CDC 7600/SCOPE SYSTEM.... 77

APPENDICES

* Published separately in Vol. IV

APPENDIX A. INPUT VARIABLES...A-1

APPENDIX B. SAMPLE PROGRAM OUTPUT.....................................B-1

APPENDIX C. SAMPLE INPUT DATA FILES...................................C-1

APPENDIX D. OUTPUT VARIABLES..D-1

ORIGINAL PAGE IS
OF POOR QUALITY

LIST OF FIGURES

Page

1. Vehicle Geometry Vectors and Reference Centers........... 8

2a. Basic LPU Geometry Configuration Vectors................. 9

2b. Basic Landing Gear Configuration......................... 9

3. LPU Euler Angles, $\underline{n}_h^i$, Namelist NGBANG, Data File GMDTA, Page A-6, Volume IV......................... 10

4a. Propeller Shaft Orientation Angles (Namelist NPRPRIG), Data File GMDTA, Page A-6, Volume IV........... 11

4b. Direct Thrust Orientation Angles, Namelist NJETHSA, Data File GMDTA, Page A-11, Volume IV........... 11

5. Program Generated Flight Control Commands Based on User Input Commands.................................. 24

6. Hover Control Operation-Hovering Trim Condition.. 25

7. Hover Control Operation-Nonhovering Trim Condition.. 26

8. Typical One-Minus-Cosine Gust............................ 28

9. Gust Input Source Geometry; RFSRCX is Positive as shown. RASRCX is Negative as shown. RSORCY is Always Positive.. 29

10. Format of the Binary Plotting File (PLOT)................ 44

11. Stability Derivative Calculation Algorithm............... 50

12. Payload Position Relative to Hull - Top View............. 56

13. Illegal Cable Geometries................................. 57

14. Cable configuration 1 (Inverted "V" and Front Lower then the Rear)..................................... 59

15. Cable Configuration 2 (Four Cables)...................... 60

16. Cable Configuration 3 (Bifilar Cables)................... 61

17. Job Control Sequence to Load and Execute Program......... 80

18. Job Control Sequence to Execute Program PPLOTF........... 81

19. Segmented Load Directive File (SLDIR) for Program HLASIM... 82

20. Segmented Load Directive File (SLDIRM) for Program HLAMOR... 83

21. Segmented Load Directive File (SLDIRP) for Program HLAPAY... 84

SECTION 1

INTRODUCTION

The heavy lift airship simulation consists of three computer programs. The vehicle is modeled as a buoyant envelope (hull) with lift propulsion units (LPUs) attached. The LPUs each consist of one rotor for upward lift, one propeller for forward propulsion, and a fuselage (nacelle). The model also includes landing gears, an attachable slung-payload, and a mooring point. The characteristics of all model elements are user defined.

The three programs are:

HLASIM — Models the powered vehicle in flight.

HLAMOR — Models the moored unpowered vehicle constrained to a mast at the mooring point.

HLAPAY — This program is the largest of the three. It models the vehicle in flight carrying a slung payload.

This simulation was divided into separate programs because:

1) This allows modular development and testing of the program during development and future alterations.

2) Large parts of the main program were unnecessary for program HLAMOR because it is unpowered, and has only three degrees of freedom. The trim for HLAMOR is consequently completely different.

3) The payload was developed and tested as a separate program. HLAPAY executes the payload program in addition to program, HLASIM. Having a vehicle only program (HLASIM) frees the user from having to prepare payload input files unless the vehicle with payload is to be modeled.

All programs have the same basic algorithm and large parts of the programs use the same code. They also share the same data files whenever possible.

The basic program algorithm is:

1) Read and initialize data.
2) Calculate the trim conditions.
3) Output the results of the trim calculations.
4) Calculate and output the stability derivatives if requested.
5) Repeat 1-4 for additional trim conditions if requested.
6) Read and initialize the time history data.
7) Write the output data at time 0.0.
8) Integrate the vehicle states forward in time, implementing appropriate control commands and gust disturbances.
9) Write program output data at the user requested print intervals.
10) Repeat 8 and 9 until the end of the time history.

A. DESCRIPTION OF THIS AND CTHER RELATED MANUALS

An overview of the various manuals associated with this simulation is in order before continuing with detailed discussion of the program. In addition to this User's Manual, there is a Programmer's Manual and a Technical Manual.

This User's Manual is designed to provide the user with the basic information necessary to run the program as it has been designed. This manual does not discuss any of the internal workings of the code or the technical details of the equations and their derivations. This manual describes the various data files necessary for the program and explains the output from the program and the various options available to the user when executing the program. The discussion of the data files is limited to:

1) The type of data contained in each file.
2) The inputs necessary to create special configurations.

3) Inputs whose nature is specialized or not obvious.

4) Additional data file information is contained in:

   Appendix A - tabulates all input variables. It indicates which values are valid for the various variables and other special considerations which the variables may have.

   Appendices B and C - contain sample sets of input files and the output resulting from those input files.

The Programmer's Manual is designed for the maintenance programmer who will be supporting this program. It explains the logic of the various program modules, and in some cases gives a detailed explanation of the reasons for various implementations. The topics discussed in the User's Manual will not be repeated in the Programmer's Manual. Consequently, the maintenance programmer will have to consult both of these manuals when working with the program. The Programmer's Manual contains several appendices, including a dictionary of program variables, a list of all subroutines and their purposes, a subroutine/common block/cross reference listing, and a calling/called subroutine cross reference listing.

The Technical Manual contains a detailed discussion of all simulation models, including derivations of all the equations, and methodology for calculating the required program input data. The user will have to consult this manual for all technical information he requires in generating the input data files or in understanding the output.

B. NOTES TO A FIRST TIME USER

The heavy-lift airship simulation programs are rather complex, containing many user controlled options. Therefore, it is recommended that any configuration to be simulated should first be analyzed using program HLASIM. By using this program, the user can define geometry and mass characteristics and make test runs to validate his input files. The biggest problem confronting a new user will be the generation of data files which accurately represent the vehicle he wishes to model. Several runs should be made to check the output for reasonable results.

Maintaining data files can be a significant problem. To assist in overcoming this bookkeeping task, input routines were designed to allow the same data files to be used by the three different programs. For example, the mooring program (HLAMOR), which does not use the flight control system inputs, will still read the same data file (HISDTA) as program HLASIM. HLAMOR passes over the unwanted data. In this way, the user can analyze the flight conditions and the moored conditions of the same vehicle, confident that the configuration input data was exactly the same for both programs.

It is recommended that the user create a basic set of input files. Then, make copies of those files and alter the data appropriate to the program run wanted. These alterations can be noted in the comment lines of the interactive questions data file, and later the run can be easily identified by knowing that the basic data with the noted changes was used. This method of copying files insures that all runs start with the same basic data. If the user alters the basic data file he runs the risk of failing to return the data to its original form and thereby invalidating future runs.

The following section describes the minimum data files necessary to execute program HLASIM. This will allow a new user to run the program with minimal initial effort. Additional data files required for special options, can be added as they are needed.

The following eight data files are required to execute the program with a minimum of options.*

The first two files define the desired printed output variables and the basic program user options:

*In order to execute the program with this minimum data set, the user must enter "False" to the interactive question "Time History?"

1) OUTLST

2) Interactive responses (INPUT)

The next file which is supplied along with the program code, contains the program error messages:

3) ERMSSG

The remaining five files are used to define the basic vehicle configuration and trim conditions:

4) GMDTA

5) ARODTA

6) INFDTA

7) PLMDTA

8) TRMDTA

Using these files, the program will calculate the trim and print the results.

Additional Data:

9) HISDTA — Time history data file. This file must be supplied if the user has answered the interactive question "Time History?" as true.

Gust string files (RG1, RG2, RG3, and RG4) are not necessary as long as the user inputs GSTFLG = F in the data file HISDTA.

This manual describes each of the above files in Section 2. The user should consult these sections along with Appendix A, B, and C when generating data files. Section 11 of this manual contains the specifics of how to create and load the job to run the program on the NASA/Ames CDC 7600/Scope System.

SECTION 2

INPUT DATA FILES

This section discusses each input data file necessary to the program (HLASIM). The discussion is not exhaustive. The methods for estimating the engineering constants are found in the "Technical Manual" and the definitions of each input variable is given in Appendix A of this manual. Sample data files are given in Appendix B. The discussion here include the file format and special or not obvious uses made of any variables.

The following is a complete list of the input data files:

GMDTA ARODTA IFCDTA PLMDTA TRMDTA HISDTA	Vehicle, Trim, and Time history variables (Discussed in Section 2-A to 2-F)
MORDTA	Mooring exclusive data (discussed in Section 8-B
PAYDTA	Payload exclusive data (discussed in Section 7-B)
RG1-RG4	Gust strings
RG5, RG6	Gust strings (discussed in Section 2-G)
OUTLST	User selected output variables code numbers (Vehicle discussed in Section 2-H)
PYOUTL	User selected output variables code numbers (Payload-discussed in Section 7-B)
INPUT	Run Parameters (discussed in Section 2-I)
ERMSSG	Error messages (Supplied with program discussed in Section 2-J)

A. DATA FILE: GMDTA

Contents: Geometry and Mass Characteristics

The variables in this file are listed under the following output headings:

- "GEOMETRY INPUTS,"
- "MOORING POINT GEOMETRY,"
- "LANDING GEAR ATTACH POINTS AND SPRING CONSTANTS,"
- "MASS AND MOMENT OF INERTIA INPUTS"

The definitions and descriptions for these input variables are found in Appendix A. The moment of inertia characteristics are discussed in the technical manual accompanying this program. Consequently, only those inputs which have special significance or are used in a special manner are discussed in this section. All inputs are in Namelist format.

Figures 1 and 2 illustrate the geometry configuration vectors and their reference centers. The legend for Figs. 1 and 2 is as follows:

$\vec{r}_1$	- RTALOC	GMDTA	$\vec{r}_7$	- RACELC	HISDTA
$\vec{r}_2$	- RATCH2	GMDTA	$\vec{r}_8$	- RVSNLC	HISDTA
$\vec{r}_3$	- RLTCH2	GMDTA	$\vec{r}_9$	- RROTR2	GMDTA
$\vec{r}_4$	- RMORPT	GMDTA	$\vec{r}_{10}$	- RPROP2	GMDTA
$\vec{r}_5$	- RATHG1	GMDTA	$\vec{r}_{11}$	- CGLP2	GMDTA
$\vec{r}_6$	- RHULCG	GMDTA	$\vec{r}_{12}$	- RACLP2	ARODTA

P_1 - Reference Center Hull (center of volume)
P_2 - Reference Center Tail
P_3 - Reference Center LPU
P_4 - Attach point of LPU
P_5 - Attach point of Landing Gear
P_6 - Center of gravity hull
P_7 - Accelerometer location
P_8 - Velocity sensor location
P_9 - Rotor hub

ORIGINAL PAGE IS
OF POOR QUALITY

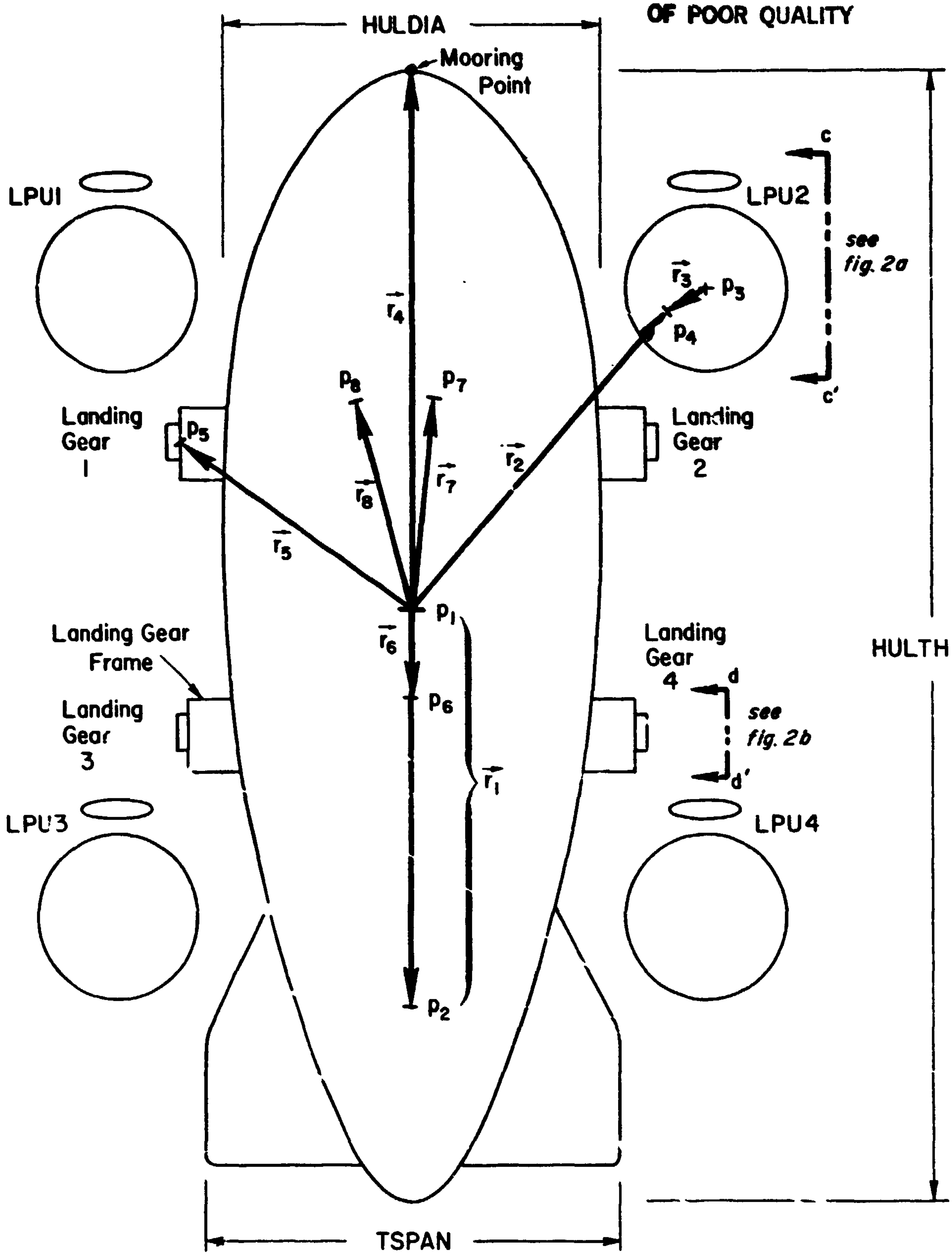
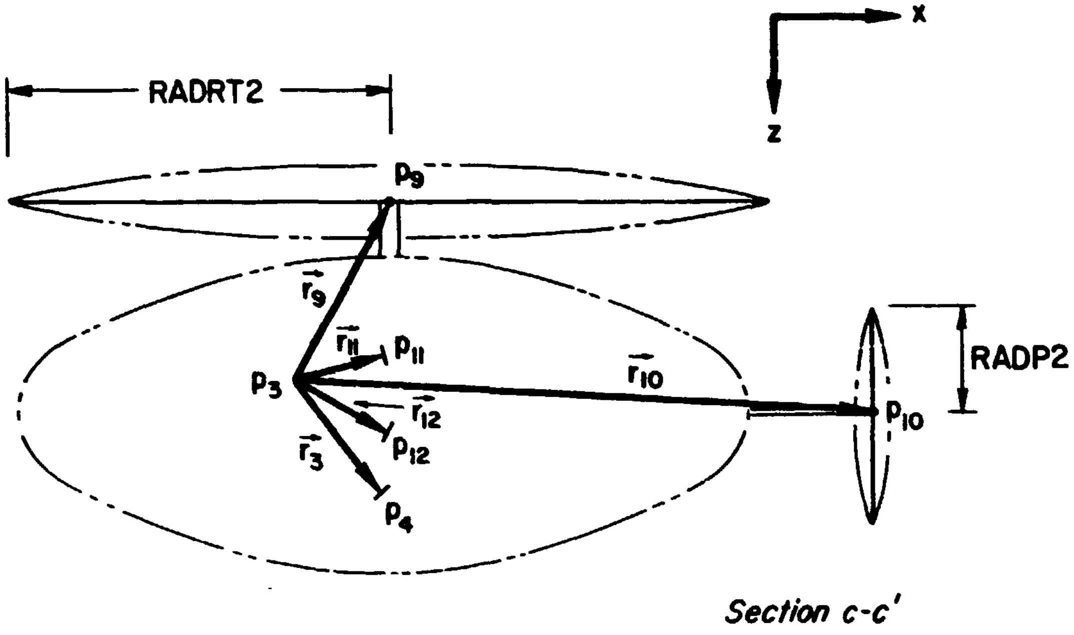


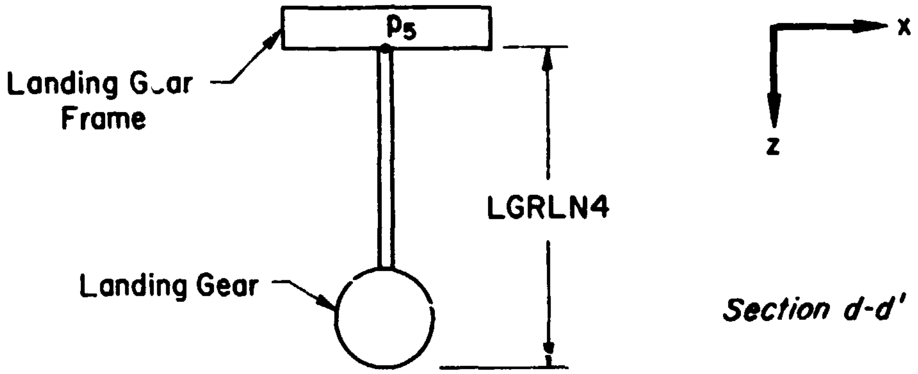
Figure 1. Vehicle Geometry Vectors and Reference Centers

ORIGINAL PAGE IS
OF POOR QUALITY



LPU2 Right Side View

Figure 2a. Easic LPU Geometry Configuration Vectors



Right Side View

Figure 2b. Basic Landing Gear Configuration

ORIGINAL PAGE IS
OF POOR QUALITY

P_{10} - Propeller hub

P_{11} - LPU center of gravity

P_{12} - LPU aerodynamic center

Figure 3 illustrates the angular orientation of the LPU reference axes with respect to the hull. Figure. 4 shows the propeller and direct thrust vector orientation with respect to the LPU.

This program was designed to model a heavy-lift airship with four lift propulsion units. By choosing special geometry and mass characteristics, the simulation will model fewer than four LPUs. [It is not

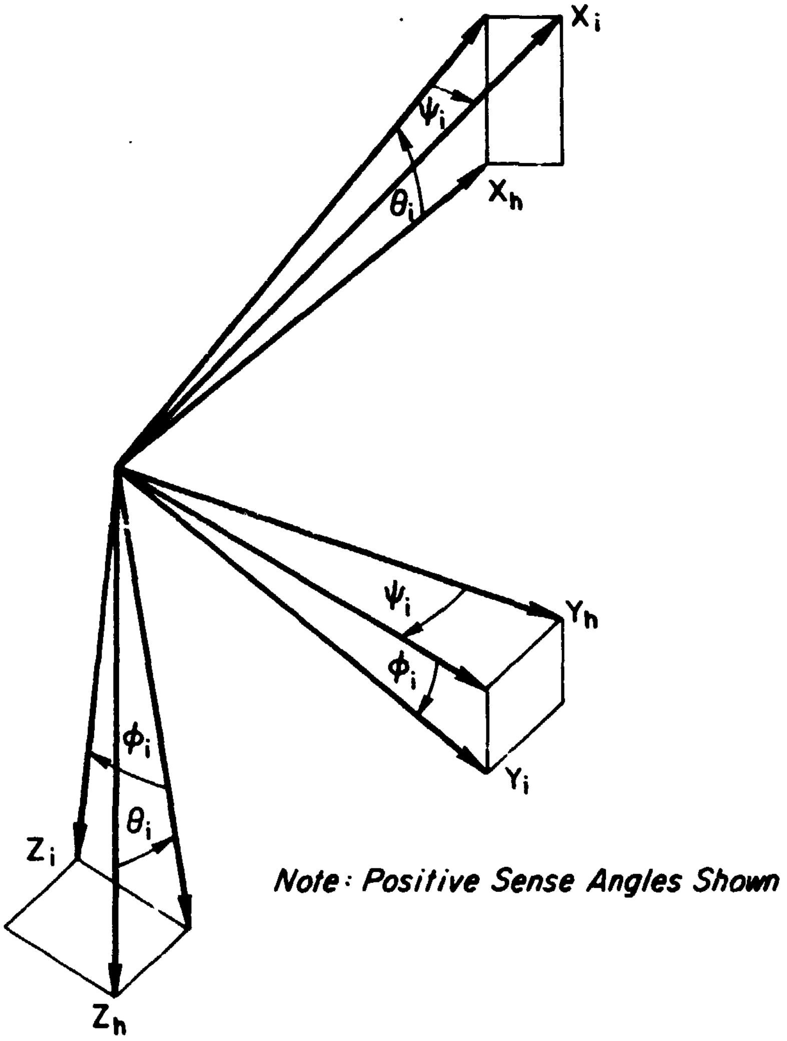


Figure 3. LPU Euler Angles, η_h^i
Namelist NGBANG, Data File GMDTA, Page A-6, Volume IV

ORIGINAL PAGE IS
OF POOR QUALITY

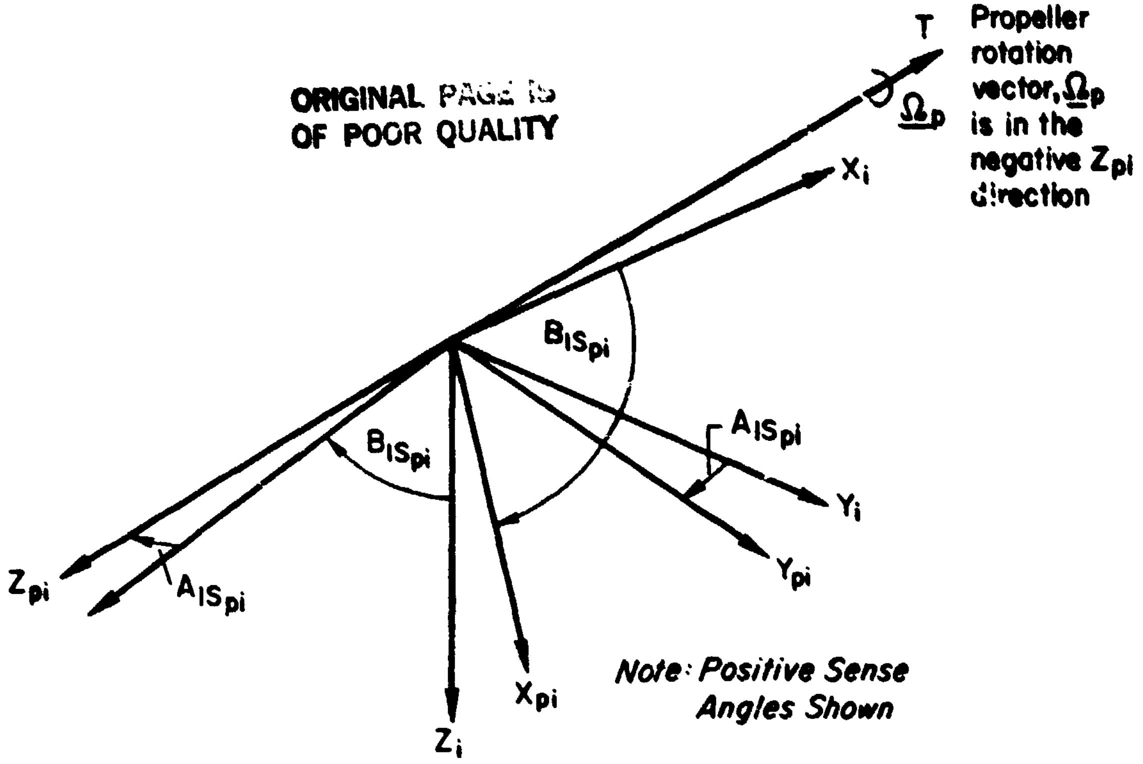

T
Propeller rotation vector, $\underline{\Omega}_p$ is in the negative Z_{pi} direction
$\underline{\Omega}_p$
X_i
$B_{IS_{pi}}$
$A_{IS_{pi}}$
Y_i
$B_{IS_{pi}}$
Y_{pi}
Z_{pi}
$A_{IS_{pi}}$
X_{pi}
Z_i
Note: Positive Sense Angles Shown

Figure 4a. Propeller Shaft Orientation Angles (Namelist NPRPRIG), Data File GMDTA, Page A-6, Volume IV

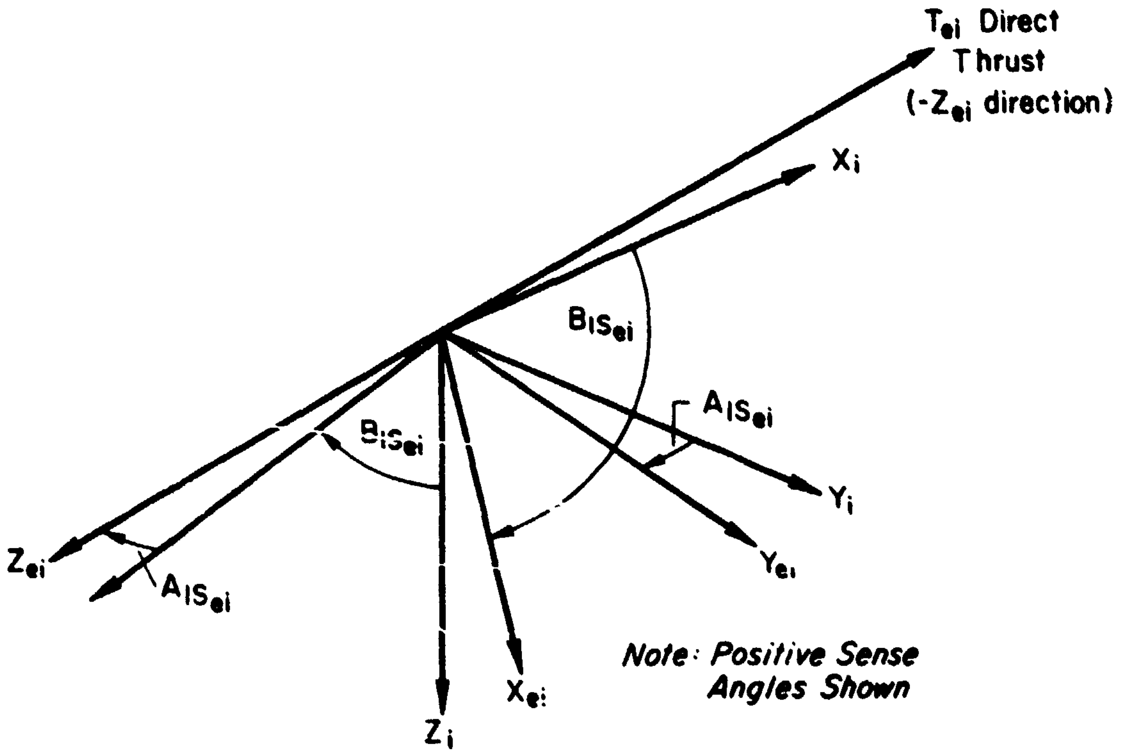


Figure 4b. Direct Third Orientation Angles (Namelist NJETHSA), Data File GMDTA, Page A-11, Volume IV

possible to model more than four LPUs.] This is done by zeroing the aerodynamic effects of one or more of the lift propulsion units and placing the LPUs at the vehicle center of gravity as follows:

a) Set the number of LPUs to the correct number [variable NUMLPU].

b) Position the LPU(s) to be zeroed so that their centers of gravity lie on the center of gravity of the hull. e.g., to eliminate LPU-1,

$$-\text{RHULCG} + \text{RATCH1} - \text{RLTCH1} + \text{RCGLP1} = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

will place the center of gravity of LPU-1 on the center of gravity of the hull.

c) The mass of the LPUs can never be set to zero. Consequently, the user should calculate the mass of the hull so that the masses of the LPU(s) being zeroed plus the mass of the hull equals the desired mass of the hull. The moments of inertia must be adjusted so they add in a similar manner.

d) There are several aerodynamic inputs, contained in data file ARODTA, which must also be set to zero.

The mooring point geometry inputs are not used in the unmoored programs (HLASIM and HLAPAY) but they must be in the input list. The landing gear spring constants and attach points are only used by HLASIM or HLAPAY if the vehicle touches the ground during the time history (the vehicle can be flown down to ground contact during a time history). They are, of course, a primary factor in HLA R.

The program was developed with four landing gears. As with the lift propulsion units, one or more of these landing gears can be removed, giving the effect of fewer than four [no greater than four landing gears can be modeled]. To remove a landing gear, no special geometry configuration is necessary; only the gear spring constant [GEARK] should be set to zero for each landing gear that is being zeroed. (i.e., GEARK1 = 0.0 will eliminate all forces associated with gear 1).

B. DATA FILE NAME: ARODTA

Contents: The Aerodynamic Constants for the Hull and Lift Propulsion Units

These input variables can be found in th~ output listing under the sections headed "LPU AERODYNAMIC PARAMETERS INPUT" and "HULL AERODYNAMIC PARAMETERS INPUT."

The user will have to consult the Technical Manual to find the method for estimating these aerodynamic parameters. Appendix A contains a list of all of these parameters and their definitions as well as a default value which will effectively cancel this parameter from the calculations of the program. This default value can be used if the user: 1) does not think that this particular effect is important to be modeled; or 2) is unsure of the method for generating a correct value. All inputs are in namelist format.

The discussion of data file GMDTA contains a diagram of the geometry configuration vectors. This diagram includes the vectors, RACLP1-4, which are in file ARODTA.

When the user is simulating a configuration that has less than four lift propulsion units there are several variables in this file which must be zeroed. They are: the lift curve slope, the drag equation coefficients, and the LPU fuselage aerodynamic X-force, Y-force, and Z-force derivatives. [The user should see the discussion of the geometry input file, GMDTA, for a discussion of the geometry configuration necessary to go with these aerodynamic parameters when zeroing some of the LPUs.] For example, the aerodynamic effects of LPU-1 will be zeroed by setting the following values to zero:

DLTP1A = 0	DLTR1A = 0
DLTP1B = 0	DLTR1B = 0
DLTP1C = 0	DLTR1C = 0
LCSP1 = 0	LCSR1 = 0
XUUAF1 = 0	
YVVAF1 = 0	
ZWWAF1 = 0	

C. DATA FILE NAME: IFCDTA

Contents: Aerodynamic Interference Constants

These input variables can be found on the output listing under the titles:

INTERFERENCE CONSTANTS ON ROTOR

INTERFERENCE CONSTANTS ON PROPELLER

INTERFERENCE CONSTANTS ON FUSELAGE

INTERFERENCE CONSTANTS ON HULL

INTERFERENCE CONSTANTS ON TAIL

The user will have to consult the technical manual to see how these constants are estimated. Appendix A lists these variables with notes on valid input and input default values. All inputs are in namelist format.

D. DATA FILE NAME: PLMDTA

Contents: The Rotor and Propeller Spin Rates and the Mechanical Control Limits on the Rotors, Propellers, and Tail

These variables are on the output listing under "ROTOR AND PROPELLER SPIN RATES" and "MECHANICAL FLIGHT CONTROL SYSTEM CONSTANTS."

The mechanical flight control system limits represent physical limits for the movement of the various control element effectors. Data file HISDTA contains control system limits which correspond to these. These limits restrict the effector commands generated by the flight control system loops. The flight control system loop limits should, in general, be less than physical (mechanical) limits, (an individual flight control system loop should never generate a command which uses all of the control power possible in any particular effector).

Appendix A contains a description of these variables. They are all in Namelist format.

E. DATA FILE NAME: TRMDTA

Contents: Trim States, Atmospheric Parameters, and Stability
Derivative Flags

These inputs are found in the outut listing under

- "INERTIAL VEHICLE STATE INPUTS,"
- "ATMOSPHERIC PARAMETER INPUTS,"
- "STABILITY DERIVATIVE FLAGS."

All inputs are in namelist format. Appendix A contains a description of each of these variables.

This data file contains the basic data defining a trim condition. The number of trim runs is defined by the response to the question "How many trim flight conditions?" (see Section 2-I). The program is designed so that it will read one block of trim data, do the associated trim calculation and print the results. It will then read a second block of trim data, and a third, and a fourth, up to the number of trim conditions requested. The data set in this file must be repeated enough times to prevent the program from trying to read beyond the end of the data file. If this data file contains more blocks of data than the number of user requested trim conditions, the program will ignore the extra blocks. The last condition calculated is used as the basic starting condition for the time history run which is to follow.

Note:

> For HLASIM and HLAPAY, the hull position (HULPOS) must be such that the landing gears, gear frames, hull belly, and tail are not touching the ground.

This data file is used by all three programs (HLASIM, HLAPAY, and HLAMOR). To insure that the file is correctly positioned, there are two flags which the input routines look for:

1) SUBROUTINE INSTAT must immediately precede $NINSTAT each time this namelist appears.

2) SUBROUTINE INATMOS must immediately precede $NINATMOS each time this namelist appears.

Following each trim, the user has the option of having open-loop stability derivatives calculated, and may choose which stability derivative matrices are to be printed. These calculations are unaffected by the closed-loop control system. The reasons for providing these options are twofold:

a) The cost of calculating stability derivatives can be quite high because the basic program state derivative vector is calculated three times for each stability derivative matrix column.

b) The listing of stability derivatives is rather long. If they are not wanted the user will probably not want to have them printed.

The meanings of the stability derivative flags are:

DERVFL = T Stability derivatives will be calculated.

= F No stability derivatives are calculated. [The remaining flags have no meaning to the program.]

AMATFL = T The A matrix will be calculated. The A matrix is generated by perturbation of the state vector.

BMATFL = T The B matrix will be calculated. The B matrix is generated by perturbation of all the individual control element effectors.

BPMTFL = T The B′ matrix will be calculated. The B′ matrix is generated by perturbation of the six linked controls.

CMATFL = T The C matrix will be calculated. The C matrix is generated by perturbation of all the gust states.

CFMTFL = T Will cause the auxiliary matrix associated with each of the above matrices to be printed.

Associated with each main matrix there is an auxiliary matrix, which is produced by the constraint forces and moments. It is calculated simultaneously with the main matrix. If the user inputs CFMTFL = T, then for each of the main matrices, which he requested, the auxiliary matrix will also be printed. In the following example the user will have the A matrix, the C matrix, the A auxiliary matrix, and the C auxiliary matrix printed:

```
DERVFL = T

BMATFL = F

BPMTFL = F

CMATFL = T

CFMTFL = T
```

If CFMTFL had been false only the A and C matrices would have been printed. Whether the user sets CFMTFL equal to true or false has no effect on the cost of running the program, since auxiliary matrices are a byproduct of the main matrix calculation.

F. DATA FILE NAME: HISDTA

Contents: All Inputs Related to the Time History Calculations

These inputs can be found in the output listing under the headings:

- "FLIGHT CONTROL SYSTEM CONFIGURATION"
- "TIME HISTORY PROFILE"
- "GUST INPUTS"
- "COMPUTER ALGORITHM TIME STEP INPUTS"

This file is read only if the user has requested a time history calculation in response to the interactive question "Time History?" If the user does not want a time history calculation, it is not necessary to load this file with the program. All inputs are in namelist format. Appendix A contains a description of each of these variables.

Generally, no single loop of the flight control system should command the maximum on any individual control effector, and no individual integrator should command the maximum of any particular control loop. The integration limits (UILM, VILM, HDTILM, PHIILM, THEILM, and RILM) should be less than the corresponding loop limits (ULLM, VLLM, HDTLLM, PHILLM, THELLM, and RLLM). Both these sets of limits should be smaller than the mechanical flight control limits input in data file PLMDTA. These mechanical limits represent maximum physical limits.

The control system feedbacks can be based on two separate reference axes. The feedback control flags (UFDBK, VFDBK, and RFDBK) when set to true cause the feedbacks in the u, v, and r loops to be based on body axis reference. When set to false, the u and v feedbacks are based on the body axis apparent velocity [relative wind], and r is the Euler yaw rate, $\dot{\psi}$.

Any of the six control loops may be open (inactive) or closed (active). The closed-loop flags are ULPFLG, VLPFLG, HDTFLG, PLPFLG, QLPFLG, and TRTFLG. [T indicates the system is closed loop; F indicates the system is open loop.] The user must refer to the technical manual for a description of the flight control system gains and how they are calculated.

The flight control system has a hover control command. This command consists of the starting and ending times [POSHT1, POSHT2] for the vehicle to hold a hover position. The user does not define the position, but only the times. The hover position is flagged as the inertial position of the vehicle at the time POSHT1. The hover control functions as follows:

1) For POSHT1 < TIME < POSHT2 the hover control module generates commands to hold and/or return the vehicle to the position flagged (PHRF:X, PHRF:Y, PHRF:Z, and PHRF:PSI on the output listing). These commands will react to any vehicle motion away from this reference position. All user input flight control commands are ignored during this time span.

2) "TEST" commands will be recognized while the hover control is turned on. As soon as the vehicle moves away from the hover position as the result of the "TEST" command the hover control will react to counter that motion. It will eventually overcome the effect of the "TEST" command.

3) When the hover control is turned off (POSHT2), the system resumes the input command interpolations. If the user has input a command at the same time the hover control is turned off, the system immediately jumps to that command, creating a step command. If there are no commands input at the same time the hover control is turned off, the system will interpolate between the last hover control command (internally generated) and the next flight control command (user input).

The user should refer to the flight control system commands below for further discussion and diagrams on how these two command systems interact.

The TIME HISTORY PROFILE contains the test commands and flight control system commands which will be executed during the time history run. The Test Commands control the individual element effectors; they move the effector a specified angle for a given time. [Note: T1 and T2 in the variable name indicate starting and ending time, respectively.] Each group of effectors (i.e., rotors, propellers, and tail) have a single user selected starting and ending time associated with its command inputs. The linked controls likewise each have one starting and ending time. Each of the six linked controls command one of the six degrees of freedom of the vehicle. These controls will manipulate all the various effectors according to the mixing scheme which has been established internal to the program. If the user wishes to change this mixing scheme, the program code must be altered (see Programmer's Manual.) This should be done with great care because it may very adversely affect the trim and closed loop response.

The test commands are added to the flight control system commands. If the vehicle is being flown closed loop, these test commands serve as control system excitations.

The FLIGHT CONTROL SYSTEM COMMANDS are input in a format different than the other inputs in this data file. This is done to allow a variable number of commands to be input without using counters or flags. The user may input from zero to twenty commands for each of the six loops [linked controls]. Each command is a time-command pair (real numbers). It is recommended that the commands be put on separate lines, although it is not necessary. The time is first and the command is second. Each number must be followed by a comma.

The six command variable names, in order, are: UCMD, VCMD, HDTCMD, PHICMD, THECMD, and TRTCMD. The namelist, NCOMAND, is followed by groups of commands each preceded by its variable name. If there are no commands for a particular variable, then that variable should not appear. If the user is not inputing any commands for the time history, the namelist name, followed immediately by the end flag must still be in the file.

```
i.e., 1$NCOMMAND,
      0$END
```

A typical set of commands could be:

```
1$NCOMMAND
   UCMD = 0.0, 2.0,
          3.0, 8.0,
 PHICMD = 3.0, 0.02,
 TRTCMD = 0.0, 0.0,
          2.0, 0.05,
0$END
```

Note:

1) All of the input values are followed by commas.

2) Variable names of controls not being commanded are excluded [i.e., VCMD, HDTCMD, and THECMD].

3) There can be up to twenty time-command pairs for each control.

4) The times [first number of each pair] must be in ascending order for each control. No two times for the same control can be equal.

5) The command pairs are written on separate rows. This is not necessary, but is highly recommended.

If the user does not input a flight control system command at time 0.0 for any of the six commands, the program generates a time 0.0 command equal to the trim condition. After the time history has begun, the program repeatedly interpolates between the input commands (or between trim and the first user command) to get an appropriate command for any particular time. In the example above, the user has input a command of 2.0 at time 0.0 for UCMD. The program will calculate a command of 4.0 at time 1.0 [interpolation between time 0.0 and time 3.0]. If the user has input a command at time 0.0 different than the trim condition, the user input command is implemented at time 0.0. This results in a step command at time 0.0.

If the user commands do not continue until the time history is complete, the program will hold the last user command until the end of the simulation. This is done to be consistent with the gust inputs (see below). If the user commands continue beyond the final simulation time, the program will interpolate toward those commands.

ORIGINAL PAGE IS
OF POOR QUALITY

As noted earlier, the user can create a step command at time 0.0 by inputting a command which is different than the trim value. A step command can also be created during a time history run by inputting two commands with times which are very close together (the times must never be equal). For example:

```
UCMD  =  3.0, 25.0,
      =  3.001, 35.0,
```

This commands 25 at time 3.0. The next algorithm step the command will be 35 which is, in effect, a step command.

The following example will demonstrate these options:

```
     VHUL     =  18.0, 0, -1.0,    (from file TRMDTA)
     HULEUL   =  0.0, 0.1, 0.0

 1$NCOMAND     (from file HISDTA)
     UCMD     =  3.0, 20.0,
                 6.0, 30.0,
                 8.0, 50.0,
     THECMD   =  0.0, 0.0,
                 5.0, 0.0,
                 7.0, 0.15,
                 12.0, 0.5,
 $END
 TSIM  =  10.0,  (simulation time; file HISDTA)
```

Note: -1.0 in the trim velocity, VHUL = 18.0, 0.0, -1.0, refers to a flight condition of 18 units forward (positive) and 1 unit vertical (upward) velocity. This negative body axis velocity corresponds to a positive climb rate command (HDTCMD = +1.0).

Figure 5 illustrates the example above. Figures 6 and 7 illustrate the operation of the hover control system.

The GUST INPUTS are next in this data file. There are two types of gust inputs, a (1-minus-cosine) gust generated at individual vehicle elements, and four gust strings which are read from four user created files.

The beginning of these gust inputs must be flagged with SUBROUTINE INGUST immediately preceding namelist NHGCOM. This is done to allow HLAMOR to skip over the unused flight control system and time history profile inputs.

ORIGINAL PAGE IS
OF POOR QUALITY

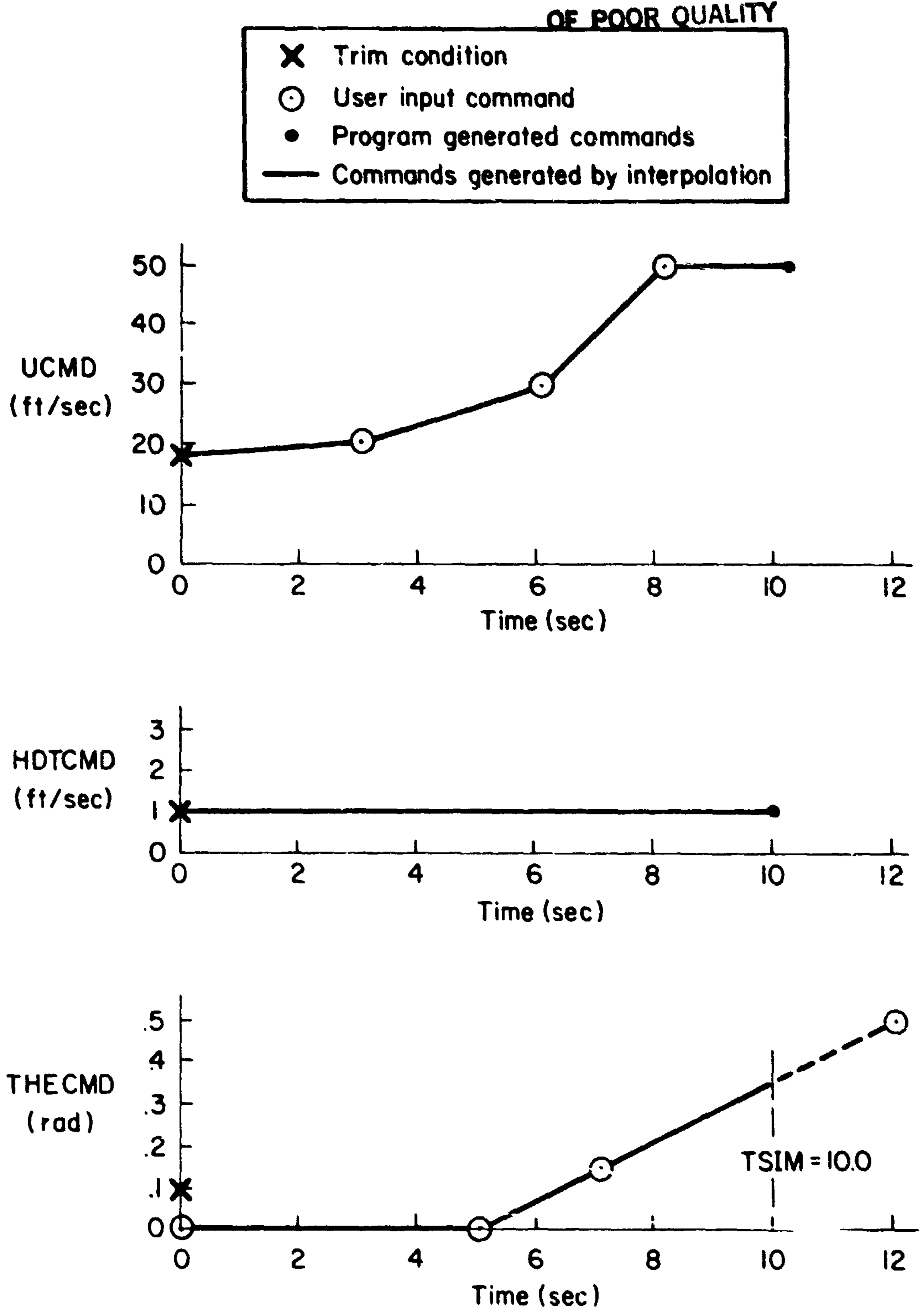


Figure 5. Program Generated Flight Control Commands Based on User Input Commands

ORIGINAL PAGE IS
OF POOR QUALITY

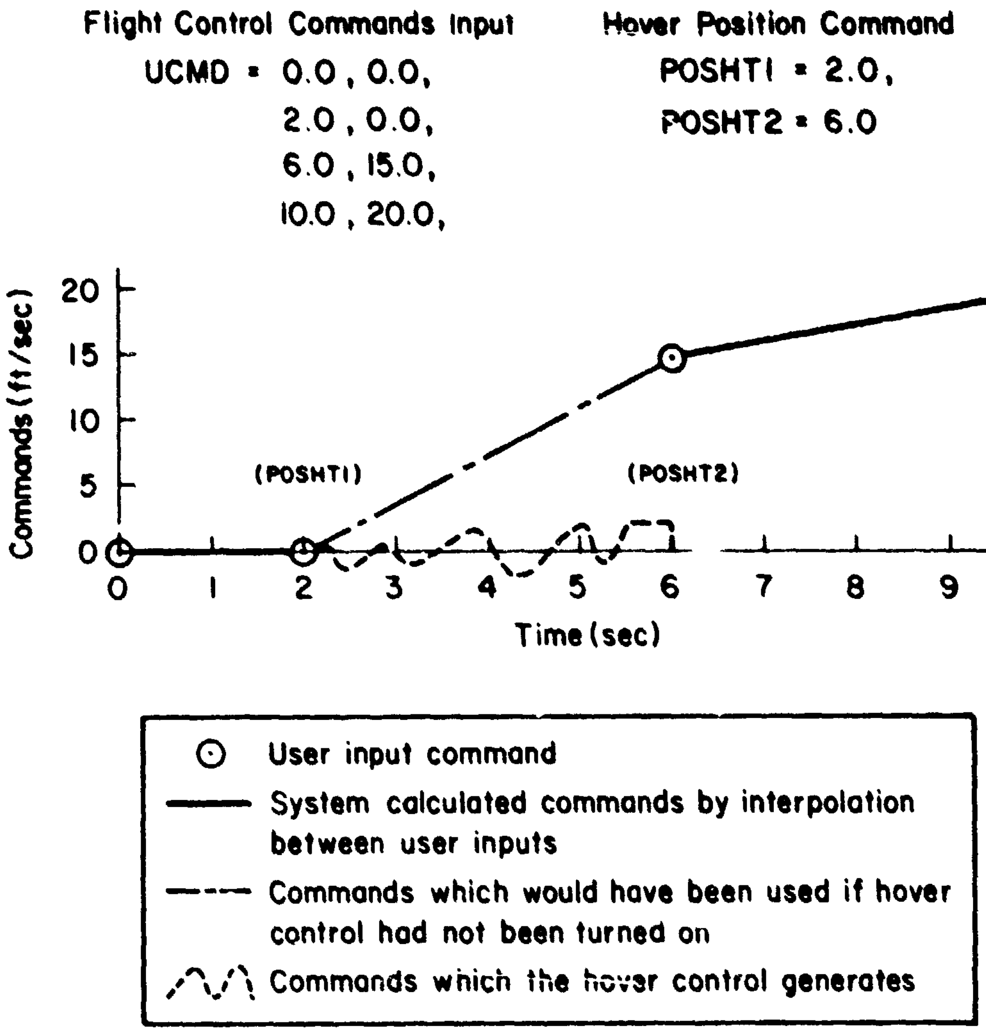


Figure 6. Hover Control Operation-Hovering
Trim Condition

ORIGINAL PAGE IS
OF POOR QUALITY

Flight Control Commands Input	Hover Position Command
UCMD = 0.0, 10.0,	POSHT1 = 3.0
2.0, 10.0,	POSHT2 = 5.0
5.0, 20.0,	
10.0, 15.0,	

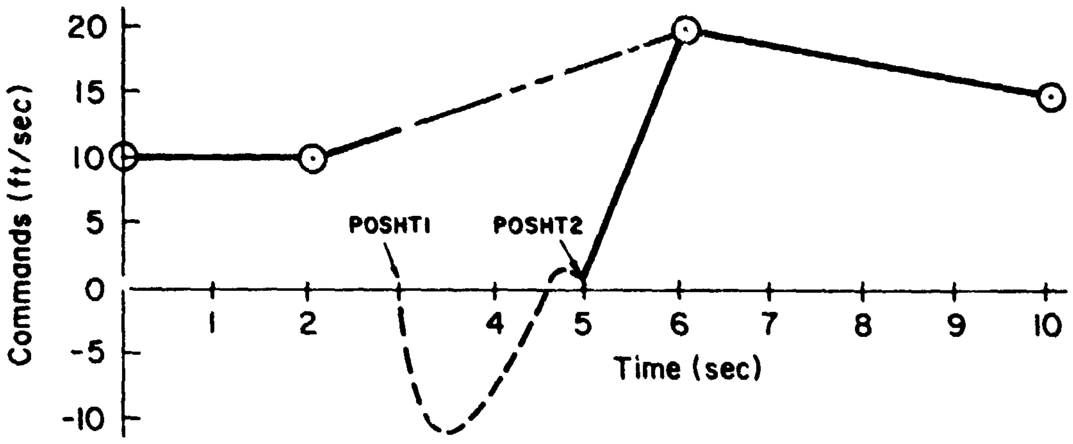


Figure 7. Hover Control Operation-
Nonhovering Trim Condition

The (1-minus-cosine) gust on each element is turned on and off by the starting (T1) and ending (T2) time, similar to the format of the test commands and the hover control command. These gusts trace out a (1-minus-cosine) curve over the time span of the gust. This is done by inputting the gust time span (variables with T1 and T2 in them) and the maximum gust values (variables, with MAX in them). If a (one-minus-cosine) input is not wanted on any or all of the vehicle components, the starting times should be set to values larger than the simulation time, or the maximum values of the gust velocities should be zero. The ending time for the gust must always be larger than the starting time. There are 6 points on the vehicle where these gusts can be generated: Hull, Tail, and each of the four LPUs (see Figure 8).

The starting and ending times for the various gusts can be different. This allows the user to stagger the starting and ending times in such a manner as to approximate a gust field passing by the vehicle. This is only a crude approximation and is not really the intention of the model. The intention is to allow the user to input a gust which acts at only one point on the vehicle and therefore isolate the effects of that gust.

The gust string input option is turned on by the flag GSTFLG = T. The user must load the gust string input files (RG1-RG6) with the program only if this flag is turned on. Discussion of files (RG1-RG4) is found in Section 2-G. RG5 and RG6 are for payload input files gusts and are discussed in Section 7, Subsection B, Article 4. The three variables, RFSRCX, RASRCX and RSORCX, determine the position of the gust input sources around the vehicle (in vehicle reference frame). The two x-variables, RFSRCX and RASRCX, define the forward and aft vectors to the input source. The single variable RSORCY (always positive) is the distance right and left to the input source locations from the hull center line. This insures that the four gust input sources are located in a laterally symmetric rectangular configuration about the vehicle. It is recommended that the rectangle defined by those sources should surround the LPUs and tail, but not be unnecessarily large. Figure 9 illustrates a typical positioning of the sources about the vehicle. The

ORIGINAL PAGE IS
OF POOR QUALITY

```
$NHGCOM
HT1GST  =  3.0 seconds               QHGMAX  =  0.0 rad/sec
HT2GST  =  7.0 seconds               RHGMAX  =  0.0 rad/sec
UHGMAX  =  0.0 ft/sec                DUXHMX  =  0.0 1/sec
VHGMAX  =  8.0 ft/sec                DUYHMX  =  0.0 1/sec
WHGMAX  =  3.0 ft/sec                DVYHMX  =  10.0 1/sec
PHGMAX  =  0.0 rad/sec               $END
```

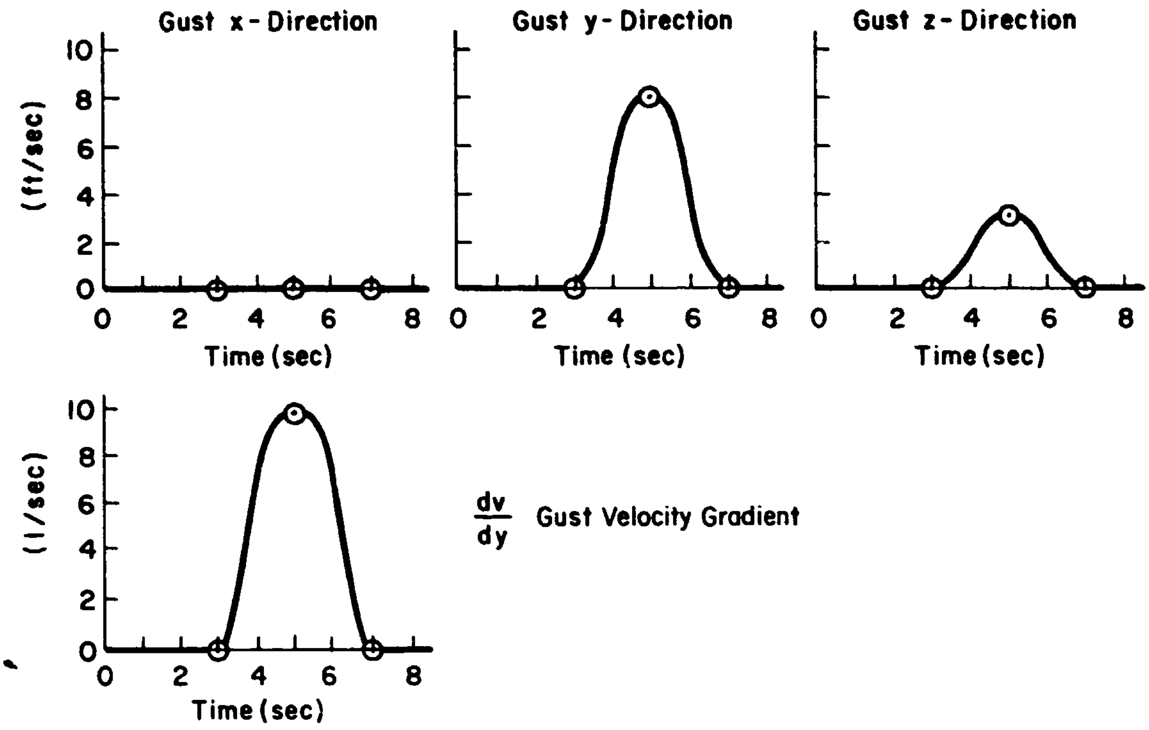


Figure 8. Typical One-Minus-Cosine Gust

ORIGINAL PAGE IS
OF POOR QUALITY

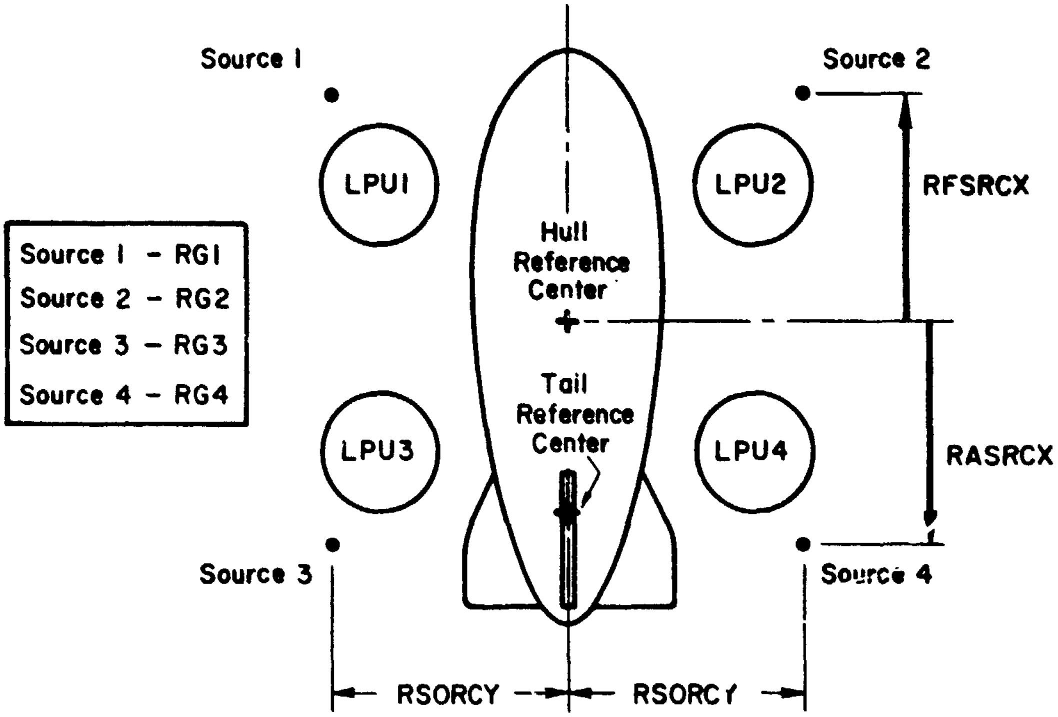


Figure 9. Gust Input Source Geometry; RFSRCX is Positive as shown. RASRCX is Negative as shown. RSORCY is Always Positive

interpolation scheme which uses these sources does not consider the different heights of the various vehicle components.

The data under "COMPUTER ALGORITHM TIME STEP INPUTS" are the last inputs from this file. They are contained in Namelist NINSTEP, which must be preceded by the flag SUBROUTINE INSTEP.

TIMSTP is the time step that is passed to the system integrator. The integrator divides this value into smaller and smaller divisions until it can meet the internal error criteria. This variable provides the initial stepsize to the integrator. It is also the maximum step the integrator can take. TIMSTP should be reasonable for the vehicle mode frequencies; unnecessarily small values will increase program execution time and cost. During development we found that values between 0.1 and 0.5 worked well.

The variable, MINSTP is the minimum allowable time step which the system integrator can take. If the system integrator is unable to meet the error criteria without going below this time step, the values of the last attempt are accepted and the program calculations continue after a warning message has been printed.

Factors affecting the choice of TIMSTP and MINSTP:

1) Rapidly changing flight conditions require smaller minimum integrator time steps (small MINSTP), i.e., gusts, and flight control commands.

2) Springs in payload cables and landing gears require smaller minimum steps (MINSTP) because of high frequency modes. (HLAPAY needs smallest steps, HLASIM is only affected if active landing gears touch the ground).

3) If the early part of the time history has changing flight conditions but the remainder is in steady flight MINSTP should be small, but TIMSTP considerably larger to allow the integrator to open up its calculation step.

4) The plotting files (see Section 3-B) are written every TIMSTP interval of the time history.

G. FILE NAMES: RG1, RG2, RG3, RG4

ORIGINAL PAGE IS
OF POOR QUALITY

Contents: The gust input strings

The gust input strings are read from these four data files. These files contain four numbers on each row. The first number is time, followed by the u, v, and w (linear) velocities of the gust. These numbers are input in free format with an unlimited number of lines. The only restriction is that the times (first entry on each line) be in ascending order. Note that first gust time entry need not be 0.0.

Gust velocities at the source location are obtained from time interpolation between the times in these files. The gust velocities and gradients for each element are determined from spatial interpolation among the pertinent source velocities. Backward difference schemes are used to determine gust accelerations.

These four data files are associated with the four gust input sources (see data file HISDTA). A typical set of gust input data and the associated explanation is presented in the following example.

Sample data files:

RG1

```
0.0,  5.0, 0.0,  0.0
1.0, 10,0, 0.0,  0.0
2.0,  8.0, 5.0, -2.0
3.0,  2.0, 1.0,  0.0
```

RG2

```
0.0,  0.0,  2.0, 0.0
1.0,  8.0,  5.0, 0.0
1.5, 10,0,  6.0, 0.0
2.0, 12.0,  3.0, 0.0
3.0,  8.0,  0.0, 0.0
4.0,  0.0,  0.0, 0.0
```

RG3

```
3.0,  0.0, 0.0,  0.0
4.0,  4.0, 0.0,  0.0
5.0,  0.0, 0.0,  0.0
```

RG4

```
3.0,  1.0,  0.0, 0.0
3.5,  6.5, -2.0, 0.0
4.0,  8.0, -5.0, 0.0
5.0, 15.0, -6.0, 0.0
6.0, 18.0, -8.0, 0.0
7.0, 12.0, -2.0, 0.0
```

Notes:

1) the various times (first entry of each line) on the different files need not correspond with each other.

2) In file RG1, there will be a u-gust of 5.0 at time 0.0. This gust will not appear on the trim data or affect the trim/stability derivative calculations. It will appear at time 0.0 on the printout and will be used in the time history solution for time 0.0.

3) The user, if he wishes to have the gusts set to zero after a certain point, must input zero gust values at that time. The program does not zero them off at the end of the data file, but rather continues the last input indefinitely. For example, if these gust files were being used for a 10 second time history, source RG1 would continue to have a gust velocity vector of (2.0, 1.0, 0.0) from the third second through the end of the time history. Source 2 (RG2) will have (0.0, 0.0, 0.0) gust velocity after time 4.0 seconds.

4) In files RG3 and RG4, the first line of input has a time of 3.0. The program, will automatically set the gust velocities to zero at time 0.0 and will then proceed to interpolate between zero and the first user input gust velocity at, in this case, three seconds. The velocity vector of (0,0,0) at 3.0 seconds in file RG3 causes no gust to appear at source 3 until after the third second. In file RG4, there is a gust of 1.0 at three seconds causing gusts to appear immediately after Time 0.0 [i.e., 1.0 second has (0.333, 0.0, 0.0) and 2.0 seconds has (0.667, 0.0, 0.0)].

ORIGINAL PAGE IS
OF POOR QUALITY

H. DATA FILE: OUTLST

Contents: The code numbers indicating which variables the user wishes to have printed on the output listing

This file consists of two lists of integers. The first list contains the code numbers for the hull output variables, and the second contains the code numbers for the lift propulsion unit output variables. Each list must be terminated by a zero or negative number. Appendix D contains a list of these variables and their code numbers. The order in which the code numbers appear in the input file determines the order in which the variables will be printed on the output listing. A variable can be requested twice and it will be printed twice in the order in which it is listed. The following example shows the input and the associated output.

```
  file OUTLST

  001                       Resulting Output
  002              U       V       W        P       Q       R
  003      HULL   xxx     xxx     xxx      xxx     xxx     xxx
  004
  005             PHID   THETD   PSID       U       V       W
  006      LPU1   xxx     xxx     xxx      xxx     xxx     xxx
  000      LPU2   xxx     xxx     xxx      xxx     xxx     xxx
  004      LPU3   xxx     xxx     xxx      xxx     xxx     xxx
  005      LPU4   xxx     xxx     xxx      xxx     xxx     xxx
  006
  001
  002
  003
```

I. FILE NAME: INPUT (TAPE5 INTERACTIVE RESPONSES)

The program was developed on an interactive system with five questions to establish the basic run parameters. The user must input the response to these five questions in the data file INPUT (TAPE5) and load it with the program. The questions are:

1. "Six degree of freedom simulation? T/F"

   T — The program will continue after completing the trim and stability derivatives to run a time history. The time history uses the data read from data file HISDTA, and will start with the last trim condition calculated.

   F — The program will stop after completing its trim calculations. File HISDTA is not necessary in this situation.

2. "How many trim flight conditions?" The answer must be a positive non zero integer in the first two columns (I2 format). This number must be less or equal to the number of sets of trim data found in file TRMDTA. If the number of trim data sets is more than this number, the extra trim sets will be ignored. If there are not enough trim sets to match this number, the program will abort for attempting to read beyond the end of file.

3. "Generation of plotting files? T/F"

   T — All output variables which are calculated during the program run will be written to a file (PLOT) at the end of each algorithm step. This file is to be read by a post processor program called PPLOTF.

   F — No plotting files will be written.

4. "Do you want English units? T/F"

   T — English units will be printed with the input variables. The power requirements will be in horsepower.

   F — Metric units will be printed with the input variables. The power requirements will be in Kilowatts.

NOTE: No numerical conversions are done to the input data. Thi flag only causes the program to print either metric or English units, and calculate the power appropriately.

5. "Full header T/F"

   T — The output listing will include a complete list of the input parameters.

   F — Only an abbreviated form of the output listing will be printed. Most of the input variables are not on this listing. All output variables and calculated data is included in this abbreviated listing.

6. "Any comments? (6 lines)" At this point the user may input six lines of alphanumeric information. This information will be printed at the top of the output listing and is intended to allow the user to record some descriptive comments about the particular simulation run.

J. DATA FILE: ERMSSG

Contents: Error Messages

This data file is supplied with the program: it is not produced by the user. It contains all error messages which may be printed by any of the programs. Each message is three lines:

1) **Code number (I3 Format)**
2) **Up to 80 characters of the message (8A10 Format)**
3) **Last 40 characters of the message or a blank line if not needed (4A10 Format).**

The user should refer to the Programmers' Manual Error Processing section if it is necessary to extend this file. Appendix E of this manual contains a list of the error messages and a brief explanation of each.

Data file ERMSSG follows:

```
001
ABSOLUTE VALUE OF PANGLE IS GREATER THAN 1/2 PI.
-
002
CONTROL COMMAND TIMES WERE NOT INPUT IN INCREASING ORDER.
-
003
CONVERGED SOLUTION OF CT AND WIN IS INCORRECT.
-
004
CT AND WIN DID NOT CONVERGE.
-
005
TVC COLUMN NUMBER EXCEEDS MAXIMUM ALLOWED.
-
006
TVC ROW NUMBER EXCEEDS MAXIMUM ALLOWED.
-
007
SROWN WILL EXCEED 30.
-
008
GUSTT1 IS GREATER-EQUAL TO GUSTT2.
-
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
009
LCS OR SIGMA IS LESS THAN ZERO.
-
010
LENGTH OF VCTR IS NOT 6, 12, 24 OR 42.
-
011
MORE THAN 20 CONTROL COMMANDS WERE INPUT.
-
012
NO REAL POSITIVE ROOTS WERE FOUND BY THE IMSL ROUTINE.
.
013
CURRENT AERODYNAMIC ANGLES DO NOT SATISFY ANY OF THE POSSIBLE CONDI-
TIONS.
-
014
SQROOT IS NEAR ZERO.  POSSIBLE DIVISION BY ZERO.
-
015
STABILITY DERIVATIVE WILL NOT BE CALCULATED FOR THIS TRIM.
-
016
STALL REGION ANGLE 1 IS GREATER THAN STALL REGION ANGLE 2.
-
017
SOME OF THE STALL REGION ANGLES WERE NEGATIVE
-
018
SOME OF THE AERODYNAMIC ANGLES OF THE TAIL ARE GREATER THAN PI.
-
019
TIME IS GREATER THAN LAST COMMAND TIME WHICH SHOULD BE THE SAME AS THE
FINAL SIMULATION TIME.
020
TIME IS LESS THAN THE FIRST COMMAND TIME WHICH SHOULD BE ZERO.
-
021
T1COM IS GREATER-EQUAL TO T2COM.
-
022
INCORRECT INPUTS
-
023
IMSL ROUTINE HAS RETURNED AN ERROR FLAG.  ROUTINE NAME IS THE FIRST
VARIABLE GIVEN BELOW.
024
CDFLAG IS NOT SET TO -1, 0 OR 1 ON RETURN FROM SUBR. ITERCT.
-
```

```
025
LESS THAN 4 ZEROS WERE FOUND BY IMSL ZRPOLY
-
026
REQUIRED TRIM CONTROL EXCEEDED AVAILABLE INTEGRATION LIMITS.  IF LOOP
CLOSED THE INTEGRATOR WILL BE SET TO LIMIT
027
THE TIME HAS MOVED BACK FURTHER THAN THE RANDOM GUST BUFFER HAS STORED
GUST VALUES.  REDUCE TIMSTP ON INPUT.
028
THE TIME READ FROM THIS FILE IS LESS-EQUAL TO ZERO.  THE TIME AND GUST
VELOCITY WILL BE IGNORED.
029
THE TIME IS GREATER-EQUAL TO 100000.
-
030
CONDITION FLAGS FROM IMSL ROUTINE DVERK.
-
031
TIME INCREMENT IS LESS THAN ZERO
-
032
THE LENGTH OF THE VECTOR PASSED INTO PPTURB IS NOT 6 OR 12.
-
033
THE VALUE OF VCTRFL IS NOT VALID
-
034
SOME OF THE INVALID STABILITY DERIVATIVES HAVE NOT BEEN FLAGGED BECAUSE
THE ARRAY IS FULL.
035
THE LINEARIZATION LINEAR INCREMENTS ARE LARGE ENOUGH TO CAUSE SOME OF
THE CABLES TO GO SLACK.  THEY ARE BEING RESET
037
THE LENGTH OF THE SV VECTOR IS NOT CONSISTANT WITH THE SIZE OF THE BLANK
BLOCKS FOR EXTRA INTEGRATOR STATES.
038
THE TIMSTP OR MINSTP INPUT IS GREATER THAN THE APPROX. PERIOD/10 AND MAY
CAUSE NUMERICAL INACCURACIES.
039
IMSL DVERK WAS UNABLE TO REACH THE SPECIFIED ERROR CRITERIA WITHOUT
GOING BELOW THE MINIMUM TIME STEP.
040
THE FLAG FOR THIS SUBROUTINE WAS NOT FOUND IN THE DATA FILE
-
041
WAKE ANGLE 1 MUST BE LESS THAN ANGLE 2, AND BOTH MUST BE BETWEEN 0 AND
2*PI
-
```

042
SOME OF THESE VALUES WILL CAUSE DIVISION BY ZERO.
-
043
MORE THAN THE MAX NUMBER OF OUTPUT VARIABLES WERE REQUESTED.
-
044
AN INITIAL GUESS WITH LANDING GEAR IN GROUND CONTACT AND PITCH ANGLE LESS THAN 1. COULD NOT BE FOUND.
045
LINEARIZATION INCREMENT COULD LIFT ONE OF THE LANDING GEARS OFF THE ROUND. IT IS BEING RESET.

SECTION 3

PROGRAM OUTPUT

A. OUTPUT LISTING

The output listing is written to file OUTPUT (TAPE6). The discussion which follows assumes the user has input "True" to the interactive question "Full Header?" (see the interactive questions section). If the has user input "False" in response to this question, the output listing will be abbreviated. While reading this section of the manual, the user will need to consult Appendix C (sample listings) of the User's Manual Appendices, (Volume IV).

The listing starts with a title and a block of information entitled Run Description. These six lines, describing the run which is being made, are provided by the user in response to interactive question "Any Comments?" This is followed by a complete listing of the program input variables. These values are listed with the variable name and brief descriptions of each variable. A more detailed definition of these variables can be found in the Programmer's Manual Appendices. The sections entitled "Geometry Inputs" through "Mechanical Flight Control System Constants" define the basic vehicle configuration. The variables found in sections entitled "Inertial Vehicle State Inputs" through "Stability Derivative Flags" define the basic trim condition and stability derivatives matrices to be calculated.

Following these inputs, the results of the trim calculations are printed. The "Trim Case Number" counts the trims which are calculated during this particular run. The next statement indicates whether the trim reached a converged solution or not. If the trim did not converge, the data printed here represents the best solution found. The trim algorithm control variables may indicate the nature of the problem if the trim does not converge. A complete discussion of the trim is contained in Section 5.

The data from the trim calculation consists of a standard data frame starting with "Simulation Time", which will be a negative number. These negative times will count toward zero for each trim solution calculated (if there are 3 trim calculations, the simulation times would be -3, -2, and -1). The data which follows are the user selected variables (see input file OUTLST). There are more than 300 variables related to the hull which the user may request and more than 150 variables related to each LPU. The user should refer to the Section 2-H for an explanation on how to choose these variables. Appendix D at the back of this manual contains a list of the available output variables and their code numbers.

At the end of this data frame the "Tail Aerodynamic Regimes" are printed for angle of attack, sideslip, and rolling angle of attack.

- "1" indicates prestall regime,
- "2" indicates transition regime,
- "3" indicates the stall regime.

Additional messages may be printed indicating special conditions that exist at this time. They include:

1) Vortex ring state condition on any propeller or rotor.
2) Any maximum control limits which have been encountered (see data file PLMDTA).
3) Any parts of the vehicle which are in ground contact; including landing gears, landing gear frame, tail, belly, bow, or stern.
4) Hover control being turned on.

At the end of each trim data frame the stability derivatives, if requested, are calculated and written to the output listing. Section 6 of this manual discusses the stability derivatives.

If the user has input a number greater than 01 in response to the interactive question "How Many Trim Flight Conditions?" the program will read a new set of trim input data from data file TRMDTA and calculate a new trim condition. Note, the user must have sufficient data blocks in data file TRMDTA to match the number of trim conditions he requests. The output listing will repeat all the information from "Inertial Vehicle State Inputs" through the Stability Derivatives as well as the new trim calculation results.

If the user has input False to the interactive question "Time Histories?" the program will now stop. If the user input True, the program will read the time history data from file HISDTA and write that data to the output listing. That data includes "Flight Control System Configuration," "Time History Profile," "Gust Inputs," and "Computer Algorithm Time Steps Input."

Following the input of the above data, the program will print a data frame at time 0.0 (i.e., before doing any time integration). This output should be the same as the trim data frame (time -1.0), except for variables related to time dependent inputs. For example, if the user has input a gust string with a nonzero gust at time 0.0, the gust value printed will not be 0.0, and the forces related to the wind will be different than the trim values. Similarly, if the user has input a flight control command which is different than the trim control at time 0.0, the flight control system will alter the commands, and consequently, there will be changes in the resulting forces at time 0.0. Test commands can also have the same effect. In all these cases the inputs can alter the forces but they cannot change the vehicle position or velocities at time 0.0. They will cause the vehicle to immediately begin moving away from the trim condition when the integration begins.

After printing the time 0.0 data frame, the program integrates the vehicle states forward in time until reaching the user input print time step. The data frame is again printed and the process is repeated until reaching the simulation time. The program then stops.

If the program encounters an error condition at any point in the simulation, a message is printed immediately. Consequently, the error occurred during the following, not the preceding data frame calculation. If an error message appears, the user should consult the Error Processing section of this manual.

B. PLOTTING FILE

If the user enters True in response to the interactive question "Plotting Files?", a binary format file (PLOT) will be created with all of the output variable names and values. This data is written after every trim calculation and at every algorithm (TIMSTP) step (not the print step; see Section 2-F) during the time history run. Therefore, the plotting file will contain much more data than the output listing. All output data generated by the program (see Appendix D) are written on the plotting file. The code numbers in file OUTLST do not effect the variables written to the plotting files. Figure 10 illustrates the structure of the plotting files.

The file starts with the program ID indicating which program generated the file. Following that, the Julian date, and then a record containing the number of hull variables and the number of LPU variables are listed. Following the above three records, the data file may vary. Programs HLASIM and HLAMOR generate identical data files (left column). Program HLAPAY generates a file with the same structure, but intersperses the payload data with the vehicle data (right column).

During the trim calculations, the payload data is written first. (Note, trim calculations are indicated by negative time). If there is more than one trim calculation, the times will count as negative numbers going toward zero. At the last block of trim data (time equal to -1.0) the variable names are printed out following the variables for both the payload and the vehicle.

At this point we enter the time history section of the program. If the user did not run a time history, then, of course, the remainder of the data file will not exist. Immediately following the variable names, the program algorithm time step and the total simulation time is written. Then, the data for each time frame of the time history is written; the time is listed first followed by the variables. This is repeated for every algorithm step until the end of the simulation. The HLAPAY time history data has the vehicle data written before the payload data (the opposite of the trim).

Program ID - HLASIM, HLAMOR, or HLAPAY
Date (Julian)
Number of hull variables, number of LPU variables

	Program ID HLASIM or HLAMOR	Program ID HLAPAY
		Number of Payload variables and number of cable variables.
		Time (Negative during Trim) Payload variables Cable variables
Repeated for multiple trims, until time = -1.0		(When Time = -1.0) Payload variable names Cable variable names
	Time (Negative during Trim) Hull variables LPU variables	Time (Negative during Trim) Hull variables LPU variables
	(When Time = -1.0) Hull variable names LPU variable names	(When Time = -1.0) Hull variable names LPU variable names
Only if Time Histories	Algorithm time step and total simulation time	Algorithm time step and total simulation time
Repeated for each algorithm step	Time (positive during time history) Hull variables LPU variables	Time (positive during time history) Hull variables LPU variables Time (positive during time history) Payload variables Cable variables

Figure 10. Format of the Binary Plotting File (PLOT)

This plotting file is to be used by a post processor which generates files appropriate to the user's plotting equipment and software. The variable names are written in the same order as the variables, allowing them to be paired by use of array subscripts. Appendix D contains the variable names and the code numbers. By matching variable names and variables, and using the time (negative for trim, positive for time history) the post processor can sort the data.

As part of the implementation of this project on the NASA/Ames 7600/Scope System, a post processor is being supplied to generate plotting files of the time history data (see Section 9 of this manual). The user may wish to write a post processor which will construct plotting files for plotting sets of trim solutions in a manner similar to the time history post processor.

SECTION 4

TRIM

The trim algorithm is a generalized secant scheme which iterates until controls are found which reduce the accelerat'ons below the trim tolerance. The trimmer manipulates six general controls in searching for the converged solution.* They are:

UDCNTL	-	x-direction control
VDCNTL	-	y-direction control
WDCNTL	-	z-direction control
PCONTL	-	roll control
QCDNTL	-	pitch control
RCONTL	-	yaw control

The limits are:

Maximum Iterations	=	200
Maximum Restarts	=	6
Maximum Trim Tolerance	=	10^{-9} (Convergence Criteria)

The "Iterations" are a count of the number of trim algorithm passes used in converging to the solution. The "Restarts" are a count of the number of times the trim algorithm was restarted because it encountered a local solution which did not conform to convergence requirements (i.e., "local minimum"). The trim is stopped when the convergence criteria is satisfied (converged), or the maximum allowable iterations or restarts is exceeded (i.e., solution not converged). The convergence criteria is a measure of the trimmed acceleration of the vehicle. The trim algorithm is based on finding controls which zero all accelerations

*While reading the following discussions the user should refer to the sample output listings in Appendix C.

acting on the vehicle for the initial conditions specified by the user on input file TRMDTA.

The "Trim Controls" are the six linked controls which achieve the convergence criteria for a specific flight condition. The following set of flags count various error conditions that may occur during the trim calculations:

TTHER - The number of times the collective pitch on each rotor exceeded the user input limits (data file PLMDTA).

TTHEP - The number of times the collective pitch on each propeller exceeded the user input limits (data file PLMDTA).

TA1SR - The number of times the lateral control axis deflection on each rotor exceeded the user input limits (data file PLMDTA).

TB1SR - The number of times the longitudinal cyclic pitch angle on each rotor exceeded the user input limits (data file PLMDTA).

SNGMTX - The number of times the trim algorithm encountered singular matrices.

AILERON - The number of times the aileron control exceeded the user input limit (data file PLMDTA).

ELEVATOR - The number of times the elevator contr exceeded the user input limit (data file PLMDTA).

RUDDER - The number of times the rudder control exceeded the user input limit (data file PLMDTA).

Note, the four numbers under the rotor and propeller elements refer to LPUs 1-4 respectively.

These flags will sometimes indicate the nature of the problem if the trim does not converge, i.e.,

```
THEP 0    58    0    52
```

indicates the trimmer was asking for more than the maximum collective pitch on LPU propellers 2 and 4 (yaw control) than available (THEPMX in data file PLMDTA).

SECTION 5

STABILITY DERIVATIVES

Following each trim solution, stability derivatives will be calculated upon request and printed. The stability derivative output begins with a header which describes the basic form of the matrix equation that generated the stability derivatives. There are four basic matrices and four auxiliary matrices (the user has the option to choose any or all of the matrices, see data file TRMDTA). These matrices are generated by the forward-backward difference scheme on each of the elements of the vehicle state vector, the individual control elements, the linked controls, and the gust values at each vehicle component. The four matrices are:

1) A Matrix - The A Matrix is generated by perturbing each of the elements of the state vector.

2) The B′ Matrix - The B′ Matrix is generated by perturbing each of the six linked controls.

3) The B Matrix - The B Matrix is generated by perturbing each of the control elements on the four LPUs (rotors and propellers) as well as the three tail controls.

4) The C Matrix - The C Matrix is generated by perturbing each of the gust values on the vehicle, tail, and the four LPUs.

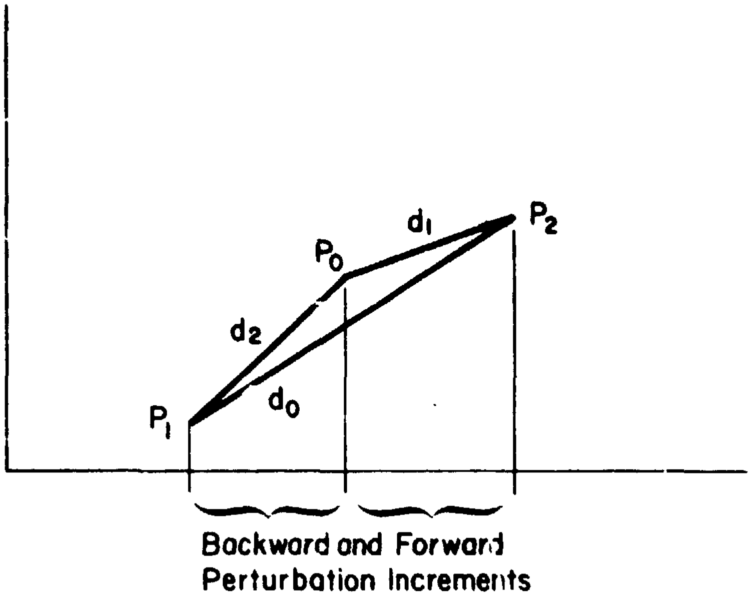
Along with the basic stabililty derivative matrices, a set of auxiliary matrices are generated, based on the constraint forces between the vehicle and the four LPUs. These matrices, A Auxiliary, B′ Auxiliary, B Auxiliary, and C Auxiliary, correspond with the A, B′, B and C Matrices.

In the process of calculating these matrices the program checks for strong nonlinearities. Following the matrices, the program lists those elements of the matrices where nonlinearities were detected. It prints the derivative at the positive increment and the negative increment of the forward-backward scheme so the user can see the nature of the nonlinearity. Figure 11 illustrates the three derivatives in relation to the forward/backward perturbations.

The eigenvalues and normalized eigenvectors (real, imaginary parts) are then printed. The normalized eigenvectors are printed in columns directly below the corresponding eigenvalues. The IMSL performarce index is an indication of the accuracy of these values. A performance index below 1.0 indicates the accuracy is excellent. An index between 1.0 and 100, indicates that the accuracy is good, and indices greater than 100 indicate poor accuracy. The user is refered to the IMSL Manual (subroutine EIGRF) for a more detailed explanation of this performance index.

Finally, for the user's information, the perturbation increments which were used in the calculation of the stability derivatives are printed. These increments are set internally in the program (subroutine INTIAL).

ORIGINAL PAGE IS
OF POOR QUALITY



P_0 - Initial values
P_1 - Result of negative perturbation
P_2 - Result of positive perturbation

Slope d_1 - Positive increment derivative } printed with nonlinear element
Slope d_2 - Negative increment derivative } printed with nonlinear element
Slope d_0 - Derivative values in stability derivative matrix

If $|d_2 - d_1| \geq 0.3\, d_0$ the element will be flagged as invalid

Figure 11. Stability Cerivative Calculation Algorithm

SECTION 6

ERROR PROCESSING

There are several different conditions which cause the program to print an error message. They are:

1. User errors - These could be invalid inputs, numbers which may cause division by zero, or other values which the program determines are invalid for that particular variable.

2. Model error conditions - Conditions with which the program models have not been designed to cope. They may not necessarily be wrong, but only that the program is unable to deal with them.

3. Conditions of particular interest — These are conditions which may be of interest to the computer programmer or the engineer.

These conditions may or may not cause the program to stop. If the condition does not invalidate the calculations, the program is allowed to continue. The error messages are input as data into the program and are supplied with the program (see input data file ERMSSG). Appendix E lists the error messages which may be printed with an explanation of what they mean and suggestions for rectifying the problem.

A typical error message consists of a line of stars, followed by a one line error message, the subroutine name which detected the error, and up to three variables with their present values from that subroutine. These variables should prove useful if it is necessary for the user to consult the program code to ascertain the cause of the error. The following example shows the format of the error message.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
22 INVALID INPUTS
SUBROUTINE INLARO
BETA1T  =  .6981E+00
BETA2T  =  .5236E+00
```

In this particular casc the input routine INLARO has detected that BETAlT is not less than BETA2T as it should be. "22" is the error message code number (see data file ERMSSG).

SECTION 7

PROGRAM HLAPAY; SIMULATION OF A HEAVY LIFT AIRSHIP CARRYING A PAYLOAD ATTACHED BY CABLES

A. INTRODUCTION

This program models the same vehicle as program HLASIM. It has, in addition, the models of a slung payload and supporting (elastic) cables. The payload and cable models were developed on an entirely separate program. HLAPAY contains both the program HLASIM and the payload-cable program. They interact through the cable forces. The user should keep this modularity in mind while creating his data file and studying the output from this program. The reasons for implementing these models in this way are:

1) It allows for a modular development and debugging of the program.

2) Will allow future alterations of either the payload or vehicle to be implemented and tested separately before being put together as one program.

It is expected that the user has already used and has a good understanding of program HLASIM before attempting to use program HLAPAY. Consequently, in this section of the manual there will be no discussion of the use of program HLASIM. We present here a detailed description of the additional inputs and considerations necessary to study and use program HLAPAY. The payload model consists of:

1) A payload module and a cable module.

2) Input subroutines which read and then write to the output listings all the payload related inputs.

3) A payload trim module which uses the same algorithm as the vehicle and determines a payload orientation which is consistent with the vehicle trim flight conditions.

4) A module to calculate the payload state vector derivative which is used both by the trimmer and by the integrator during the time history.

The combined trim is achieved by establishing the payload trim conditions consistent with the vehicle trim input flight conditions. The payload orientation is then calculated to meet the payload trim conditions. On completing this, the values of the cable forces are passed to and become part of the vehicle trim calculations.

The vehicle trim is based on finding commands (6 linked commands) which will balance all forces acting on the vehicle and result in no accelerations at the user input flight conditions. The payload trim uses the same algorithm, but attempts to find a position (linear and angular) where all forces are balanced. The accelerations will be consistent with the hull/payload rigid body motion determined by the vehicle trim flight conditions.

The stability derivative matrices include both the vehicle and the payload states. A detailed discussion of the stability derivatives is in Part E of this section. The time history integration loads the payload and vehicle state vectors into one vector and integrates them forward together. During the time history calculations, the cable spring and damping forces are the "elastic" constraints which control the payload position below the vehicle.

B. VEHICLE-PAYLOAD INPUT DATA FILES

1. Data Files: GMDTA, ARODTA, IFCDTA, PLMDTA, TRMDTA, and HISDTA

These data files are used just as in program HLASIM.

2. Data Files: PAYDTA

PAYDTA contains all the basic payload data including: geometry, mass, aerodynamic and time history variables. (There are no interference effects on the payload). These inputs are listed in Appendix A and the user will have to refer to the Technical Manual for the derivation of the engineering constants. The form of the data very closely parallels that of the vehicle and LPUs. The payload inputs are similar to the hull, and the inputs for the four cables are similar in format to the inputs for the four LPUs. For most of these inputs, the descriptions given in Appendix A and in the Technical Manual are sufficient. Only inputs which have special considerations will be discussed here.

Figure 12 illustrates the four cable positions with respect to the payload and the vehicle. Note, that "attach point 1" on the hull (RATHP1) must correspond with "attach point 1" on the payload (RPTCH1) and those attach points, by definition, are connected by "cable 1" (no crossed cables allowed).

The user may eliminate any of the four cables in a manner similar with the landing gears. To zero off any cable, the spring constant (CABLK) for that particular cable should be set to 0.0. The program will then calculate the force exerted by that particular cable to be zero, effectively removing it from the model. The geometry inputs for that cable must still be in the file but will not affect the calculations.

The restrictions on the payload cable geometry are:

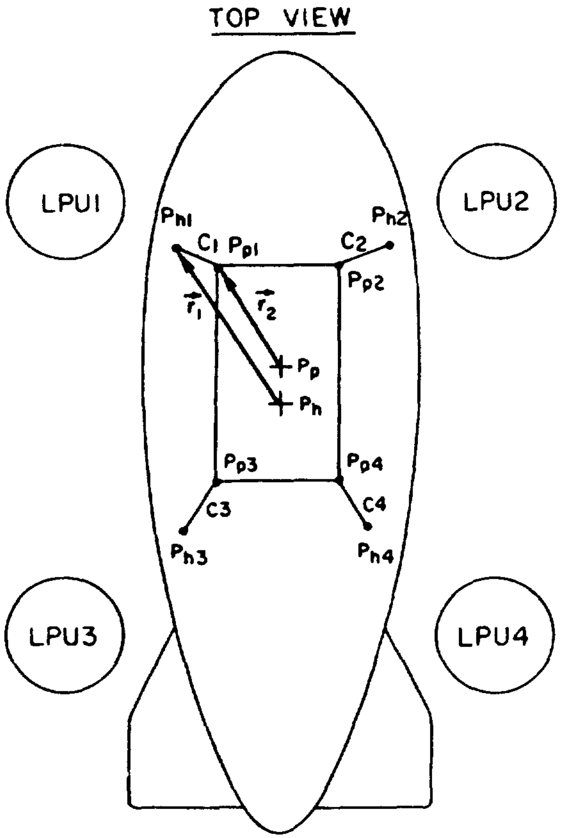
1) All the cable attach points on the payload may not all be on a common line segment that intersects the payload center-of-gravity. This condition cannot be trimmed.

2) If the relative velocity in x-y plane or NUVP is zero, all the active cables must not be attached to the same point on the vehicle or payload as this removes all yaw control and can not be trimmed.

3) The user must insure that none of the active cables (CABLK non-zero) will be slack at the trim position. The trimmer demands that all active cables have some tension in them.

4) "Cable 1" must connect "Hull attach point 1" with "Payload attach point 1", and similarly for cables 2, 3 and 4 (i.e., No crossed cables).

These restrictions are illustrated in Fig. 13.



P_{H_1} - Hull cable attach points (1-4) in hull reference axis
P_{H_2} -
P_{H_3} -
P_{H_4} -

P_{P1} - Payload cable attach point (1-4) in payload reference axis
P_{P2} -
P_{P3}
P_{P4} -

C_1 - USLTH1 Unstretched cable lengths (scalars) Cables 1-4 respectively
C_2 -
C_3 -
C_4 -

P_h Hull reference center

P_p Payload reference center

$\vec{r}_1$ - RATHP1 - Vector from hull reference center to Cable 1 attach point in hull reference axis (RATHP2-4 are similarly placed)

$\vec{r}_2$ - RPTCH1 - Vector from payload reference center to Cable 1 attach point in payload reference axis. (RPTCH2-4 are similarly placed).

ORIGINAL PAGE IS OF POOR QUALITY

Figure 12. Payload Position Relative to Hull - Top View

ORIGINAL PAGE IS
OF POOR QUALITY

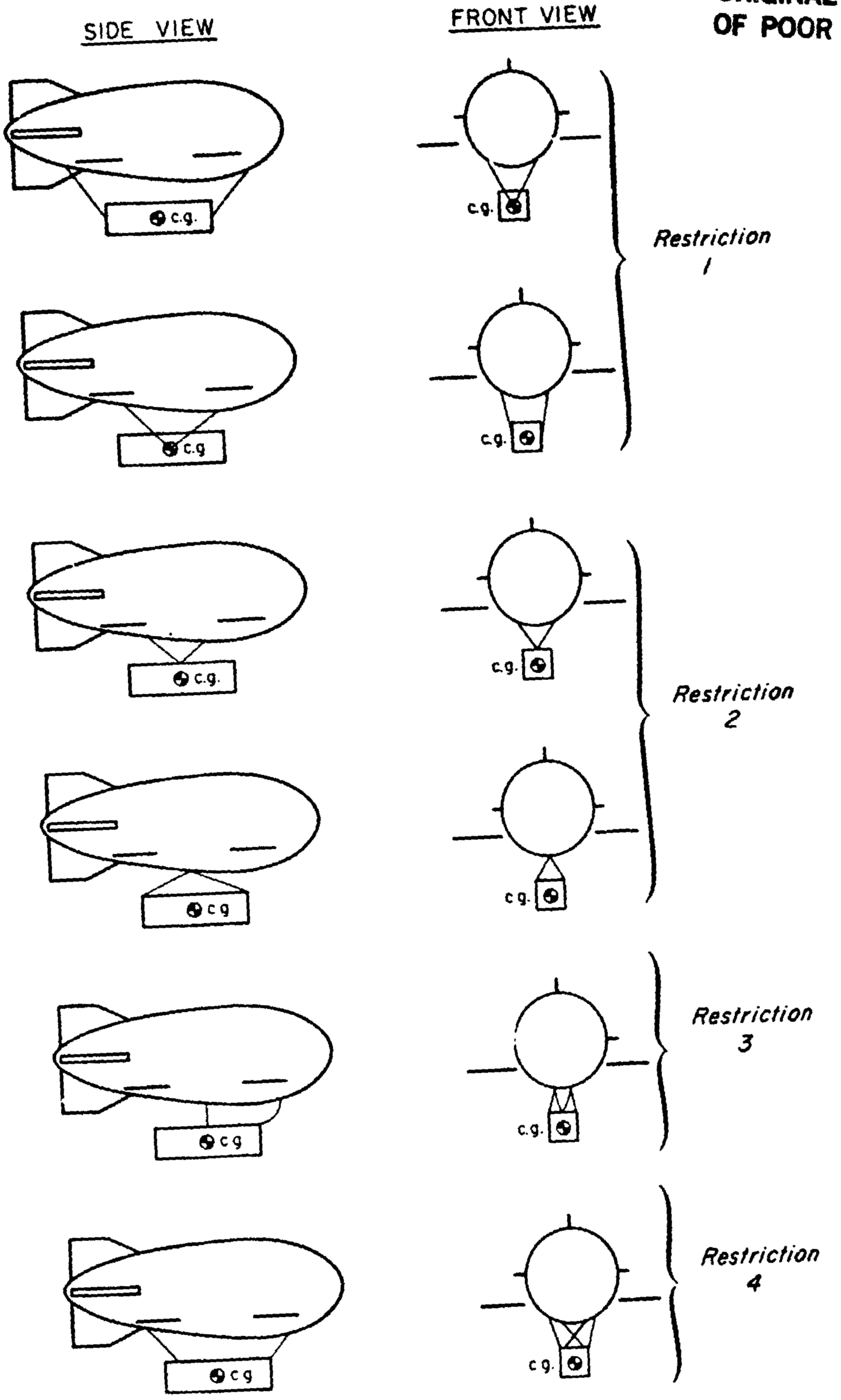


Figure 13. Illegal Cable Geometries

The cables (modeled as springs and dampers) cause the trim calculation to be much more difficult than that for the vehicle. The user's choice of a cable geometry and spring stiffness and damping constants may sometimes cause the trimmer to fail to converge. These conditions are not predictable in that they arise from numerical problems involved with the trim algorithm. There may be cases where the user may find it impossible to model a certain geometry. Figures 14, 15 and 16 includes those geometries which have been implemented successfully during program development.

The payload (1-minus-cosine) gusts are input in the same manner as those for the vehicle (see namelist NPYGCOM). The payload gust strings are slightly different. The payload gust strings are turned on by the flag PGSTFL similar to the vehicle, but payload gust strings are read from two files. The user should read the section on the input files RG5 and RG6 for a description of these files. The linear velocities are read from RG5, and angular velocities from RG6. These linear and angular velocity vectors act at the payload aerodynamic reference center and are scaled by the variables PGVSCF (linear gusts), and POGSCF (angular gusts).

The namelist NINDPST contains a set of variables which are used differently than the other input variables. These variables, which are orientation perturbations away from the calculated payload trim condition, are applied at the beginning of the time history (time = 0.0). These inputs allow the user to displace the payload from its trim condition and study the resulting actions of the coupled vehicle/payload system during the time history. The payload perturbation variables are:

DVPYLD - The displacement from the payload trim velocity.

DHRPYL - The displacement from the payload trim location.

DPYELR - The displacement from the payload trim Euler rates.

DPYEUL - The displacement from the payload trim Euler angles.

ORIGINAL PAGE IS
OF POOR QUALITY

```
$NRPTCH
RPTCH1  =  40.0, -6.0, -6.0
RPTCH2  =  40.0, 6.0, -6.0
RPTCH3  =  -40.0, -6.0, -6.0,
RPTCH4  =  -40.0, 6.0, -6.0,
$END
$NRATHP
RATHP1  =  36.0, 0.0, 50.0,
RATHP2  =  36.0, 0.0, 50.0,
RATHP3  =  -36.0, 0.0, 50.0,
RATHP4  =  -36.0, 0.0, 50.0,
$END
$NUSCLTH
USLTH1  =  22.0,
USLTH2  =  22.0,
USLTH3  =  18.0,
USLTH4  =  18.0,
$END
$NRPAYCG
RPAYCG  =  0.0, 0.0, 0.0,
$END

$NCABLK
CABLK1  =  .9643E + 04
CABLK2  =  .9643E + 04
CABLK3  =  .9643E + 04
$END
$NCABLC
CABLC1  =  .3096E + 03
CABLC2  =  .3096E + 03
CABLC3  =  .3096E + 03
CABLC4  =  .3096E + 03
$END
```

Figure 14. Cable configuration 1 (Inverted "V" and Front Lower then the Rear)

ORIGINAL PAGE IS
OF POOR QUALITY

```
$NRPTCH
RFTCH1  =   40.00,     -6.00,        -6.00,
RFTCH2  =   40.00,      6.00,        -6.00,
RFTCH3  =  -40.00,     -6.00,        -6.00
RFTCH4  =  -40.00,      6.00         -6.00,
$END

$NRATHP
RATHF1  -   36.00,     -6.00,        50.00,
RATHF2  =   36.00,      6.00,        50.00,
RATHF3  =  -36.00,     -6.00,        50.00,
RATHF4  =  -36.00,      6.00,        50.00,
$END

$USLTH
USLTH1  =  22.0000,
USLTH2  =  22.0000,
USLTH3  =  18.0000,
USLTH4  =  18.0000,
$END

$NRPAYCG
RPAYCG  =  0.0, 0.0, 0.0,
$END

$NCABLK
CABLK1  =  .3100E+05,
CABLK2  =  .3100E+05,
CABLK3  =  .3100E+05,
CABLK4  =  .3100E+05,
$END

$NCABLC
CABLC1  =  .1243E+04,
CABLC2  =  .1243E+04,
CABLC3  =  .1243E+04,
CABLC4  =  .1243E+04,
$END
```

Figure 15. Cable Configuration 2 (Four Cables)

ORIGINAL PAGE IS
OF POOR QUALITY

```
$NRPTCH
RPTCH1  =   20.00,     0.00,      -8.00,
RPTCH2  =   20.00,     0.00,      -8.00,
RPTCH3  =  -20.00,     0.00,      -8.00
RPTCH4  =  -20.00,     0.00,      -8.00
$END

$NRATHP
RATHP1  =   20.00,     0.00,      50.00,
RATHP2  =   20.00,     0.00,      50.00,
RATHP3  =  -20.00,     0.00,      50.00,
RATHP4  -  -20.00,     0.00,      50.00,
$END

$NUSLTH
USLTH1  =  17.5000,
USLTH2  =  17.5000,
USLTH3  =  17.5000,
USLTH4  =  17.5000,
$END

$NRPAYCG
RPAYCG  =  0.0,        0.0,       0.0,
$END

$NCABLK
CABLK1  =  .9643E+04,
CABLK2  =  0.0,
CABLK3  =  .9643E+04,
CABLK4  =  0.0,
$END

$NCABLC
CABLC1  =  0.3096E+03,
CABLC2  =  0.0,
CABLC3  =  0.3096E+03
CABLC4  =  0.0,
$END
```

Figure 16. Cable Configuration 3 (Bifilar Cables)

3. Data File: PYOUTL

Data file PYOUTL contains the two lists of code numbers for payload outpuc variables. This data file is identical in format with data file OUTLST. Each list of positive numbers is terminated by zero or a negative number. The first list contains the code numbers of the payload variables, and the second contains code numbers of the cable variabies to be printed. The cable variables, like those of the LPUs, will be printed in groups of four, one value for each of the four cables under the variable name.

4. Data Files: RG5 AND RG6

These two data files have the same format as RG1-RG4. Each line of the data file contains four numbers. The first number is the time and the following three numbers are the gust source vector for that time. Data file RG5 contains linear gust velocities at the payload aerodynamic reference center. RG6 inputs contains angular gust velocities at the payload aerodynamic reference center. Payload gusts strings are different than the vehicle gust strings in that they are applied directly to the payload, without spatial interpolation. This requires angular velocities to be input, not calculated, hence the need for RG6. As before, interpolation is used to obtain the gust values at the appropriate simulation time, and coordinate transformations convert the input data to the payload reference axes.

The reference center for these gusts like the vehicle translates with the payload, but it does not rotate with the payload. Consequently, the gusts input in file RG5 and RG6 are in terms of the inertial reference axis, although their center is translating with the payload. The gust values being input at the vehicle sources are not interpolated down to the payload nor are the payload gust values interpolated up to the vehicle.

5. Data File: INPUT (TAPE5 Interactive Questions)

The interactive questions for the payload program are the same as those for the main program (HLASIM).

C. PROGRAM HLAPAY OUTPUT

The output listing for this program contains all the input values for program HLASIM as well as the payload module. Appendix C at the back of this manual contains a sample output listing (Appendix B contains the corresponding input data files).

Following the list of input values, the payload trim conditions are printed. The "Payload Trim Case Number" is a counter of the trims being calculated (similar to HLASIM). The trim algorithm control is printed next, indicating the converged solution or the best guess achieved during the trim calculations.

The payload trim algorithm is the same as that for HLASIM, and the meaning of "Iterations", "Restarts", and "Convergence Criteria" are the same. The variable HRPYLC is the vector indicating the payload location relative to the hull, and the three variables, PAY PHI, PAY THETA and PAY PSI are the three Euler angles of the payload. These six numbers are the payload trim controls which are calculated during payload trim.

During the payload trim calculation, there are two conditions encountered which are flagged as model errors. The position limit, HRPLFL, indicates that a guess for the payload position violated the condition that all active cables must have some tension for a valid trim. This particular variable is often flagged a few times during the trim calculation and does not indicate there is anything wrong. If the trim failed to converge and this variable is a large number, then the user geometry is probably such that one of the cables is carrying very little weight and trim guesses frequently causes it to be slack (restriction 2, Section 7-B2). This can be remedied by altering the cable geometry slightly. The second error condition, indicated by counter PSNGMT, is a singular matrix. A large number for PSNGMT during a failed trim calculation often indicates violation of or near violation of payload geometry restrictions 1 or 2.

Following this, the payload variables are printed. The time indicated here is a negative number and functions in the same manner as the time does for the vehicle (it will be the same time as the corresponding vehicle trim). The variables printed are those variables the user has chosen through the input data file, PYOUTL.

The vehicle trim data follows the payload data. This order indicates the order of the trim calculations. The user should consult Section 3 and 4 for a discussion of the vehicle trim.

Program HLAPAY will generate a plotting file (PLOT) containing all the output data. The plot file has structure similar with the file generated by program HLASIM. The post-processor PPLOTF (part of the Heavy-Lift Airship Simulation package) will read this file and produce time history files compatible with NASA/Ames plotting software. This file is described in detail in Section 3-5 and 9 of this manual.

D. HLAPAY TRIM LIMITATIONS AND DIFFICULTIES

The payload trim is much more difficult than the vehicle trim. This is caused primarily by the springs that are modeled in the cables. The user's cable geometry has a large effect on the trim. If the trim will not converge for a particular geometry, there are several things which should be tried:

1) Insure that none of the cable geometry restrictions are violated (see Section 7-B).

2) A reduction in the spring stiffness constants will often improve the payload convergence characteristics.

3. The user can alter his cable geometry slightly, as the program sometimes encounters numerical conditions which it can not handle and a slight change will avoid those problems.

The discussion of data file PAYDTA (Section 7-B-2) contains a list of cable geometries which have been implemented successfully during program development

E. HLAPAY STABILITY DERIVATIVES

The stability derivatives for the program HLAPAY contain the same basic matrices as are found in program HLASIM. Each of the four main matrices (A-Matrix, B-Matrix, B′-Matrix and C Matrix) have 24 rows instead of 12 (12 payload states are added to the 12 vehicle states). Each of the auxiliary matrices contain 28 rows instead of 24. The four additional rows contain the cable force derivatives.

The matrices generated are:

A and A Auxiliary Matrices — Generated by perturbing the 24 (12 vehicle and 12 payload states).

B and B Auxiliary Matrices — Generated by perturbing the 27 vehicle control element effectors (same as program HLASIM).

B′ and B′ Auxiliary Matrices — Generated by perturbing each of the six linked vehicle controls (same as program HLASIM).

C and C Auxiliary Matrices — Generated by perturbing the 48 (42 vehicle and 6 payload) gust velocities and velocity gradients).

The same testing is done for nonlinearities and those found are listed on the output, as in program HLASIM. Because of the symmetry of the vehicle and likely cable configurations, many of the cable derivatives will be flagged as nonlinear when the payload position is perturbed (i.e., pertubations right and left both increase cable tension).

There are two methods to obtain stability derivatives of the payload alone

1) Make the vehicle much larger (order of 100), so that the ratio of payload to vehicle mass is small. The input variables involved are:

   HULVOL
   MASHUL
   IHULXX
   IHULYY
   IHULZZ
   IHULXZ

The values for MASHUL and HULVOL must be such that the rotors are able to support the difference between g(MASHUL) and the buoyancy.

The resulting matrix rows and columns pertaining to the payload only will be the decoupled payload stability derivatives. Similarly, the eigenvalues and eigenvectors pertaining to the payload are those for the decoupled payload system only.

2) Subroutine PLINAR calculates the payload only stability derivatives. This module is in the program library, but it is not called in the overlay structure at present. The subroutine TLINAR, called from the main program HLAPAY, calculates the combined vehicle/payload stability derivatives. If the user wishes to have only the payload stability derivatives calculated, he should change this call to subroutine PLINAR. There are no other changes necessary to the code, and the program will run as usual. The trim will be calculated for the payload and the vehicle, but stability derivatives will be calculated for the payload alone as if attached to an infinite mass vehicle.

The above methods provide information on the payload alone characteristics, without the complex coupling between the vehicle and payload systems. If a new payload model is developed, these payload only stability derivatives may prove useful in debugging and verifying the payload model. This would be much cheaper since the program will calculate only the elements of the payload only derivative matrices. The user should refer to the Programmer's Manual for detailed description of the structure of the stability derivative modules if he is contemplating making these changes.

SECTION 8

PROGRAM HLAMOR; SIMULATION OF THE MOORED UNPOWERED HEAVY LIFT AIRSHIP

A. INTRODUCTION

This program simulates an unpowered heavy lift airship which is translationally moored to a mast. Since the mooring point is modeled as a translationally constrained perfect gimbal, the airship has only three degrees of freedom (angular gimbal motion). No linear motion at the mast attach point is allowed. The vehicle is unpowered and consequently has only minimal rotor and propeller aerodynamics. The trim is considerably different than that for HLASIM, as it is based on the mooring point constraint, the active landing gear forces, and the wind forces. There is no flight control system and no control commands. The only data which disturbs the trimmed vehicle are the gusts and the mooring state displacements.

B. MOORED VEHICLE INPUT DATA FILES

This program uses the same basic data and consequently will model the same vehicle as program HLASIM. Many of the input data for program HLASIM are not used in the mooring program because of the unpowered condition. Even so, the input files have been structured in such a manner that the same files that were used by program HLASIM may be used here. This program will ignore those inputs which are not relevant to it. This has been done to simplify the user's problem of insuring the data used in HLASIM and HLAMOR are the same.

1. Data Files: GMDTA, ARODTA, IFCDTA, TRMDTA, HISDTA

These files may be used just as they are from program HLASIM. Note, file PLMDTA is not used at all in this program.

In file TRMDTA, the only necessary data is contained in namelists NATMOS and NSTABDV. The data in namelist NINSTAT is not used. For this reason the flag "SUBROUTINE INATMOS" has been inserted in this file just proceeding the namelist "NATMOS". This allows this program to read past the unwanted data. It is highly recommended that the user, if he is modeling the same vehicle in program HLASIM and HLAMOR, use the complete data file and not remove the unnecessary data. Otherwise, the user may inadvertently alter data in the file, and invalidate the comparison between different runs.

Similarly in file HISDTA, the flight control system data is not necessary to the program. The program reads past all the data until it encounters the flag "SUBROUTINE INGUST" indicating the beginning of the gust inputs.

2. Data File: MORDTA

This data file contains the variables which are exclusive to HLAMOR. The variables are listed in the tables in Appendix A. They are all in namelist format.

The variable PSIO serves two functions:

1) The vehicle inertial Euler yaw angle is set to PSIO if there is no wind.

2) The vehicle Euler yaw angle relative to the wind for the trim initial guess is set to PSIO when there is any x-y plane wind velocity.

The moored vehicle trim may have three converged yaw positions: nose into the wind or yawed positive or negative relative to the wind. The trimmer can be made to converge to any of the three positions by using PSIO to set the initial trim guess close to the trim yaw position wanted. PSIO is the wind relative vehicle trim initial guess Euler yaw angle if the x-y plane wind is nonzero. PSIO = 0.0 causes nose into the wind trim initial guess. The other two positions will depend on the user vehicle characteristics.

The trim solution for a no-wind condition is achieved by creating an artificial yawing moment. It is scaled to the difference between the desired yaw angle (PSIO) and the actual Euler yaw angle, and will only exist during trim (MFORCE, CLMTRM). This places the following restriction on the possible landing gear geometry.

If there is only one landing gear, or if all active landing gears are coplanar and aligned parallel to the mast, then the composite vehicle center of gravity must lie in the (landing gear) plane. Otherwise the artificial moment generator will cause the trimmer to converge on a non-existent solution. This geometry restriction will only apply in the no-wind condition.

The three variables DELTAL, DELTEL, and DELTRD are the tail deflection angles for the ailerons, the elevator, and the rudder. These values are fixed positions which are held throughout the time history.

The variable DHLEUL is a perturbation vector away from the trim value of the Euler angles (similar to the variables in namelist NIHDLST of data file PAYDTA). Its purpose is to allow the user to move the vehicle away from the trim condition at the beginning of the time history and study the subsequent motion of the vehicle. Aside from inputing a gust, this is the only means of causing the vehicle to move away from the trim (during the time history).

3. Data File: OUTLST

Program HLAMOR, like HLASIM, uses OUTLST to specify the output variables to be printed. If the user requests variables which have not been calculated, the program will print either zero or an undefined ("I"). Because HLAMOR is unpowered and does not use the sophisticated aerodynamic models for the rotors and propellers, many of the variables which can be requested are not being calculated.

4. Data Files: RG1, RG2, RG3, RG4

These four gust string input files are used just as in program HLASIM.

C. PROGRAM HLAMOR OUTPUT

The HLAMOR output listing is the same as the HLASIM listing, except the input variables not used in HLAMOR do not appear. Output variables which are not calculated in this program should not be requested through file OUTLST. If they are requested they will appear as zero or undefined values.

The plotting file (PLOT) is the same as the HLASIM plotting file.

D. HLAMOR TRIM CONSIDERATIONS

The trim calculation for HLAMOR is a three degree-of-freedom problem. The vehicle linear position is determined by the user input mast location and mooring point. Consequently, the vehicle can only move in the three angular directions. The trim is much easier and usually requires fewer iterations than program HLASIM or HLAPAY. There are a few restrictions however.

1) The geometry of the mooring point, mast location, and landing gears must be such that all <u>active</u> landing gears will be touching the ground. The landing gear frames, vehicle tail or belly, must not be touching the ground. Deactivating all landing gears will cause a buoyant (airborne) trim to be calculated. In this case the buoyancy and wind must be enough to hold the vehicle belly off the ground.

2) The only physical means for establishing a yaw angle is the force of the wind acting on the vehicle. Consequently, the user must input a value (PSIO) indicating the yaw angle in the event that the wind vector is equal to zero.

3) If only one landing gear is active it must be in the same lateral position as the composite center of gravity of the entire vehicle (hull and LPU's).

E. HLAMOR STABILITY DERIVATIVES

Stability derivatives for the moored vehicle are reduced to the A-Matrix and the C-Matrix and their corresponding auxiliary matrices.

- A-Matrix — This 6 by 6 matrix is generated by perturbing the 3 Euler angles and the 3 Euler angle rates of the vehicle.

- C-Matrix — This 42 by 6 matrix is generated by perturbing all of the gusts acting at the various component positions (hull, tail, LPU1-4); identical to HLASIM.

The auxiliary matrices each contain 28 rows, consisting of the 24 constraint forces and moments between the LPUs and the vehicle, and the four compression forces acting through the landing gears.

SECTION 9

PLOTTING FILE POST-PROCESSOR PROGRAM PPLOTF

Program PPLOTF (process the plotting file) will reformat the plotting file (PLOT) produced by any of the simulation programs (HLASIM, HLAMOR, or HLAPAY). The input for program PPLOTF is the binary format file PLOT. The output is a binary file (THPLOT) of the time history data in a format compatible with the NASA/Ames flight data plotting software. The Heavy-Lift Airship Simulation data will be plotted with the NASA/Ames flight data plotting software.

PPLOTF program writes to the standard output file, OUTPUT, a title which includes the simulation name (HLASIM, HLAMOR or HLAPAY), the date of the simulation and the simulation times (Trim and Time history). This informs the user which simulation run plotting file is being processed. If this program finds something wrong with the input file (PLOT) it will print a message and stop.

File PLOT contains all the trim data, variable names and time history data. Program PPLOTF ignores the trim data. The variable names contained in file PLOT are exactly as they appear on the simulation programs' output listing. They are changed slightly when written to file THPLOT.

1) Each LPU and cable variable name has four values (LPU1-4 and Cable 1-4). This program writes each name four times inserting numbers 1 to 4 in the blank sixth position.

   For example RFIV :X becomes

   RFIV 1:X, RFIV 2:X, RFIV 3:X, and RFIV 4:X.

2) All leading and embedded blanks are then removed.

   RFIV becomes finally:

   RFIV1:X, RFIV2:X, RFIV3:X, and RFIV4:X

3) The independent variable, Time, is given the name "ETIME" and its value is written with the other variables in each data frame.

These alterations are not seen by the user until particular variables are plotted with the NASA/Ames flight data plotting software.

This program is executed by loading the program with the data file PLOT. The file THPLOT should be saved after execution is complete.

Section 3-B contains a detailed description of the file PLOT. The file THPLOT is outlined in the diagram below.

Record 1	Time step, Number of variables, Date (Julian) and Variable names
Record 2	Data time frame 1
Record 3	Data time frame 2
.	.
.	.
.	.
	(Continued until end of time history)

Note: The number of variables includes the time, and the variable names include "ETIME" in the appropriate place. The time is written as a data value in each time frame.

ORIGINAL PAGE IS
OF POOR QUALITY

SECTION 10

LINEARIZATION MATRIX FROM GUST SOURCES TO VEHICLE COMPONENTS

Program GSRCSB is provided to aid gust model analyses. It generates a gust source interpolation matrix (GSI) which shows the relationship between the gust velocities at the four gust sources and the gust velocities and gradients at the vehicle components. Using the vehicle and gust source geometry and che trim states, the program perturbs each gust source velocity and loads the resulting vehicle component gust velocities and gradient derivatives into columns of the GSI matrix. The following equation defines the GSI matrix.

$$\begin{bmatrix} D \\ U \\ G \\ B \\ A \\ R \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \\ P \\ Q \\ \cdot \\ \cdot \\ \cdot \\ W_{g4} \end{bmatrix} = \begin{bmatrix} \text{GSI} \\ \text{Matrix} \end{bmatrix} \begin{bmatrix} U^{s1} \\ V^{s1} \\ W^{s1} \\ U^{s2} \\ \cdot \\ \cdot \\ \cdot \\ V^{s4} \\ W^{s4} \end{bmatrix} \qquad (10\text{-}1)$$

where DUGBAR is the same vector of gust velocities, accelerations, and gradients found in the main program C-matrix stability derivative calculations.

This matrix will change only when the vehicle geometry, gust source geometry or hull Euler trim angles are changed. Consequently, program GSRCSB should be executed only if some of those variables have been altered.

The program uses the main simulation input routines and can read the main program data files. The files required are: GMDTA, ARODTA, TRMDTA, HISDTA and ERMSSG. The output is written to file OUTPUT.

ORIGINAL PAGE IS
OF POOR QUALITY

Program GSRCSB does not calculate the hull or tail gust acceleration rows (U DOT, V DOT, •••, R DOT) of GSI. For example, the row corresponding to the hull gust acceleration $(\dot{u}_h)$ term is:

GSI(1,1) * D/DT, GSI(1,2) * D/DT, •••, GSI(1,12) D/DT

This nomenclature symbolizes the following mathematical expression for $\dot{u}_h$ ($GSI_{1,1}$ denotes row 1, column 1 of the GSI matrix):

$$\begin{aligned}\dot{u}_h = \; & GSI_{1,1}\frac{du^{s1}}{dt} + GSI_{1,2}\frac{dv^{s1}}{dt} + GSI_{1,3}\frac{dw^{s1}}{dt} + GSI_{1,4}\frac{du^{s2}}{dt} \\ & + \cdots + GSI_{1,11}\frac{dv^{s4}}{dt} + GSI_{1,12}\frac{dw^{s4}}{dt} \quad ,\end{aligned} \tag{10-2}$$

and similarly for the remaining hull and tail gust accelerations.

The main program stability derivative C-matrix is generated according to the following equation (see stability derivative output Sec. 5).

$$\begin{bmatrix} X \\ B \\ A \\ R \\ D \\ O \\ T \end{bmatrix} = \begin{bmatrix} \text{C-Matrix} \end{bmatrix} \begin{bmatrix} D \\ U \\ G \\ B \\ A \\ R \end{bmatrix} \tag{10-3}$$

The GSI matrix is used with the main program linearization C-matrix in the following manner. Equations 10-1 and 10-3 are conceptually combined to form Eq. 10-4, which shows the effect of gust source velocities on the vehicle.

$$\begin{bmatrix} X \\ B \\ A \\ R \\ D \\ O \\ T \end{bmatrix} = \begin{bmatrix} \text{C-Matrix} \end{bmatrix} \begin{bmatrix} \text{GSI} \\ \text{Matrix} \end{bmatrix} \begin{bmatrix} u^{s1} \\ v^{s1} \\ w^{s1} \\ u^{s2} \\ \cdot \\ \cdot \\ \cdot \\ v^{s4} \\ w^{s4} \end{bmatrix} \tag{10-4}$$

The method to manipulate the d/d_t terms of the gust acceleration rows (GSI matrix) in the calculation of Eq. 10-4 will depend on the software utilities available to the user.

SECTION 11

OVERLAYS, EXECUTION OF THE PROGRAMS ON THE NASA/AMES CDC 7600/SCOPE SYSTEM

The program has been installed on the NASA/Ames CDC 7600/SCOPE system in the following manner:

1) Simulation program source code is stored on disk using the UPDATE facility.

2) All data files are stored as separate disk files with the names given in Section 2.

3) The three main program routines (HLAMOR, HLASIM, and HLAPAY) are stored as separate disc files as well as in the UPDATE facility (see Item 1).

4) The segmented load directive files are stored as separate disc files (SLDIR, for HLASIM; SLDIRM, for HLAMOR, and SLDIRP, for HLAPAY).

5) The binary object code for all program subroutines (excluding main program routines) is maintained in a Library on a a disk file called MAINLB. It can be accessed and edited with the LIBRARY facility.

6) The post processing programs (PPLOTF and GSRCSB) are in separate disc files. The files are named PPLOTF and GSRCSB, respectively.

A. PROGRAM EXECUTION CONTROL SEQUENCES

Figure 17 (page 79) is a sample job control sequence for executing the main simulation program. A line by line discussion of this sequence follows:

1-2 Job control and accounting information.

3 The IMSL library is attached.

4-5 The disc containing the program files is identified and mounted.

6 The plotting file will be written here (if it is generated).

7-23 Data files are attached. All data files being attached are from the basic set (see Section 2). If a special geometry was to be implemented a copy of GMDTA would be made containing the desired change. This file, named GM1DTA, would be attached by: ATTACH(GMDTA,GM1DTA,ID=userID).

Similar changes could be introduced in the other data files. To ease the "bookkeeping" problems, it is recommended that the notation just introduced be continued (name other data files AR01DTA, etc.). In this case the output file could be called RUN1.

25 The filename in this line should be HLASIM, HLAMOR, or HLAPAY.

26 The main program routine attached in line 25 is compiled.

27 LMAP is a file for the load map. It will be saved only if the program aborts.

28 MAINLB and IMSL, Libraries for use by the loader, are declared.

29 Files containing the segmented load directives [illegible] attached. The file attached must correspond with the program to be run (SLDIR, for HLASIM; SLDIRM, for HLAMOR; and SLDIRP for HLAPAY).

30-34 Load and execute sequence.

35 Normal termination of job.

36-37 In the event of abnormal job termination dump the load map onto output file.

39-49 Responses to interactive questions (see Sec. 2-1).

Between lines 34 and 35

If plotting files are being generated the following line should be inserted at this point.

CATALOG(PLOT,PLOT1,ID=userID)

This saves the plotting file as PLOT1 (continuing the notation from above).

Alternatively, the complete command string could be inserted here to execute PPLOTF (see below).

The plotting file post processor, PPLOTF, can be executed with the job control sequence shown in Fig. 18. The filename in line 6 is the plotting file written by the main simulation program. Following the example given above, the filename would be PLOT1. THPLOT is the output file and is cataloged in line 10. Following our notation, the filename would be THPLOT1. This file is read into NASA/Ames flight data plotting software.

The gust source stability derivatives program can be executed using the control sequence given in Fig. 18 with a few changes.

1. Make filename in line 25 to be GSRCSB.

2. Delete line 31.

Many of the other lines are unnecessary. If the user wishes to delete those lines, Section 10 of this manual should be consulted.

B. SEGMENTED LOAD DIRECTIVES

The segmented load directives for each of the three programs are listed in Figs. 19, 20, and 21. They are listed here without explanation; they are discussed in the Programmer's Manual.

As noted eariler, the appropriate segmented load directive file must be loaded with each of the three programs.

ORIGINAL PAGE IS
OF POOR QUALITY

```
AIRSHIP.T3ØØ,PN,YD1,YL1
ACCOUNT,userID,T1498L2
ATTACH(IMSL,IMSLLIB,ID=AMESLIB)
SETNAME(HHLABQR)
MOUNT(VSN=DØ419A)
REQUEST(PLOT,*PF)
ATTACH(PYOUTL,PYOUTL,ID=userID)
ATTACH(PAYDTA,PAYDTA,ID=userID)
ATTACH(GMDTA,GMDTA,ID=userID)
ATTACH(ARODTA,ARODTA,ID=userID)
ATTACH(HISDTA,HISDTA,ID=userID)
ATTACH(PLMDTA,PLMDTA,ID=userID)
ATTACH(TRMDTA,TRMDTA,ID=userID)
ATTACH(IFCDTA,IFCDTA,ID=userID)
ATTACH(MORDTA,MORDTA,ID=userID)
ATTACH(ERMSSG,ERMSSG,ID=userID)
ATTACH(OUTLST,OUTLST,ID=userID)
ATTACH(RG1,RG1,ID=userID)
ATTACH(RG2,RG2,ID=userID)
ATTACH(RG3,RG3,ID=userID)
ATTACH(RG4,RG4,ID=userID)
ATTACH(RG5,RG5,ID=userID)
ATTACH(RG6,RG6,ID=userID)
ATTACH(MAINLB,MAINLB,ID=userID)
ATTACH(PG,filename,ID=userID)
FTN(I=PG,L=Ø,PL=5ØØØØ)
REQUEST(LMAP,*PF)
LIBRARY(MAINLB,IMSL,*)
ATTACH(LOADIR,filename,ID=userID)
MAP(ON)
SEGLOAD(I=LOADIR)
LDSET,MAP=X/LMAP,PRESET=INDEF.
LOAD(LGO)
EXECUTE
EXIT
REWIND(LMAP)
COPY(LMAP,OUTPUT)
ZZEOR
T
Ø1
F
T
T
SIX LINES OF COMMENTS
TO APPEAR AS RUN DESCRIPTION
ON OUTPUT LISTING
;;
;;
;;
```

Figure 17. Job Control Sequence to Load and Execute Program

ORIGINAL PAGE IS
OF POOR QUALITY

```
AIRSHIP,T5Ø,PN,YD1,YL1
ACCOUNT,FVSHM,T1498L2
SETNAME(HHLABQR)
MOUNT(VSN-DØ419A)
ATTACH(PPLOTF,PPLOTF,ID=FVSHM)
ATTACH(PLOT,filename,ID=FVSHM)
REQUEST(THPLOT,*PF)
FTN(I=PPLOTF,L=0)
LGO
CATALOG(THPLOT,filename,ID=FVSHM)
EXIT
```

Figure 18. Job Control Sequence to Execute Program PPLOTF

ORIGINAL PAGE IS
OF POOR QUALITY

```
 INPUT1 TREE   QUESTN-INGEOM-INMOOR-INGEAR-INMASS-INEXST-INLALO-MCGDST-C
,GDIST-INHARO
 INPUT2 TREE   INRIFC-INPIFC-INFIFC-INHIFC-INTIFC-INPROP-INMCLC-INSTAT-I
,NATMOS-INSTAB-AEFFCT-MATRIX
 TRMSTO TREE   TRIM-STORE-INTGTR
 TMHIST TREE   INFCSC-INPROF-INGUST-INSTEP
 HLA    TREE   HLASIM-(INPUT1,INPUT2,TRMSTO,LINEAR,TMHIST)
     LEVEL
 CNTL   TREE   SETFCS-PROFIL-TRNFRM-MAXVEC-AUXVEC-BODRAT-EULRAT
 FOR1   TREE   GRAVTY-HCABLE-LGEAR
 SHDOW  TREE   SHADOW-NDMLOC
 INFHUL TREE   RPHIFC-GHVIFC-RPTIFC-GTIFC-HULARO-ROTEFC
 FOR2   TREE   AERO-(SHDOW,LPUARO,INFHUL)
 FORC   TREE   FORCE-(FOR1,FOR2)
 GRP1   TREE   WINDS-(CNTL,FORC)
 GRP2   TREE   LOADT-LOADCA-LOADUA-CALCFC-GETSD-HGEEZ-IACLOD
 GRP3   TREE   EIGEN-WRTSTB
        GLOBAL ATACHP
        GLOBAL ATACH
        GLOBAL ATAHG
        GLOBAL ATMOS
        GLOBAL AUXGST
        GLOBAL AUXVTR
        GLOBAL BTRANS
        GLOBAL CLOSLP
        GLOBAL COMAND
        GLOBAL DELTAX
        GLOBAL EMASMX
        GLOBAL ERATES
        GLOBAL FCDINT
        GLOBAL FCSGNS
        GLOBAL FCSINT
        GLOBAL FCSLIM
        GLOBAL FDBKFL
        GLOBAL FORMOM
        GLOBAL FSAROM
        GLOBAL GBACL
        GLOBAL GCMPRS
        GLOBAL GEARC
        GLOBAL GEARK
        GLOBAL GEARLC
        GLOBAL GEFP
        GLOBAL GEFR
        GLOBAL GERILC
        GLOBAL GFRAMK
        GLOBAL GSTRNG
        GLOBAL GUSTS
        GLOBAL HCBLFO
        GLOBAL HGCOM
```

```
  FILE CONTINUED
GLOBAL HLAROM
GLOBAL HLCNTC
GLOBAL HULL
GLOBAL INVALD
GLOBAL KGHCN
GLOBAL KGP
GLOBAL KGR
GLOBAL KGT
GLOBAL KHP
GLOBAL KHR
GLOBAL KPF
GLOBAL KPH
GLOBAL KPT
GLOBAL KRF
GLOBAL KRH
GLOBAL KRP
GLOBAL KRT
GLOBAL LANDGL
GLOBAL LCOMND
GLOBAL LGCNTC
GLOBAL LNKCOM
GLOBAL LPATCH
GLOBAL LPGCOM
GLOBAL LPUAC
GLOBAL LPU
GLOBAL LTRANS
GLOBAL MASS
GLOBAL MAST
GLOBAL MCLMFL
GLOBAL MECLIM
GLOBAL MODLFL
GLOBAL MUKG
GLOBAL NDHTHT
GLOBAL NDPHT
GLOBAL NDRHT
GLOBAL OPWANL
GLOBAL OUTDTA
GLOBAL OUTHD
GLOBAL PAROCN
GLOBAL PFETHR
GLOBAL PGEOM
GLOBAL POSHCS
GLOBAL PROP
GLOBAL PRPRIG
GLOBAL PSTATE
GLOBAL RAROCN
GLOBAL RELVEL
GLOBAL RGEOM
GLOBAL RHRLOC
GLOBAL RMASCN
```

```
  FILE CONTINUED
GLOBAL RMASCN
GLOBAL ROTOR
GLOBAL RSRCLC
GLOBAL RSTATE
GLOBAL RSWASH
GLOBAL SDOTCP
GLOBAL SENSOR
GLOBAL SHDFCN
GLOBAL SHDPCN
GLOBAL SHDRCN
GLOBAL SPDINT
GLOBAL SPRINT
GLOBAL STABDV
GLOBAL STALLS
GLOBAL SVECTR
GLOBAL TAIL
GLOBAL TAUTS
GLOBAL TDEFLC
GLOBAL TDRVS
GLOBAL TGCOM
GLOBAL TLAROM
GLOBAL TPARAM
GLOBAL TRIMFL
GLOBAL TRMCNT
GLOBAL TRMQT
GLOBAL TSDEFL
GLOBAL UCCFWC
GLOBAL UCTLCS
GLOBAL UNILST
GLOBAL VRINGP
GLOBAL VRINGR
GLOBAL PRINTC
GLOBAL SGUSTS
GLOBAL DGUSTS
GLOBAL GBUFF
GLOBAL POSHD
GLOBAL JETHST
  END  HLASIM
```

Figure 19. Segmented Load Directive File (SLDIR) for Program HLASIM

ORIGINAL PAGE IS
OF POOR QUALITY

```
 INPUT1 TREE   QUESTN-INGEOM-INMOOR-INGEAR-INMASS-INLARO-MCGDST-CGDIST-I
,NHARO
 INPUT2 TREE   INRIFC-INPIFC-INFIFC-INHIFC-INTIFC-INMTRA-INATMOS-INSTAB-
,AEFFCT-MATRIX
 TRMSTO TREE   MTRIM-STORE-MINTGR
 TMHIST TREE   INGUST-INMRST-INSTEP-MCTSTP
 HLA    TREE   HLAMOR-(INPUT1,INPUT2,TRMSTO,MLINAR,TMHIST)
    LEVEL
 CNTL   TREE   MPRFIL-TRNFRM-MAXVEC-AUXVEC-BODRAT-EULRAT
 FOR1   TREE   GRAVTY-HCABLE-LGEAR
 SHDOW  TREE   SHADOW-NDMLOC
 INFHUL TREE   GHVIFC-GTIFC-HULARO-ROTEFC
 FOR2   TREE   MAERO-(SHDOW,MLPARO,INFHUL)
 FORC   TREE   MFORCE-(FOR1,FOR2)
 GRP1   TREE   WINDS-(CNTL,FORC)
 GRP2   TREE   LOADMT-LODMCA-LODMUA-CLCMFC-GETMSD-HGEEZ-IACLOD
 GRP3   TREE   MEIGEN-WRTMSB
        GLOBAL ATACHP
        GLOBAL ATACH
        GLOBAL ATAHG
        GLOBAL ATMOS
        GLOBAL AUXGST
        GLOBAL AUXVTR
        GLOBAL BTRANS
        GLOBAL CALMHD
        GLOBAL COMAND
        GLOBAL DELTAX
        GLOBAL DGUSTS
        GLOBAL EMASMX
        GLOBAL ERATES
        GLOBAL FCDINT
        GLOBAL FCSINT
        GLOBAL FORMOM
        GLOBAL FSAROM
        GLOBAL GBACL
        GLOBAL GBUFF
        GLOBAL GCMPRS
        GLOBAL GEARC
        GLOBAL GEARK
        GLOBAL GEARLC
        GLOBAL GERILC
        GLOBAL GFRAMK
        GLOBAL GSTRING
        GLOBAL GUSTS
        GLOBAL HCBLFO
        GLOBAL HGCOM
        GLOBAL HLAROM
        GLOBAL HLCNTC
        GLOBAL HULL
        GLOBAL IMRLOD
```

FILE CONTINUED

```
GLOBAL INVALD
GLOBAL KGHCN
GLOBAL KGP
GLOBAL KGR
GLOBAL KGT
GLOBAL KHP
GLOBAL KHR
GLOBAL KPF
GLOBAL KPH
GLOBAL KPT
GLOBAL KRF
GLOBAL KRH
GLOBAL KRP
GLOBAL KRT
GLOBAL LANDGL
GLOBAL LGCNTC
GLOBAL LPATCH
GLOBAL LPGCOM
GLOBAL LPUAC
GLOBAL LPU
GLOBAL LTRANS
GLOBAL MASS
GLOBAL MAST
GLOBAL MCLMFL
GLOBAL MDELTX
GLOBAL MODLFL
GLOBAL MTRMCN
GLOBAL MTRMFL
GLOBAL MTRMPC
GLOBAL MUKG
GLOBAL NDHTHT
GLOBAL NDPHT
GLOBAL NDRHT
GLOBAL OPWANT
GLOBAL OUTDTA
GLOBAL OUTHD
GLOBAL PAROCN
GLOBAL PFETHR
GLOBAL PGEOM
GLOBAL POSHCS
GLOBAL POSHD
GLOBAL PLINTC
GLOBAL PROP
GLOBAL PRPRIG
GLOBAL PSTATE
GLOBAL RAROCN
GLOBAL RELVEL
GLOBAL RGEOM
GLOBAL RHRLCC
GLOBAL RMASCN
```

FILE CONTINUED

```
GLOBAL ROTOR
GLOBAL RSRCLC
GLOBAL RSTATE
GLOBAL RSWASH
GLOBAL SENSOR
GLOBAL SGUSTS
GLOBAL SHDFCN
GLOBAL SHDPCN
GLOBAL SHDRCN
GLOBAL SPDINT
GLOBAL SPRINT
GLOBAL STABDV
GLOBAL STALLS
GLOBAL SVECTR
GLOBAL TAIL
GLOBAL TAUTS
GLOBAL TDRVS
GLOBAL TGCOM
GLOBAL TLAROM
GLOBAL TPARAM
GLOBAL TRIMFL
GLOBAL TRMCNT
GLOBAL TRMQT
GLOBAL TSDEFL
GLOBAL UCCFWC
GLOBAL UCTLCS
GLOBAL UNILST
GLOBAL VRINGP
GLOBAL VRINGR
  END  HLAMOR
```

Figure 20. Segmented Load Directive File (SLDIRM) for Program HLAMOR

ORIGINAL PAGE IS
OF POOR QUALITY

```
 INPUT1 TREE   TQUEST-INPGEO-INGEOM-INMOOR-INGEAR-INPMAS-INCABL-INMASS-I
,NEXST-INPARO-INLARO-PCGDST-MCGDST-CGDIST-INHARO
 INPUT2 TREE   INRIFC-INPIFC-INFIFC-INHIFC-INTIFC-INPROP-INMCLC-INSTAT-I
,NATMOS-AERFCT-PMATRX-MATRIX-INSTAB
 TRMSTO TREE   PTRIM-PSTORE-TRIM-STORE-PTRMRT-TINTGR
 TMHIST TREE   INPYST-INFCSC-INPROF-INPGST-INGUST-INSTEP-CKTSTP
 HLAP   TREE   HLAPAY-(INPUT1,INPUT2,TRMSTO,TLINAR,TMHIST)
    LEVEL
 CNTL   TREE   SETFCS-PROFIL-TRNFRM-BODRAT-MAXVEC-AUXVEC-EULRAT
 FOR1   TREE   GRAVITY-HCABLE-LGEAR
 SHDOW  TREE   SHADOW-NDMLOC
 PRPA   TREE   RPIFC-PRPARO
 FUSA   TREE   FUSARO-DSKLOD-CALCHP-RPFIFC
 LPARO  TREE   LPUARO-(ROTARO,PRPA,FUSA)
 INFHUL TREE   RPHIFC-GHVIFC-RPTIFC-GTIFC-HULARO-ROTEFC
 FOR2   TREE   AERO-(SHDOW,LPARO,INFHUL)
 FORC   TREE   FORCE-(FOR1,FOR2)
 GRP1   TREE   WINDS-(CNTL,FORC)
 GRP2   TREE   LOADT-LOADCA-LOADUA-CALCFC-GETSD-HGEEZ-IACLOD-CLCPSD
        TREE   TEIGEN
        TREE   WRT1SB
        GLOBAL ATACHP
        GLOBAL ATACH
        GLOBAL ATAHG
        GLOBAL ATMOS
        GLOBAL AUXGST
        GLOBAL AUXVTR
        GLOBAL BTRANS
        GLOBAL CLOSLP
        GLOBAL COMAND
        GLOBAL DELTAX
        GLOBAL EMASMX
        GLOBAL ERATES
        GLOBAL FCDINT
        GLOBAL FCSGNS
        GLOBAL FCSINT
        GLOBAL FCSLIM
        GLOBAL FDBKFL
        GLOBAL FORMOM
        GLOBAL FSAROM
        GLOBAL GBACL
        GLOBAL GCMPRS
        GLOBAL GEARC
        GLOBAL GEARK
        GLOBAL GEARLC
        GLOBAL GEFP
        GLOBAL GEFR
        GLOBAL GERILC
        GLOBAL GFRAMK
        GLOBAL GSTRNG
```

```
FILE CONTINUED

GLOBAL GUSTS
GLOBAL HCBLFO
GLOBAL HGCOM
GLOBAL HLAROM
GLOBAL HLCNTC
GLOBAL HULL
GLOBAL INVALD
GLOBAL KGHCN
GLOBAL KGP
GLOBAL KGR
GLOBAL KGT
GLOBAL KHP
GLOBAL KHR
GLOBAL KPF
GLOBAL KPH
GLOBAL KPT
GLOBAL KRF
GLOBAL KRH
GLOBAL KRP
GLOBAL KRT
GLOBAL LANDGL
GLOBAL LCOMND
GLOBAL LGCNTC
GLOBAL LNKCOM
GLOBAL LPATCH
GLOBAL LPGCOM
GLOBAL LPUAC
GLOBAL LPU
GLOBAL LTRANS
GLOBAL MASS
GLOBAL MAST
GLOBAL MCLMFL
GLOBAL MECLIM
GLOBAL MODLFL
GLOBAL MUKG
GLOBAL NDHTHT
GLOBAL NDPHT
GLOBAL NDRHT
GLOBAL OPWANT
GLOBAL OUTDTA
GLOBAL OUTHD
GLOBAL PAROCN
GLOBAL PFETHR
GLOBAL PGEOM
GLOBAL POSHCS
GLOBAL PROP
GLOBAL PRPRIG
GLOBAL PSTATE
GLOBAL RAROCN
GLOBAL RELVEL
```

```
FILE CONTINUED

GLOBAL RGEOM
GLOBAL RHRLOC
GLOBAL RMASCN
GLOBAL ROTOR
GLOBAL RSRCLC
GLOBAL RSTATE
GLOBAL RSWASH
GLOBAL SDOTCP
GLOBAL SENSOR
GLOBAL SHDFCN
GLOBAL SHDPCN
GLOBAL SHDRCN
GLOBAL SPDINT
GLOBAL SPRINT
GLOBAL STABDV
GLOBAL STALLS
GLOBAL SVECTR
GLOBAL TAIL
GLOBAL TAUTS
GLOBAL TDEFLC
GLOBAL TDRVS
GLOBAL TGCOM
GLOBAL TLAROM
GLOBAL TPARAM
GLOBAL TRIMFL
GLOBAL TRMCNT
GLOBAL TRMQT
GLOBAL TSDEFL
GLOBAL UCCFWC
GLOBAL UCTLCS
GLOBAL UNILST
GLOBAL VRINGP
GLOBAL VRINGR
GLOBAL POPWNT
GLOBAL PAYLOD
GLOBAL USCLTH
GLOBAL PMASS
GLOBAL CABLK
GLOBAL CABLC
GLOBAL PYAROM
GLOBAL PATCH
GLOBAL PSVCTR
GLOBAL PTRMCN
GLOBAL PTRMFL
GLOBAL PYOPUT
GLOBAL PERATS
GLOBAL PBTRNS
GLOBAL PYGCOM
GLOBAL PGSTRN
GLOBAL PDLTAX
```

```
FILE CONTINUED

GLOBAL PGUSTS
GLOBAL CBLTEN
GLOBAL PLTRNS
GLOBAL PAXVTR
GLOBAL CABLE
GLOBAL PTRMPC
GLOBAL PMDLFL
GLOBAL PPRNTC
GLOBAL PRINTC
GLOBAL SGUSTS
GLOBAL DGUSTS
GLOBAL GBUFF
GLOBAL POSHD
GLOBAL JETHST
GLOBAL PGBUFF
   END HLAPAY
```

Figure 21. Segmented Load Directive File (SLDIRP) fcr Program HLAPAY